Department
of
Electrical Engineering
and
Renewable Energy

# Instrumentation and Control Engineering

Frank Rytkonen, P.E.

Version 1.3

February 2014

© 2013, Frank Rytkonen

This text is based on a book by Tony R. Kuphaldt titled "Lessons in Industrial Instrumentation." The original majority of this text was taken under the Creative Commons License from v1.12 of that book, but as his book has undergone revisions, some (but not all) of the material herein has changed accordingly. I am incredibly indebted to the amount of work Mr. Kuphaldt put into his book, and I greatly appreciate his willingness to share his source documentation under the Creative Commons License. His original work has been modified with some of my own additions and deletions.

This book is a copyrighted work, but licensed under the Creative Commons Attribution 3.0 United States License. To view a copy of this license, turn to page 1210, or visit `http://creativecommons.org/licenses/by/3.0/us/` or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA. The terms and conditions of this license allow for free copying, distribution, and/or modification of all licensed works by the general public.

# Contents

**Preface** **2**

**I The Instrumentation and Control Profession** **6**

**1 Introduction to Instrumentation and Control** **7**
1.1 Example: boiler water level control system . . . 10
1.2 Example: wastewater disinfection . . . 14
1.3 Example: chemical reactor temperature control . . . 16
1.4 Other types of instruments . . . 17
1.4.1 Indicators . . . 17
1.4.2 Recorders . . . 21
1.4.3 Process switches and alarms . . . 25
1.5 Summary . . . 33

**2 Instrumentation and Control Documentation** **35**
2.1 I&C Drawings . . . 37
2.1.1 Process Flow Diagrams . . . 37
2.1.2 Process and Instrument Diagrams . . . 39
2.1.3 Loop diagrams . . . 40
2.1.4 Mechanical installation details . . . 43
2.1.5 Location Drawings . . . 43
2.1.6 Control System Architecture Drawings . . . 44
2.2 Instrument and process equipment symbols . . . 44
2.2.1 Line types . . . 45
2.2.2 Process/Instrument line connections . . . 46
2.2.3 Instrument bubbles . . . 47
2.2.4 Process valve types . . . 48
2.2.5 Valve actuator types . . . 49
2.2.6 Valve failure mode . . . 50
2.2.7 Flow measurement devices (flowing left-to-right) . . . 51
2.2.8 Process equipment . . . 52
2.2.9 Single-line electrical diagram symbols . . . 53
2.2.10 Instrument identification tags . . . 54

2.3 Other I&C Documents . . . 58
2.3.1 Guide Specifications . . . 58
2.3.2 Instrument Specification Sheets . . . 59
2.3.3 The Instrument Index . . . 59
2.3.4 Control System Narratives . . . 60

**II Instrumentation 62**

**3 Analog Signal Transmission 63**
3.1 Analog electronic communication . . . 64
3.2 4 to 20 mA analog current signals . . . 64
3.3 Relating 4 to 20 mA signals to instrument variables . . . 66
3.3.1 Graphical interpretation of signal ranges . . . 69
3.4 2-wire transmitter current loops . . . 71
3.5 3- and 4-wire transmitter current loops . . . 73
3.6 4 to 20 mA loop accessories . . . 75
3.6.1 Resistors . . . 75
3.6.2 Signal conditioners . . . 76
3.7 Controller output current loops . . . 76
3.8 Troubleshooting current loops . . . 78
3.8.1 Using a standard milliammeter to measure loop current . . . 80
3.8.2 Using a clamp-on milliammeter to measure loop current . . . 82
3.8.3 Using "test" diodes to measure loop current . . . 84
3.8.4 Using shunt resistors to measure loop current . . . 85
3.8.5 Troubleshooting current loops with voltage measurements . . . 86
3.8.6 Using loop calibrators . . . 89
3.8.7 NAMUR signal levels . . . 94
3.9 Analog pneumatic communication . . . 95
3.9.1 Foxboro model E69 electro-pneumatic transducer . . . 99
3.9.2 Fisher model 546 electro-pneumatic transducer . . . 104
3.9.3 Fisher-Rosemount model 846 electro-pneumatic transducer . . . 109
3.10 Proper care and feeding of pneumatic instruments . . . 112
3.11 Trade-offs of pneumatic instruments . . . 113

**4 Digital Networks 116**
4.1 Digital communication . . . 118
4.2 Digital representation of numerical data . . . 120
4.2.1 Integer number formats . . . 121
4.2.2 Fixed-point number formats . . . 123
4.2.3 Floating-point number formats . . . 124
4.2.4 Example of industrial number formats . . . 125
4.3 Digital representation of text . . . 128
4.3.1 Morse and Baudot codes . . . 128
4.3.2 EBCDIC and ASCII . . . 129
4.3.3 Unicode . . . 131

4.4 Analog-digital conversion . . . 131
4.4.1 Converter resolution . . . 131
4.4.2 Converter sampling rate . . . 135
4.5 Digital data communication theory . . . 137
4.5.1 Serial communication principles . . . 139
4.5.2 Physical encoding of bits . . . 141
4.5.3 Communication speed . . . 144
4.5.4 Data frames . . . 146
4.5.5 Channel arbitration . . . 152
4.5.6 The OSI Reference Model . . . 156
4.6 TIA-232, 422, and 485 networks . . . 159
4.6.1 TIA-232 . . . 159
4.6.2 TIA-422 and TIA-485 . . . 162
4.7 Ethernet networks . . . 168
4.7.1 Repeaters (hubs) . . . 169
4.7.2 Ethernet cabling . . . 172
4.7.3 Switching hubs . . . 175
4.8 Internet Protocol (IP) . . . 176
4.8.1 IP addresses . . . 177
4.8.2 Subnetworks and subnet masks . . . 180
4.8.3 IP version 6 . . . 183
4.8.4 DNS . . . 183
4.8.5 Command-line diagnostic utilities . . . 184
4.9 Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) . . . 187
4.10 The HART digital/analog hybrid standard . . . 189
4.10.1 Basic concept of HART . . . 189
4.10.2 HART physical layer . . . 192
4.10.3 HART multidrop mode . . . 198
4.10.4 HART multi-variable transmitters . . . 200
4.11 Modbus . . . 200
4.11.1 Modbus data frames . . . 201
4.11.2 Modbus function codes and addresses . . . 202
4.11.3 Modbus function command formats . . . 203

**5 Introduction to Measurement** **211**
5.1 Measurement . . . 212
5.2 The measurement process . . . 212
5.3 Measurement error . . . 212
5.3.1 Systematic error . . . 212
5.3.2 Random error . . . 213
5.4 Instrument Characteristics . . . 213
5.5 Instrument dynamics . . . 215
5.6 Reporting measurements . . . 215

**6 Probability and Statistics for Measurement** **217**
6.1 Estimates of true values . . . 218
6.2 Probabilities and statistics of measurement data . . . 218
6.3 Cumulative distribution function . . . 218
6.4 Probability density functions . . . 218
6.4.1 Moments of a Random Variable . . . 219
6.4.2 Important PDFs in measurement . . . 219

**7 Uncertainty in Measurement** **228**
7.1 Measurement Reporting . . . 229
7.1.1 Relative uncertainty . . . 229
7.1.2 Significant figures . . . 229
7.1.3 The normal distribution . . . 229
7.2 Classifications of Uncertainty . . . 230
7.2.1 Type A Uncertainty . . . 230
7.2.2 Type B Uncertainty . . . 230
7.3 Combining Standard Uncertainties . . . 231
7.3.1 General Propagation of Uncertainty . . . 231
7.3.2 Uncertainty Propagation - Special Cases . . . 231
7.4 Expanded Uncertainty . . . 232
7.5 ISO rules . . . 232

**8 Instrument Calibration** **234**
8.1 Calibration versus ranging . . . 235
8.2 Zero and span adjustments (analog transmitters) . . . 235
8.3 Damping adjustments . . . 238
8.4 LRV and URV settings, digital trim (digital transmitters) . . . 242
8.5 An analogy for calibration versus ranging . . . 245
8.6 Calibration procedures . . . 246
8.6.1 Linear instruments . . . 246
8.6.2 Nonlinear instruments . . . 247
8.6.3 Discrete instruments . . . 248
8.7 Typical calibration errors . . . 249
8.7.1 As-found and as-left documentation . . . 252
8.7.2 Up-tests and Down-tests . . . 253
8.8 NIST traceability . . . 253
8.9 Instrument turndown . . . 254
8.10 Practical calibration standards . . . 255
8.10.1 Electrical standards . . . 255
8.10.2 Temperature standards . . . 257
8.10.3 Pressure standards . . . 260
8.10.4 Flow standards . . . 266
8.10.5 Analytical standards . . . 266

**9 Discrete Event Detection** **272**
9.1 Status of a switch . . . 273
9.2 Actuation Method . . . 275
9.2.1 Hand switches . . . 275
9.2.2 Limit switches . . . 277
9.2.3 Proximity switches . . . 279

**10 Pressure Measurement** **286**
10.1 Manometers . . . 287
10.2 Mechanical pressure elements . . . 292
10.3 Electrical pressure elements . . . 299
10.3.1 Piezoresistive (strain gauge) sensors . . . 299
10.3.2 Differential capacitance sensors . . . 302
10.3.3 Resonant element sensors . . . 308
10.3.4 Mechanical adaptations . . . 310
10.4 Force-balance pressure transmitters . . . 311
10.5 Differential pressure transmitters . . . 315
10.6 Pressure switches . . . 319
10.7 Pressure sensor installation and accessories . . . 325
10.7.1 Valve manifolds . . . 325
10.7.2 Bleed fittings . . . 330
10.7.3 Pressure pulsation damping . . . 331
10.7.4 Remote and chemical seals . . . 333
10.7.5 Filled impulse lines . . . 341
10.7.6 Purged impulse lines . . . 342
10.7.7 Heat-traced impulse lines . . . 344
10.7.8 Water traps and pigtail siphons . . . 346
10.7.9 Mounting brackets . . . 348
10.7.10 Heated enclosures . . . 349
10.8 Process/instrument suitability . . . 351

**11 Temperature and Humidity Measurement** **354**
11.1 Bi-metal temperature sensors . . . 355
11.2 Filled-bulb temperature sensors . . . 357
11.3 Resistance Temperature Elements . . . 360
11.3.1 Thermistors . . . 360
11.3.2 Resistance Temperature Detectors (RTDs) . . . 360
11.4 Thermocouples . . . 365
11.4.1 The Seebeck Effect . . . 365
11.4.2 The Peltier Effect . . . 366
11.4.3 The Thomson Effect . . . 366
11.4.4 Metal Junctions . . . 367
11.4.5 Laws of Thermocouples . . . 367
11.4.6 Thermocouple types . . . 382
11.4.7 Burnout detection . . . 382
11.4.8 Connector and tip styles . . . 383

11.4.9 Connector and tip styles . . . 383
11.5 Optical temperature sensing . . . 388
11.6 Temperature switches . . . 389
11.7 Temperature sensor accessories . . . 391
11.8 Humidity sensors . . . 396
11.8.1 Capacitive humidity sensing . . . 396
11.8.2 Resistive humidity sensing . . . 396
11.8.3 Thermal conductivity humidity sensing . . . 396
11.9 Humidity switches . . . 396
11.10 Process/instrument suitability . . . 397

**12 Level Measurement** **399**
12.1 Level gauges (sightglasses) . . . 401
12.2 Float . . . 407
12.3 Hydrostatic pressure . . . 411
12.3.1 Bubbler systems . . . 416
12.3.2 Transmitter suppression and elevation . . . 418
12.3.3 Compensated leg systems . . . 422
12.3.4 Tank expert systems . . . 427
12.3.5 Hydrostatic interface level measurement . . . 431
12.4 Displacement . . . 436
12.4.1 Displacement interface level measurement . . . 441
12.5 Echo . . . 443
12.5.1 Ultrasonic level measurement . . . 444
12.5.2 Radar level measurement . . . 448
12.6 Laser level measurement . . . 458
12.7 Weight . . . 459
12.8 Capacitive . . . 461
12.9 Radiation . . . 464
12.10 Level switches . . . 465
12.10.1 Float-type level switches . . . 465
12.10.2 Tuning fork level switches . . . 466
12.10.3 Paddle level switches . . . 467
12.10.4 Ultrasonic level switches . . . 468
12.10.5 Capacitive level switches . . . 469
12.10.6 Conductive level switches . . . 470
12.11 Level sensor accessories . . . 472
12.12 Level instrument installation details . . . 477

**13 Flow Measurement** **480**
13.1 Pressure-based flowmeters . . . 482
13.1.1 Venturi tubes and basic principles . . . 488
13.1.2 Orifice plates . . . 496
13.1.3 Other differential producers . . . 508
13.1.4 Proper installation . . . 515
13.1.5 High-accuracy flow measurement . . . 519

13.1.6 Equation summary . . . 525
13.2 Laminar flowmeters . . . 526
13.3 Variable-area flow gauges . . . 528
13.3.1 Rotameters . . . 528
13.3.2 Open-channel flowmeters . . . 531
13.4 Velocity-based flowmeters . . . 539
13.4.1 Turbine flowmeters . . . 540
13.4.2 Vortex flowmeters . . . 545
13.4.3 Magnetic flowmeters . . . 549
13.4.4 Ultrasonic flowmeters . . . 556
13.5 Inertia-based (true mass) flowmeters . . . 558
13.5.1 Coriolis flowmeters . . . 558
13.6 Thermal-based (mass) flowmeters . . . 568
13.7 Positive displacement flowmeters . . . 571
13.8 Weighfeeders . . . 573
13.9 Change-of-quantity flow measurement . . . 573
13.10 Insertion flowmeters . . . 577
13.11 Flow switches . . . 583
13.12 Process/instrument suitability . . . 584

**14 Electrical Measurements** **588**
14.1 Current . . . 589
14.1.1 Current transformers (CT) . . . 589
14.1.2 CT design . . . 597
14.1.3 Ratings . . . 597
14.2 Voltage . . . 597
14.2.1 Voltage Transformers (VT) . . . 597
14.2.2 Capacitively-Coupled Voltage Transformers (CCVTs) . . . 601
14.3 Transformer polarity . . . 601
14.4 Instrument transformer burdens and safety . . . 608
14.5 Instrument transformer test switches . . . 609

**15 Signal Characterization** **615**
15.1 Flow measurement in open channels . . . 622
15.2 Liquid volume measurement . . . 624
15.3 Radiative temperature measurement . . . 633
15.4 Analytical measurements . . . 634

**16 Final Control Elements** **637**
16.1 On/off valves . . . 638
16.2 Solenoid valves . . . 639
16.2.1 2-way solenoid valves . . . 640
16.2.2 3-way solenoid valves . . . 642
16.2.3 Normal energization states . . . 646
16.3 Control valves . . . 647
16.3.1 Sliding-stem valves . . . 647

16.3.2 Rotary-stem valves . . . 656
16.3.3 Valve packing . . . 659
16.3.4 Valve seat leakage . . . 665
16.3.5 Control valve actuators . . . 666
16.3.6 Valve failure mode . . . 676
16.3.7 Actuator bench-set . . . 680
16.3.8 Pneumatic actuator response . . . 686
16.3.9 Valve positioners . . . 689
16.3.10 Control valve sizing . . . 700
16.3.11 Control valve characterization . . . 710
16.3.12 Control valve problems . . . 723
16.3.13 Dampers and louvers . . . 737
16.4 On/off electric motor control circuits . . . 740
16.4.1 Two-wire motor control . . . 740
16.4.2 Three-wire motor control . . . 742
16.4.3 Manual/remote motor control . . . 744
16.4.4 Networked motor control . . . 744
16.5 Variable-speed motor controls . . . 746
16.5.1 DC motor speed control . . . 747
16.5.2 AC motor speed control . . . 755
16.5.3 Motor drive features . . . 759
16.5.4 Metering pumps . . . 760

**III Control Systems** **764**

**17 Introduction to Control Systems** **765**
17.1 System structure . . . 766
17.1.1 State . . . 766
17.1.2 Inputs . . . 766
17.1.3 Outputs . . . 766
17.2 System properties . . . 766
17.2.1 Linearity . . . 767
17.2.2 Time-invariance . . . 767
17.3 Transfer functions . . . 767
17.3.1 Evans form . . . 767
17.3.2 Bode form . . . 768
17.4 System problems . . . 768
17.4.1 System dynamics . . . 769
17.4.2 System design . . . 769
17.4.3 System identification . . . 770
17.4.4 System control . . . 770
17.5 Types of control . . . 771
17.5.1 Open-loop control . . . 771
17.5.2 Closed-loop control . . . 772
17.6 Block diagram mathematics . . . 773

**18 Process Dynamics** **778**
18.1 Laplace transform review . . . 779
18.1.1 Definition . . . 779
18.1.2 Laplace Transforms . . . 779
18.2 Stable (self-regulating) processes . . . 779
18.2.1 Examples . . . 779
18.2.2 Mathematical descriptions . . . 781
18.2.3 Inverse-response processes . . . 792
18.3 Unstable processes . . . 794
18.3.1 Integrating processes . . . 794
18.3.2 Runaway processes . . . 801
18.4 Delay processes . . . 802

**19 System Modeling** **807**
19.1 Component analogies . . . 808
19.1.1 Variables . . . 808
19.1.2 Electrical . . . 808
19.1.3 Mechanical - Translational Motion . . . 809
19.1.4 Mechanical - Rotational Motion . . . 809
19.1.5 Thermal . . . 810
19.1.6 Fluid . . . 810
19.2 Conservation laws . . . 811
19.2.1 Electrical conservation laws . . . 811
19.2.2 Mechanical translational conservation laws . . . 812
19.2.3 Mechanical rotational conservation laws . . . 812
19.2.4 Thermal conservation laws . . . 812
19.2.5 Fluid conservation laws . . . 812
19.3 Transfer functions . . . 812
19.3.1 Evans form . . . 813
19.3.2 Bode form . . . 813
19.3.3 Transfer function model limitations . . . 813
19.4 State space models . . . 815
19.4.1 State equations . . . 816
19.4.2 Creating a state space model . . . 818
19.4.3 Transformations to/from state space representations . . . 822
19.5 Linearization . . . 824
19.5.1 Linearizing differentiable functions . . . 824
19.5.2 Linearizing differentiable nonlinear differential equations . . . 825
19.5.3 Developing transfer functions for systems with differentiable nonlinearities . . . 825
19.5.4 Linearizing periodic switching differential equations . . . 827

**20 System Identification** **835**
20.1 Introduction to system identification . . . 836
20.2 Obtaining data . . . 836
20.2.1 Step responses . . . 836
20.2.2 Frequency responses . . . 837

20.2.3 More complex inputs . . . 838
20.3 Data preprocessing . . . 839
20.4 Selecting a model . . . 839
20.4.1 Types of models . . . 839
20.4.2 Classes of models . . . 847
20.5 Model validation . . . 848

**21 System Analysis** **850**
21.1 Type number . . . 851
21.2 Stability . . . 852
21.2.1 Routh-Hurwitz analysis . . . 852
21.3 Root locus analysis . . . 856
21.4 Frequency response analysis . . . 862
21.4.1 Bode plot . . . 863
21.4.2 Nyquist plot . . . 864
21.4.3 Nichols plot . . . 864

**22 Principles of Feedback Controllers** **866**
22.1 Basic feedback control principles . . . 867
22.2 On/off control . . . 872
22.3 Proportional-only control . . . 874
22.4 Proportional-only offset . . . 878
22.5 Integral (reset) control . . . 883
22.6 Derivative (rate) control . . . 885
22.7 Summary of PID control terms . . . 886
22.7.1 Proportional control mode (P) . . . 887
22.7.2 Integral control mode (I) . . . 887
22.7.3 Derivative control mode (D) . . . 888
22.8 P, I, and D responses graphed . . . 889
22.8.1 Responses to a single step-change . . . 889
22.8.2 Responses to a momentary step-and-return . . . 890
22.8.3 Responses to two momentary steps-and-returns . . . 892
22.8.4 Responses to a ramp-and-hold . . . 893
22.8.5 Responses to an up-and-down ramp . . . 895
22.8.6 Responses to a sine wave . . . 896
22.9 Different PID equations . . . 898
22.9.1 Parallel PID equation . . . 898
22.9.2 Ideal PID equation . . . 899
22.9.3 Series PID equation . . . 899
22.10 Analog electronic PID controllers . . . 899
22.11 Digital PID controllers . . . 906
22.11.1 Digital PID algorithms . . . 906
22.11.2 Stand-alone digital controllers . . . 907
22.11.3 Direct digital controllers (DDC) . . . 912
22.11.4 Programmable logic controllers (PLC) . . . 914
22.11.5 Supervisory Control And Data Acquisition (SCADA) . . . 916

22.11.6 Distributed Control Systems (DCS) . . . 919
22.11.7 Fieldbus control . . . 924
22.12 Practical PID controller features . . . 927
22.12.1 Manual and automatic modes . . . 927
22.12.2 Output and setpoint tracking . . . 928
22.12.3 Alarm capabilities . . . 930
22.12.4 Output and setpoint limiting . . . 930
22.12.5 Security . . . 931

**23 PID Controller Tuning** **934**
23.1 Before tuning . . . 935
23.1.1 Identifying operational needs . . . 936
23.1.2 Identifying process and system hazards . . . 937
23.1.3 Identifying the problem(s) . . . 938
23.1.4 Final precautions . . . 939
23.2 Heuristic PID tuning procedures . . . 939
23.2.1 Ziegler-Nichols closed-loop ("Ultimate Gain") . . . 940
23.2.2 Ziegler-Nichols open-loop . . . 944
23.2.3 Other heuristic PID tuning procedures . . . 947
23.2.4 Features of P, I, and D actions . . . 947
23.2.5 Tuning recommendations based on process dynamics . . . 948
23.3 Tuning techniques compared . . . 949
23.3.1 Tuning a "generic" process . . . 950
23.3.2 Tuning a liquid level process . . . 954
23.3.3 Tuning a temperature process . . . 958

**24 Programmable Logic Controllers** **965**
24.1 PLC examples . . . 966
24.2 Input/Output (I/O) capabilities . . . 974
24.2.1 Discrete I/O . . . 975
24.2.2 Analog I/O . . . 981
24.2.3 Network I/O . . . 983
24.3 Logic programming . . . 983
24.3.1 Relating I/O status to virtual elements . . . 984
24.3.2 The PLC scan . . . 991
24.3.3 Memory maps and I/O addressing . . . 992
24.4 Ladder Diagram (LD) programming . . . 996
24.4.1 Contacts and coils . . . 998
24.4.2 Counters . . . 1016
24.4.3 Timers . . . 1022
24.4.4 Data comparison instructions . . . 1027
24.4.5 Math instructions . . . 1030
24.4.6 Sequencers . . . 1033
24.5 Function Block Diagram (FBD) programming . . . 1043
24.5.1 A FBD programming example . . . 1043
24.5.2 A second FBD programming example . . . 1043

24.6 Structured Text (ST) programming . . . 1045
24.6.1 A ST programming example . . . 1045
24.7 Instruction List (IL) programming . . . 1045
24.7.1 An IL programming example . . . 1046
24.8 Sequential Function Charts (SFC) . . . 1046
24.8.1 Sequencing . . . 1047
24.8.2 Looping . . . 1047
24.8.3 Selection . . . 1048
24.8.4 Simultaneous execution . . . 1050
24.9 Human-Machine Interfaces . . . 1050

**25 Analytical Control Design** **1056**
25.1 Controller design goals . . . 1057
25.2 Performance evaluation . . . 1057
25.2.1 Tracking vs. regulating controllers . . . 1057
25.2.2 Time domain criteria . . . 1057
25.2.3 Frequency domain criteria . . . 1059
25.3 Controller design methods . . . 1062
25.3.1 Root locus design . . . 1062
25.3.2 Frequency response design using loop shaping . . . 1062
25.4 Controller implementation . . . 1063
25.4.1 Analog compensator implementation . . . 1063
25.4.2 Digital compensator implementation . . . 1066

**26 Multivariable Process Control Strategies** **1069**
26.1 Supervisory control . . . 1070
26.2 Split-range control . . . 1073
26.2.1 Complementary valve sequencing . . . 1073
26.2.2 Exclusive valve sequencing . . . 1074
26.2.3 Progressive valve sequencing . . . 1076
26.2.4 Valve sequencing methods . . . 1078
26.3 Cascade control . . . 1083
26.4 Feedforward control . . . 1090
26.4.1 Feedforward with dynamic compensation . . . 1103
26.5 Ratio control . . . 1121
26.6 Limit, Selector, and Override controls . . . 1130
26.6.1 Limit controls . . . 1133
26.6.2 Selector controls . . . 1138
26.6.3 Override controls . . . 1142

**IV Appendices** **1148**

**A Instrumentation Connections** **1149**
A.1 Pipe and pipe fittings . . . 1149
A.1.1 Flanged pipe fittings . . . 1149

A.1.2 Tapered thread pipe fittings . . . 1153
A.1.3 Parallel thread pipe fittings . . . 1156
A.1.4 Sanitary pipe fittings . . . 1157
A.2 Tube and tube fittings . . . 1161
A.2.1 Compression tube fittings . . . 1162
A.2.2 Common tube fitting types and names . . . 1165
A.2.3 Bending instrument tubing . . . 1168
A.3 Electrical signal and control wiring . . . 1169
A.3.1 Connections and wire terminations . . . 1171
A.3.2 DIN rail . . . 1180
A.3.3 Cable routing . . . 1183
A.3.4 Signal coupling and cable separation . . . 1190
A.3.5 Electric field (capacitive) decoupling . . . 1195
A.3.6 Magnetic field (inductive) decoupling . . . 1201
A.3.7 High-frequency signal cables . . . 1203

**B Example Sequence of Operations 1207**
B.1 System Name . . . 1207
B.2 System ID Number . . . 1207
B.3 Equipment List . . . 1207
B.4 System Description . . . 1207
B.4.1 General Description . . . 1207
B.4.2 Philosophy of Operation . . . 1207
B.5 Control Loop Descriptions . . . 1208
B.5.1 Loop TIC-XX . . . 1208
B.5.2 Loop TS-XX . . . 1208
B.6 Reference Documents . . . 1209
B.7 Troubleshooting . . . 1209

**C Creative Commons Attribution License 1210**
C.1 A simple explanation of your rights . . . 1210
C.2 Legal code . . . 1210

# Preface

I did not want to write this book either (see the original preface below). Fortunately, I didn't have to write most of it.

As a practicing instrumentation and controls (I&C) engineer turned college professor, I intended to teach an integrated two-course sequence on both topics. Finding a single textbook to address my needs was a problem. The majority of texts I found were written for either instrumentation or control, but not both. Many textbooks I found were written for technicians in a "math-phobic" style well below the level of engineering education I intended to provide. Other textbooks I found contained great information and were written for engineers, but neglected the practical side in favor of the theoretical. Many of the texts contained outdated material and methods of analysis that are no longer in use. Also, most of the textbooks I found do not work well with my teaching style. I needed a text giving good, current, practical information and theoretical coverage at the same time, with coverage of material I desired, and written in a way that suited my teaching style.

I found *Lessons in Industrial Instrumentation* by Mr. Tony Kuphaldt published on the Internet under a Creative Commons open-source license. He made all of his source material available for download. I am deeply indebted to Mr. Kuphaldt for the amount of material that fit my needs. As he says in the preface to the original work, "There do exist many fine references on the subject of industrial instrumentation and control," but they are just that - references. They are not textbooks, and are not typically suitable for pedagogy. His book was written with instruction as the primary objective. I have endeavored to raise the level of the text to make it suitable for engineering students, removed some material that wasn't relevant to my courses, and added additional material as necessary.

This text is written in three parts. Part I covers the basics of the I&C engineering profession, such as an introduction to the field and information regarding design. Instrumentation is introduced in Part II, and measurement methods and equipment are introduced along with measurement theory. Part III discusses control theory, including process dynamics, stability, and modeling, along with control design methods and equipment. Hopefully, the three parts combine to provide an integrated introduction to the world of I&C engineering.

Students who wish to use this book should have a background that includes the following topics: physics, math through differential equations and linear algebra, electrical circuits through transient analysis with Laplace transforms and Bode plots, thermodynamics, heat transfer, and fluid flow.

Frank Rytkonen, P.E.
Oregon Institute of Technology

# Original Preface

From *Lessons in Industrial Instrumentation*, Version 1.12

I did not want to write this book . . . honestly.

My first book project began in 1998, titled *Lessons In Electric Circuits*, and I didn't call "quit" until six volumes and five years later. Even then it was not complete, but being an open-source project it gained traction on the internet to the point where other people took over its development and it grew fine without me. The impetus for writing this first tome was a general dissatisfaction with available electronics textbooks. Plenty of textbooks exist to describe things, but few really *explain* things well for students, and the field of electronics is no exception. I wanted my book(s) to be different, and so they were. No one told me how time-consuming it was going to be to write them, though!

The next few years' worth of my spare time went to developing a set of question-and-answer worksheets designed to teach electronics theory in a Socratic, active-engagement style. This project proved quite successful in my professional life as an instructor of electronics. In the summer of 2006, my job changed from teaching electronics to teaching industrial instrumentation, and I decided to continue the Socratic mode of instruction with another set of question-and-answer worksheets.

However, the field of industrial instrumentation is not as well-represented as general electronics, and thus the array of available textbooks is not as vast. I began to re-discover the drudgery of trying to teach with inadequate texts as source material. The basis of my active teaching style was that students would spend time researching the material on their own, then engage in Socratic-style discussion with me on the subject matter when they arrived for class. This teaching technique functions in direct proportion to the quality and quantity of the research sources at the students' disposal. Despite much searching, I was unable to find a textbook adequately addressing my students' learning needs. Many textbooks I found were written in a shallow, "math-phobic" style well below the level I intended to teach to. Some reference books I found contained great information, but were often written for degreed engineers with lots of Laplace transforms and other mathematical techniques well above the level I intended to teach to. Few on either side of the spectrum actually made an effort to explain certain concepts students generally struggle to understand. I needed a text giving good, practical information and theoretical coverage at the same time.

In a futile effort to provide my students with enough information to study outside of class, I scoured the internet for free tutorials written by others. While some manufacturer's tutorials were nearly perfect for my needs, others were just as shallow as the textbooks I had found, and/or were little more than sales brochures. I found myself starting to write my own tutorials on specific topics to "plug the gaps," but then another problem arose: it became troublesome for students to navigate through dozens of tutorials in an effort to find the information they needed in their studies. What my students really needed was a *book*, not a smorgasbord of tutorials.

So here I am again, writing another textbook. This time around I have the advantage of wisdom gained from the first textbook project. For this project, I will *not*:

- . . . attempt to maintain a parallel book in HTML markup (for direct viewing on the internet). I had to go to the trouble of inventing my own quasi-XML markup language last time in an effort to generate multiple format versions of the book from the same source code. Instead, this time I will use stock LaTeX as the source code format and regular Adobe PDF format for

the final output, which anyone may read thanks to its ubiquity. If anyone else desires the book in a different format, I will gladly let them deal with issues of source code translation. Not that this should be a terrible problem for anyone technically competent in markup languages, as LaTeX source is rather easy to work with.

- . . . use a GNU GPL-style copyleft license. Instead, I will use the Creative Commons Attribution-only license, which is far more permissive for anyone wishing to incorporate my work into derivative works. My interest is maximum flexibility for those who may adapt my material to their own needs, not the imposition of certain philosophical ideals.
- . . . start from a conceptual state of "ground zero." I will assume the reader has certain familiarity with electronics and mathematics, which I will build on. If a reader finds they need to learn more about electronics, they should go read *Lessons In Electric Circuits*.
- . . . avoid using calculus to help explain certain concepts. Not all my readers will understand these parts, and so I will be sure to explain what I can without using calculus. However, I want to give my more mathematically adept students an opportunity to see the power of calculus applied to instrumentation where appropriate. By occasionally applying calculus and explaining my steps, I also hope this text will serve as a practical guide for students who might wish to learn calculus, so they can see its utility and function in a context that interests them.

There do exist many fine references on the subject of industrial instrumentation. I only wish I could condense their best parts into a single volume for my students. Being able to do so would certainly save me from having to write my own! Listed here are some of the best books I can recommend for those wishing to explore instrumentation outside of my own presentation:

- *Instrument Engineers' Handbook* series (Volumes I, II, and III), edited by Béla Lipták – by far my favorite modern references on the subject. Unfortunately, there is a fair amount of material within that lies well beyond my students' grasp (Laplace transforms, etc.), and the volumes are incredibly bulky and expensive (nearly 2000 pages, and at a cost of nearly $200.00, *apiece!*). These texts also lack some of the basic content my students do need, and I don't have the heart to tell them to buy yet *another* textbook to fill the gaps.
- *Handbook of Instrumentation and Controls*, by Howard P. Kallen. Perhaps the best-written textbook on general instrumentation I have ever encountered. Too bad it is both long out of print – my copy dates 1961 – and technologically dated. Like most American textbooks written during the years immediately following Sputnik, it is a masterpiece of practical content and conceptual clarity. I consider books like this useful for their presentations of "first principles," which of course are timeless.
- *Industrial Instrumentation Fundamentals*, by Austin E. Fribance. Another great post-Sputnik textbook – my copy dates 1962.
- *Instrumentation for Process Measurement and Control*, by Norman A. Anderson. An inspiring effort by someone who knows the art of teaching as well as the craft of instrumentation. Too bad the content doesn't seem to have been updated since 1980.
- Practically anything written by Francis Greg Shinskey.

Whether or not I achieve my goal of writing a better textbook is a judgment left for others to make. One decided advantage my book will have over all the others is its *openness*. If you don't like anything you see in these pages, you have the right to modify it to your liking! Delete content, add content, modify content – it's all fair game thanks to the Creative Commons licensing. My

only condition is declared in the license: you must give me credit for my original authorship. What you do with it beyond that is wholly up to you[1]. This way, perhaps I can spare someone else from having to write their own textbook from scratch!

Tony Kuphaldt
Bellingham Technical College

[1]This includes selling copies of it, either electronic or print. Of course, you must include the Creative Commons license as part of the text you sell (see Section 4, subsection 1 of the license for details), which means anyone will be able to tell it is an open text and can probably figure out how to download an electronic copy off the internet for free. The only way you're going to make significant money selling this text is to add your own value to it, either in the form of expansions or bundled product (e.g. simulation software, learning exercises, etc.), which of course is perfectly fair – you must profit from your *own* labors. All my work does for you is give you a starting point.

# Part I

# The Instrumentation and Control Profession

# Chapter 1

# Introduction to Instrumentation and Control

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: process, process variable, setpoint, primary sensing element (sensor), transducer, transmitter, lower (upper) range value, zero, span, controller, final control element, manipulated variable, automatic mode, manual mode
2. Identify sensors, controllers, and final control elements in common control systems
3. Identify process variables, setpoints, and manipulated variables in common control systems
4. Discuss the fields of knowledge that must be learned to become a competent instrumentation and control engineer
5. Explain why lifelong learning is important for someone working in the field of instrumentation and control

Instrumentation is the branch of engineering that deals with the process of measurement. Control is the engineering discipline that manipulates matter and energy in a system to create desired performance. Since control of a process usually requires measurement, they are often lumped together as an engineering discipline referred to as *instrumentation and control (I&C) engineering*. I&C engineering is used to measure and automate systems. Applications of this practice abound in modern research, industry, and everyday living. From automobile engine control systems to home thermostats to aircraft autopilots to the manufacture of pharmaceutical drugs, automation is present in nearly every system in modern life. This chapter explains some of the fundamental principles of industrial instrumentation and control.

The first step, naturally, is measurement. If a parameter cannot be measured (either directly or indirectly), it is really pointless to try to control it. This parameter usually takes one of the following forms in industry:

- Fluid pressure
- Fluid flow rate
- The temperature of an object
- Fluid volume stored in a vessel
- Chemical concentration
- Machine position, motion, or acceleration
- Physical dimension(s) of an object
- Count (inventory) or presence of objects
- Electrical voltage, current, or resistance

Once the desired quantity is measured, a signal representing this quantity is typically transmitted to an indicating or computing device where either human or automated decision then takes place. If the controlling decision is automated, the controller sends an action signal to a final controlling device which then influences the quantity being measured. This process of measure, decide, and act form the basis of any control system.

This final control device usually takes one of the following forms:

- Control valve (for throttling the flow rate of a fluid)
- Electric motor driving a mechanical component
- Electric heater

Both the measurement device and the final control device connect to some physical system called the *process*. A general block diagram of this is shown in Figure 1.1.

Figure 1.1: Process control block diagram

The common home thermostat is an example of a measurement and control system, with the home's internal air temperature being the "process" under control. In this example, the thermostat usually serves two functions: sensing and control, while the home's heater adds heat to the home to increase temperature, and/or the home's air conditioner extracts heat from the home to decrease temperature. The job of this control system is to maintain air temperature at some comfortable level, with the heater or air conditioner taking action to correct temperature if it strays too far from the desired value (called the *setpoint*).

Industrial measurement and control systems have their own unique terms and standards, which is the primary focus of this lesson. Here are some common instrumentation terms and their definitions [1]:

**Process**: The physical system to control or measure. *Examples: water filtration system, molten metal casting system, steam boiler, oil refinery unit, power generation unit.*

**Process Variable**, or **PV**: The specific quantity measured in a process. *Examples: pressure, level, temperature, flow, electrical conductivity, pH, position, speed, vibration.*

**Setpoint**, or **SP**: The desired value of the process variable. In other words, the "target" value of the process variable.

**Primary Sensing Element**, or **PSE**: A device that directly senses the process variable and translates that sensed quantity into an analog representation (electrical voltage, current, resistance; mechanical force, motion, etc.). *Examples: thermocouple, thermistor, bourdon tube, microphone, potentiometer, electrochemical cell, accelerometer.*

**Transducer**: A device that converts one standardized instrumentation signal into another standardized instrumentation signal, and/or performs some sort of processing on that signal. Often referred to as a *converter* and sometimes as a "relay." *Examples: I/P converter (converts 4-20 mA*

*electric signal into 3-15 psig pneumatic signal), P/I converter (converts 3-15 psig pneumatic signal into 4-20 mA electric signal), square-root extractor (calculates the square root of the input signal).*

Note: in general science parlance, a "transducer" is any device that converts one form of energy into another, such as a microphone or a thermocouple. In industrial instrumentation, however, it is typical to use the phrase "primary sensing element" to describe this concept and reserve the word "transducer" to specifically refer to a conversion device for standardized instrumentation signals.

**Transmitter**: A device that translates the signal produced by a primary sensing element (PSE) into a *standardized* instrumentation signal such as 3-15 psig air pressure, 4-20 mA DC electric current, Fieldbus digital signal packet, etc., which may then be conveyed to an indicating device, a controlling device, or both.

**Lower- and Upper-range values**, abbreviated **LRV** and **URV**, respectively: the values of process measurement deemed to be 0% and 100% of a transmitter's calibrated range. For example, if a temperature transmitter is calibrated to measure a range of temperature starting at 300 degrees Celsius and ending at 500 degrees Celsius, 300 degrees would be the LRV and 500 degrees would be the URV.

**Zero** and **Span**: alternative descriptions to LRV and URV for the 0% and 100% points of an instrument's calibrated range. "Zero" refers to the beginning-point of an instrument's range (equivalent to LRV), while "span" refers to the width of its range (URV − LRV). For example, if a temperature transmitter is calibrated to measure a range of temperature starting at 300 degrees Celsius and ending at 500 degrees Celsius, its zero would be 300 degrees and its span would be 200 degrees.

**Controller**: A device that receives a process variable (PV) signal from a primary sensing element (PSE) or transmitter, compares that signal to the desired value for that process variable (called the setpoint), and calculates an appropriate output signal value to be sent to a final control element (FCE) such as an electric motor or control valve.

**Final Control Element**, or **FCE**: A device that receives the signal from a controller to directly influence the process. Also known as an *actuator. Examples: variable-speed electric motor, control valve, electric heater.*

**Manipulated Variable**, or **MV**: Another term to describe the output signal generated by a controller. This is the signal commanding ("manipulating") the final control element to influence the process. Also known as the Control Variable, or CV.

**Automatic mode**: When the controller generates an output signal based on the relationship of process variable (PV) to the setpoint (SP).

**Manual mode**: When the controller's decision-making ability is bypassed to let a human operator directly determine the output signal sent to the final control element.

Some practical examples of measurement and control systems are presented below to better illustrate these fundamental concepts.

## 1.1 Example: boiler water level control system

Steam boilers are very common in industry, principally because steam power is so useful. Common uses for steam in industry include doing mechanical work (e.g. a steam engine moving some sort of machine), heating, producing vacuums (through the use of "steam eductors"), and augmenting chemical processes (e.g. reforming of natural gas into hydrogen and carbon dioxide).

The process of converting water into steam is quite simple: heat up the water until it boils. Anyone who has ever boiled a pot of water for cooking knows how this process works. Making steam continuously, however, is a little more complicated. An important variable to measure and control in a continuous boiler is the level of water in the "steam drum" (the upper vessel in a water-tube boiler). In order to safely and efficiently produce a continuous flow of steam, the steam drum must never run too low on water, or too high. If there is not enough water in the drum, the water tubes may run dry and burn through from the heat of the fire. If there is too much water in the drum, liquid water may be carried along with the flow of steam, causing impingement damage downstream to piping and/or equipment.

The essential elements of a water level control system, including transmitter, controller, and control valve, are shown in Figure 1.2.

Figure 1.2: Water level control system

The first instrument in this control system is the *level transmitter*, or "LT". The purpose of this device is to sense the water level in the steam drum and report that measurement to the controller

in the form of an instrument signal. In this case, the type of signal is *pneumatic*: a variable air pressure sent through metal or plastic tubes. The greater the water level in the drum, the more air pressure output by the level transmitter. Since the transmitter is pneumatic, it must be supplied with a source of clean, compressed air on which to run. This is the meaning of the "A.S." tube (Air Supply) entering the top of the transmitter.

This pneumatic signal is sent to the next instrument in the control system, the *level indicating controller*, or "LIC". The purpose of this instrument is to compare the level transmitter's signal with a *setpoint* value entered by a human operator (the desired water level in the steam drum). The controller then generates an *output* signal telling the control valve to either introduce more or less water into the boiler to maintain the steam drum water level at setpoint. As with the transmitter, the controller in this system is pneumatic, operating entirely on compressed air. This means the output of the controller is also a variable air pressure signal, just like the signal output by the level transmitter. Naturally, the controller requires a constant supply of clean, compressed air on which to run, which explains the "A.S." (Air Supply) tube connecting to it.

The last instrument in this control system is the control valve, being operated directly by the air pressure signal generated by the controller. This particular control valve uses a large diaphragm to convert the air pressure signal into a mechanical force to move the valve open and closed. A large spring inside the valve mechanism provides the force necessary to return the valve to its normal position, while the force generated by the air pressure on the diaphragm works against the spring to move the valve the other direction.

When the controller is placed in the "automatic" mode, it will move the control valve to whatever position it needs to be in order to maintain a constant steam drum water level. The phrase "whatever position it needs to be" suggests that the relationship between the controller output signal, the process variable signal (PV), and the setpoint (SP) can be quite complex. If the controller senses a water level above setpoint, it will take whatever action is necessary to bring that level back down to setpoint. Conversely, if the controller senses a water level below setpoint, it will take whatever action is necessary to bring that level up to setpoint. What this means in a practical sense is that the controller's output signal (equating to valve position) is just as much a function of process load (i.e., how much steam is being used from the boiler) as it is a function of setpoint.

Consider a situation where the steam demand from the boiler is very low. If there isn't much steam being drawn off the boiler, this means there will be little water boiled into steam and therefore little need for additional feedwater to be pumped into the boiler. Therefore, in this situation, it is expected that the control valve hover near the fully-closed position, allowing just enough water into the boiler to keep the steam drum water level at setpoint.

If, however, there is great demand for steam from this boiler, the rate of evaporation will be much higher. This means the control system will have to add feedwater to the boiler at a much greater flow rate in order to maintain the steam drum water level at setpoint. In this situation the control valve should much closer to being fully-open as the control system "works harder" to maintain a constant water level in the steam drum.

A human operator running this boiler has the option of placing the controller into "manual" mode. In this mode, the control valve position is under direct control of the human operator, with the controller essentially ignoring the signal sent from the water level transmitter. Being an indicating controller, the controller faceplate will still show how much water is in the steam drum, but it is now the human operator's sole responsibility to move the control valve to the appropriate position to hold water level at setpoint.

Manual mode is useful to the human operator(s) during start-up and shut-down conditions. It is

also useful to the technician for troubleshooting a misbehaving control system. When a controller is in automatic mode, the output signal (sent to the control valve) changes in response to the process variable (PV) and setpoint (SP) values. Changes in the control valve position, in turn, naturally affect the process variable signal through the physical relationships of the process. This is a situation where causality is uncertain. If the process variable is changing erratically over time, does this mean there is faulty transmitter (outputting an erratic signal), or does it mean the controller output is erratic (causing the control valve to shift position unnecessarily), or does it mean the steam demand is fluctuating and causing the water level to vary as a result? So long as the controller remains in automatic mode, it is impossible to be completely sure what is causing what to happen, because the chain of causality is actually a *loop*, with everything affecting everything else in the system.

A simple way to diagnose such a problem is to place the controller in manual mode. Now the output signal to the control valve will be fixed at whatever level the human operator or technician sets it to. If the process variable signal suddenly stabilizes, the problem has something to do with the controller output. If the process variable signal suddenly becomes even more erratic once the controller is in manual mode, the controller was actually trying to do its job properly in automatic mode and the cause of the problem lies within the process itself.

As was mentioned before, this is an example of a *pneumatic* (compressed air) control system, where all the instruments operate on compressed air, and use compressed air as the signaling medium. Pneumatic instrumentation is an old technology, dating back many decades. While most modern instruments are electronic in nature, pneumatic instruments still find application within industry. The most common industry standard for pneumatic pressure signals is 3 to 15 psig, with 3 psig representing low end-of-scale and 15 psig representing high end-of-scale. Alternative pressure ranges for pneumatic signals sometimes encountered in industry include 3 to 27 psig, and 6 to 30 psig. The following table shows the meaning of different signal pressures are they relate to the level transmitter's output:

| **Transmitter air signal pressure** | **Steam drum water level** |
|---|---|
| 3 psig | 0% (Empty) |
| 6 psig | 25% |
| 9 psig | 50% |
| 12 psig | 75% |
| 15 psig | 100% (Full) |

Likewise, the controller's pneumatic output signal to the control valve uses the same 3 to 15 psig standard to command different valve positions:

| **Controller output signal pressure** | **Control valve position** |
|---|---|
| 3 psig | 0% open (Fully shut) |
| 6 psig | 25% open |
| 9 psig | 50% open |
| 12 psig | 75% open |
| 15 psig | 100% (Fully open) |

It should be noted the previously shown transmitter calibration table assumes the transmitter measures the *full range* of water level possible in the drum. Usually, this is not the case. Instead, the transmitter will be calibrated so it only senses a narrow range of water level near the middle

of the drum. Thus, 3 psig (0%) will not represent an empty drum, and neither will 15 psig (100%) represent a completely full drum. Calibrating the transmitter like this helps avoid the possibility of actually running the drum completely empty or completely full in the case of an operator incorrectly setting the setpoint value near either extreme end of the measurement scale.

An example table showing this kind of realistic transmitter calibration is shown here:

| Transmitter air signal pressure | Actual steam drum water level |
|---|---|
| 3 psig | 40% |
| 6 psig | 45% |
| 9 psig | 50% |
| 12 psig | 55% |
| 15 psig | 60% |

## 1.2 Example: wastewater disinfection

The final step in treating wastewater before releasing it into the natural environment is to kill any harmful bacteria in it. This is called *disinfection*, and chlorine gas is a very effective disinfecting agent. However, just as it is bad to mix too little chlorine in the outgoing water (effluent) because it may not disinfect the water thoroughly enough, there is also danger of injecting too much chlorine in the effluent which could begin poisoning animals and beneficial micro-organisms in the natural environment.

To ensure the right amount of chlorine injection, a dissolved chlorine analyzer can be used to measure the chlorine concentration in the effluent, and a controller can automatically adjust the chlorine control valve to inject the right amount of chlorine at all times based on the process measurement. Figure 1.3 is a Process and Instrument Diagram (P&ID) that shows how such a control system might look.

Figure 1.3: Example Process and Instrument Diagram (P&ID)

Chlorine gas coming through the control valve mixes with the incoming water (influent), then has time to disinfect in the contact chamber before exiting out to the environment.

The transmitter is labeled "AT" (Analytical Transmitter) because its function is to *analyze* the concentration of chlorine dissolved in the water and *transmit* this information to the control system. The "$Cl_2$" (chemical notation for a chlorine molecule) written near the transmitter bubble declares this to be a chlorine analyzer. The dashed line coming out of the transmitter denotes the signal is electronic in nature, not pneumatic as was the case in the previous (boiler control system) example. The most common and likely standard for electronic signaling in industry is 4 to 20 milliamps DC, which represents chlorine concentration in much the same way as the 3 to 15 psig pneumatic signal standard represented steam drum water level in the previous system:

| **Transmitter signal current** | **Chlorine concentration** |
|---|---|
| 4 mA | 0% (no chlorine) |
| 8 mA | 25% |
| 12 mA | 50% |
| 16 mA | 75% |
| 20 mA | 100% (Full concentration) |

The controller is labeled "AIC" because it is an Analytical Indicating Controller. Controllers are always designated by the process variable they are charged with controlling, in this case the chlorine analysis of the effluent. "Indicating" means there is some form of display that a human operator or technician can read showing the chlorine concentration. "SP" refers to the setpoint value entered by the operator, which the controller tries to maintain by adjusting the position of the chlorine injection valve.

A dashed line going from the controller to the valve indicates another electronic signal, most likely 4 to 20 mA DC again. Just as with the 3 to 15 psig pneumatic signal standard in the pneumatic boiler control system, the amount of electric current in this signal path directly relates to a certain valve position:

| **Controller output signal current** | **Control valve position** |
|---|---|
| 4 mA | 0% open (Fully shut) |
| 8 mA | 25% open |
| 12 mA | 50% open |
| 16 mA | 75% open |
| 20 mA | 100% (Fully open) |

Note: it is possible, and in some cases even preferable, to have either a transmitter or a control valve that responds in reverse fashion to an instrument signal such as 3 to 15 psig or 4 to 20 milliamps. For example, this valve could have been set up to be wide open at 4 mA and fully shut at 20 mA. The main point to recognize here is that both the process variable sensed by the transmitter and the position of the control valve are proportionately represented by analog signals.

The letter "M" inside the control valve bubble denotes a motor-actuated valve. Instead of using compressed air pushing against a spring-loaded diaphragm as was the case in the boiler control system, this valve is actuated by an electric motor turning a gear-reduction mechanism. The gear reduction mechanism allows slow motion of the control valve stem even though the motor spins at a fast rate. A special electronic control circuit inside the valve actuator modulates electric power

to the electric motor in order to ensure the valve position accurately matches the signal sent by the controller. In effect, this is another control system in itself, controlling valve position according to a "setpoint" signal sent by another device (in this case, the AIT controller which is telling the valve what position to go to).

## 1.3 Example: chemical reactor temperature control

There is often a mix of instrument signal standards in one control system. Such is the case for this particular chemical reactor temperature control system, where three different signal standards are used to convey information between the instruments. The P&ID in Figure 1.4 shows the interrelationships of the process piping, vessels, and instruments.

Figure 1.4: Chemical reactor temperature control system

The purpose of this control system is to ensure the chemical solution inside the reactor vessel is maintained at a constant temperature. A steam-heated "jacket" envelops the reactor vessel, transferring heat from the steam into the chemical solution inside. The control system maintains a constant temperature by measuring the temperature of the reactor vessel, and throttling steam from a boiler to the steam jacket to add more or less heat as needed.

Start as usual with the temperature transmitter, located near the bottom of the vessel. Note

the different line type used to connect the temperature transmitter (TT) with the temperature-indicating controller (TIC): solid dots with lines in between. This signifies a *digital electronic instrument signal* – sometimes referred to as a *fieldbus* – rather than an analog type (such as 4 to 20 mA or 3 to 15 psig). The transmitter in this system is actually a computer, and so is the controller. The transmitter reports the process variable (reactor temperature) to the controller using digital bits of information. Here there is no analog scale of 4 to 20 milliamps, but rather electric voltage/current pulses representing the 0 and 1 states of binary data.

Digital instrument signals are not only capable of transferring simple process data, but they can also convey device status information (such as self-diagnostic test results). In other words, the digital signal coming from this transmitter not only tells the controller how hot the reactor is, but it can also tell the controller how well the transmitter is functioning!

The dashed line exiting the controller shows it to be analog electronic: most likely 4 to 20 milliamps DC. This electronic signal does not go directly to the control valve, however. It passes through a device labeled "TY", which is a *transducer* to convert the 4 to 20 mA electronic signal into a 3 to 15 psig pneumatic signal which then actuates the valve. In essence, this signal transducer acts as an electrically-controlled air pressure regulator, taking the supply air pressure (usually 20 to 25 psig) and regulating it down to a level commanded by the controller's electronic output signal.

At the temperature control valve (TV) the 3 to 15 psig pneumatic pressure signal applies a force on a diaphragm to move the valve mechanism against the restraining force of a large spring. The construction and operation of this valve is the same as for the feedwater valve in the pneumatic boiler water control system.

## 1.4 Other types of instruments

The three functions present in an automated system sense, control, and influence process variables. Transmitters, controllers, and control valves are respective examples of equipment that performs each process. However, other instruments exist to perform useful functions.

### 1.4.1 Indicators

One common "auxiliary" instrument is the *indicator*, the purpose of which is to provide a human-readable indication of an instrument signal. Quite often process transmitters are not equipped with readouts for whatever variable they measure: they just transmit a standard instrument signal (3 to 15 psig, 4 to 20 mA, etc.) to another device. An indicator gives a human operator a convenient way of seeing what the output of the transmitter is without having to connect test equipment (pressure gauge for 3-15 psig, ammeter for 4-20 mA) and perform conversion calculations. Moreover, indicators may be located far from their respective transmitters, providing readouts in locations more convenient than the location of the transmitter itself. An example where remote indication would be practical is shown in Figure 1.5.

Figure 1.5: Nuclear reactor temperature measurement system

No human can survive near the nuclear reactor when it is in full-power operation, due to the strong radiation flux it emits. The temperature transmitter is built to withstand the radiation, though, and it transmits a 4 to 20 milliamp electronic signal to an indicating recorder located outside of the containment building where it is safe for a human operator. Multiple indicators, at multiple locations, may be connected to the same 4 to 20 milliamp signal wires coming from the temperature transmitter. This allows the display of the reactor temperature in multiple locations (the power supply provides the only limitation on how far a DC milliamp signal can be sustained along copper wires).

A numerical and bargraph panel-mounted indicator appears in Figure 1.6.

Figure 1.6: Panel-mounted indicator

This particular indicator, manufactured by Weschler, shows the position of a flow-control gate in a wastewater treatment facility, both by numerical value (98.06%) and by the height of a bargraph (very near full open – 100%).

A less sophisticated style of panel-mounted indicator shows only a numeric display, such as this Red Lion Controls model shown in Figure 1.7.

Figure 1.7: Panel-mounted numeric indicator

Indicators may also be used in "field" (process) areas to provide direct indication of measured variables if the transmitter device lacks a human-readable indicator of its own. Figure 1.8 shows a Rosemount brand field-mounted indicator, operating directly from the electrical power available in the 4-20 mA loop.

Figure 1.8: Field-mounted indicator

### 1.4.2 Recorders

Another common "auxiliary" instrument is the *recorder* (sometimes specifically referred to as a *chart recorder* or a *trend recorder*), the purpose of which is to draw a graph of process variable(s) over time. Recorders usually have indications built into them for showing the instantaneous value of the instrument signal(s) simultaneously with the historical values, and for this reason are usually designated as *indicating* recorders. A temperature indicating recorder for the nuclear reactor system shown previously would be designated as a "TIR" accordingly.

A *circular* chart recorder uses a round sheet of paper, rotated beneath a pen moved side-to-side by a servomechanism driven by the instrument signal. Two such chart recorders are shown in Figure 1.9.

Figure 1.9: Chart recorders

Two more chart recorders appear in Figure 1.10, a *strip* chart recorder on the right and a *paperless* chart recorder on the left. The strip chart recorder uses a scroll of paper drawn past one or more lateral-moving pens, while the paperless recorder does away with paper entirely by drawing graphic trend lines on a computer screen.

Figure 1.10: Paperless chart recorder

Recorders are extremely helpful for troubleshooting process control problems. This is especially true when the recorder is configured to record not just the process variable, but also the controller's setpoint and output variables as well. Figure 1.11 is an example of a typical "trend" showing the relationship between process variable, setpoint, and controller output in automatic mode, as graphed by a recorder:

Figure 1.11: Process trend data

Here, the setpoint (SP) appears as a perfectly straight (red) line, the process variable as a slightly bumpy (blue) line, and the controller output as a very bumpy (purple) line. This trend shows that the controller is doing exactly what it should: holding the process variable value close to setpoint, manipulating the final control element as far as necessary to do so. The erratic appearance of the output signal is not really a problem, contrary to most peoples' first impression. The fact that the process variable never deviates significantly from the setpoint indicates the control system is operating quite well. What accounts for the erratic controller output, then? Variations in process load. As other variables in the process vary, the controller is forced to compensate for these variations in order that the process variable does not drift from setpoint. Now, maybe this does indicate a problem somewhere else in the process, but there is certainly no problem in this control system.

Recorders become powerful diagnostic tools when coupled with the controller's manual control mode. A great deal of information can be obtained about a process by placing a controller in "manual" mode and allowing direct human control over the final control element (valve, motor, heater). Figure 1.12 is an example of a trend recording for a process in manual mode, where the process variable response is seen graphed in relation to the controller output as that output is increased and decreased in steps:

Figure 1.12: Process trend data in manual mode

Notice the time delay between when the output signal is "stepped" to a new value and when the process variable responds to the change. This sort of delay is generally not good for a control system. Imagine trying to steer an automobile whose front wheels responded to the input from the steering wheel only after a 5-second delay! This would be a very challenging car to drive, because the steering is grossly delayed. The same problem plagues any industrial control system with a time lag between the final control element and the transmitter. Typical causes of this problem include *transport delay* (where there is a physical delay resulting from transit time of a process medium from the point of control to the point of measurement) and mechanical problems in the final control element.

Figure 1.13 shows another type of problem revealed by a trend recording during manual-mode testing.

Figure 1.13: Process trend used in troubleshooting

Here, the process quickly responding to all step-changes in controller output except for those involving a change in direction. This problem is usually caused by mechanical friction in the final control element (e.g. sticky valve stem packing in a pneumatically-actuated control valve), and is analogous to "loose" steering in an automobile, where the driver must turn the steering wheel a little bit extra after reversing steering direction. Anyone who has ever driven an old farm tractor knows what this phenomenon is like, and how it detrimentally affects one's ability to steer the tractor in a straight line.

### 1.4.3 Process switches and alarms

Another type of instrument commonly seen in measurement and control systems is the *process switch*. The purpose of a switch is to turn on and off with varying process conditions. Usually, switches are used to activate alarms to alert human operators to take special action. In other situations, switches are directly used as control devices.

The P&ID in Figure 1.14 of a compressed air control system shows both uses of process switches:

Figure 1.14: Process switches in a compressed air system

The "PSH" (*pressure switch, high*) opens when the air pressure inside the vessel reaches its high control point. The "PSL" (*pressure switch, low*) opens when the air pressure inside the vessel drops down to its low control point. Both switches feed discrete (on/off) electrical signals to a logic control device (signified by the diamond) which then controls the starting and stopping of the electric motor-driven air compressor.

Another switch in this system labeled "PSHH" (*pressure switch, high-high*) opens only if the air pressure inside the vessel exceeds a level beyond the high shut-off point of the high pressure control switch (PSH). If this switch activates, something has gone wrong with the compressor control system, and the high pressure alarm (PAH, or *pressure alarm, high*) activates to notify a human operator.

All three switches in this air compressor control system are directly actuated by the air pressure in the vessel. In other words these are process-sensing switches. It is possible to build switch devices that interpret standardized instrumentation signals such as 3 to 15 psig (pneumatic) or 4 to 20 milliamps (analog electronic), which allows the design of on/off control systems and alarms for any type of process variable measurable with a transmitter.

For example, the chlorine wastewater disinfection system shown earlier may be equipped with a couple of alarm switches to alert an operator if the chlorine concentration ever exceeds predetermined high or low limits, as seen in Figure 1.15.

Figure 1.15: Chlorine concentration alarm switches

The labels "AAL" and "AAH" refer to *analytical alarm low* and *analytical alarm high*, respectively. Since both alarms work off the 4 to 20 milliamp electronic signal output by the chlorine analytical transmitter (AT) rather than directly sensing the process, their construction is greatly simplified. If these were process-sensing switches, each one would have to be equipped with the capability of directly sensing chlorine concentration. In other words, each switch would have to be its own chlorine concentration analyzer, with all the inherent complexity of such a device!

An example of such an alarm module (triggered by a 4-20 mA current signal) is the Moore Industries model SPA ("Site Programmable Alarm"), shown in Figure 1.16.

Figure 1.16: An alarm module

In addition to providing alarm capability, this SPA module also provides a digital display (an small LCD screen) to show the analog signal value for operations or diagnostic purposes.

Like all current-operated alarm modules, the Moore Industries SPA may be configured to "trip" electrical contacts when the current signal reaches a variety of different programmed thresholds. Some of the alarm types provided by this unit include high process, low process, out-of-range, and high rate-of-change.

In a similar manner, pressure-actuated process alarm switches to pneumatic (3-15 psig) signal lines coming from pneumatic transmitters may be used to provide alarm capability to a system designed for continuous measurement. For example, if high- and low-alarm capability were desired for the steam drum water level process described earlier in this chapter, a pair of pressure-actuated switches could be added to the pneumatic level transmitter's 3-15 psig output signal line, as shown Figure 1.17.

Figure 1.17: Pneumatic alarm switches

These two pressure-actuated switches serve as *water level* alarms, because the air pressure signal actuating them comes from the pneumatic *level* transmitter, which outputs an air pressure signal in direct proportion to water level in the boiler's steam drum. Even though the physical stimulus actuating each switch is an air *pressure*, the switches still serve the purpose of liquid *level* alarm signaling because that air pressure is an analogue (representation) of water level in the steam drum.

The alternative to using pressure-actuated switches to signal water level alarm status would be to use independent level-sensing switches attached directly to the steam drum, each switch equipped with its own method[1] of sensing water level, as shown in Figure 1.18.

[1] These might be float-driven switches, where each switch is mechanically actuated by the buoyancy of a hollow metal float resting on the surface of the water. Another technology uses metal electrodes inserted into the water from above, sensing water level by electrical conductivity: when the water level reaches the probe's tip, an electrical circuit is closed. For more information on liquid level switches, refer to section 12.10 beginning on page 465.

Figure 1.18: Level switches

It should be mentioned that the choice between using process alarm switches directly actuated by the process versus alarm switches actuated by the analog signal coming from a transmitter is not arbitrary. In the system where the two alarm switches actuate from the transmitter's 3-15 psig output signal, the integrity of the water level control *and* that of the high- and low-level alarms all depend on the proper function of one transmitter. If that one transmitter were to fail, all three system functions would be compromised. This places a lot of importance on a single instrument, which may not be desirable from the perspective of reliability and process safety. In the system where each level alarm switch independently sensed steam drum water level, one device could fail without compromising either of the other two functions. This independence is desirable because it greatly reduces the probability of "common-cause" failures, where one fault causes multiple functions to stop operating properly.

The final determination should be based on a rigorous analysis of device versus system reliability, which is typically the task of a process engineer.

Process alarm switches may be used to trigger a special type of indicator device known as an *annunciator*. An annunciator is an array of indicator lights and associated circuitry designed to secure a human operator's attention[2] by blinking and sounding an audible buzzer when a process switch actuates into an abnormal state. The alarm state may be then "acknowledged" by an operator pushing a button, causing the alarm light to remain on (solid) rather than blink, and silencing the buzzer. The indicator light does not turn off until the actual alarm condition (the process switch) has returned to its regular state.

Figure 1.19 shows an annunciator located on a control panel for a large engine-driven pump. Each white plastic square with writing on it is a translucent pane covering a small light bulb. When an alarm condition occurs, the respective light bulb flashes, causing the translucent white plastic to

[2]D.A. Strobhar, writing in *The Instrument Engineers' Handbook* on the subject of alarm management, makes the interesting observation that alarms are the only form of instrument "whose sole purpose is to alter the operator's behavior." Other instrument devices may automatically respond to process changes by directly influencing the process, but only alarms work to *control the operator*.

glow, highlighting to the operator which alarm is active.

Figure 1.19: An alarm annunciator

Note the two pushbutton switches below labeled "Test" and "Acknowledge." Pressing the "Acknowledge" button will silence the audible buzzer and also turn any blinking alarm light into a steady (solid) alarm light until the alarm condition clears, at which time the light turns off completely. Pressing the "Test" button turns all alarm lights on, to ensure all light bulbs are still functional.

Opening the front panel of this annunciator reveals modular relay units controlling the blinking and acknowledgment latch functions, one for each alarm light. This is shown in Figure 1.20.

Figure 1.20: Internals of an alarm annunciator

This modular design allows each alarm channel to be serviced without necessarily interrupting the function of all others in the annunciator panel.

A simple logic gate circuit in Figure 1.21 illustrates the acknowledgment latching feature (here implemented by an S-R latch circuit) common to all process alarm annunciators.

Figure 1.21: Logic of an alarm annunciator

Panel-mounted annunciators are becoming a thing of the past, as computer-based alarm displays

replace them with advanced capabilities such as time logging, first-event recording, and multiple layers of ackowledgement/access. Time logging is of particular importance in the process industries, as the sequence of events is often extremely important in investigations following an abnormal operating condition. Knowing *what went wrong, in what order* is much more informative than simply knowing which alarms have tripped.

## 1.5 Summary

I&C engineers are responsible for designing safe and efficient operation of industrial measurement and control systems. As this chapter shows, this requires a broad command of technical skill. Instrumentation and control engineering requires knowledge of physics, chemistry, mathematics, electrical engineering, mechanical engineering, chemical engineering, computer programming, computer networking, communications systems, human factors, and database management. An I&C engineer must understand all these subject areas to some degree, and more importantly how these knowledge areas relate to each other.

The all-inclusiveness of this profession makes it very challenging and interesting. Adding to the challenge is the continual introduction of new technologies. The advent of new technologies, however, does not necessarily relegate legacy technologies to the scrap heap. It is quite common to find state-of-the-art instruments in the very same facility as decades-old instruments; digital fieldbus networks running alongside 3 to 15 psig pneumatic signal tubes; microprocessor-based sensors mounted right next to old mercury tilt-switches. Thus, the competent I&C engineer must be comfortable working with both old and new technologies, understanding the relative merits and weaknesses of each.

The most important skill for an I&C engineer is the ability to learn independently. It is impossible to fully prepare for a career in the I&C discipline with any amount of schooling. The profession is so broad and the responsibility so great, and the landscape so continuously and rapidly changing, that life-long learning for the engineer is a matter of professional survival.

# References

[1] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1.

# Chapter 2

# Instrumentation and Control Documentation

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Read and produce the following documentation:
   (a) Process Flow Diagrams (PFD)
   (b) Piping and Instrumentation Diagrams (P&ID)
   (c) Instrument loop diagrams
   (d) Installation detail drawings
   (e) Instrument location drawings
   (f) Instrument specification sheets
   (g) Instrument index
   (h) Control system narratives
   (i) Control system architecture drawings
   (j) Control system specifications
2. Identify the symbols used for:
   (a) Types of valves
   (b) Types of valve operators
   (c) Types of instrumentation
   (d) Types of instrument signal controllers and modifiers
   (e) Types of system components (pumps, etc.)
   (f) Types and sizes of piping
3. Identify the symbols used to denote the location of instruments, indicators, and controllers.
4. Identify how valve conditions are depicted.
5. Determine the system flowpath for a given valve lineup.

Every technical discipline has its own standardized way(s) of making diagrams and associated design documentation, and I&C is no exception. The scope of instrumentation is so broad, however, that no one form of document is sufficient to capture all required design information. This chapter will discuss five different types of instrumentation diagrams:

- Process Flow Diagrams (PFDs)
- Process and Instrument Diagrams (P&IDs)
- Loop diagrams ("loop sheets")
- Installation detail diagrams
- Instrument location drawings

This chapter will also address other documents produced for I&C system design:

- Instrument specification sheets
- The instrument index
- Control system narratives

At the highest level, the I&C engineer is interested in the interconnections of process vessels, pipes, and flow paths of process fluids. The proper form of diagram to represent the "big picture" of a process is called a *process flow diagram* (PFD). Individual instruments are sparsely represented in a PFD, because the focus of the diagram is the process itself. This document is typically produced by a process, chemical, or mechanical engineer, not an I&C engineer, However, it informs the I&C engineer as to the values of typical process state variables such as flow, level, temperature, and pressure at various points in the system. It also typically indicates which variables are measured and used in control loops in a rough format.

At the lowest level, the I&C engineer is interested in the interconnections of individual instruments, including all the wire numbers, terminal numbers, cable types, instrument calibration ranges, etc. The proper form of diagram for this level of fine detail is called a *loop diagram*. Here, the process vessels and piping are sparsely represented, because the focus of the diagram is the instruments themselves.

*Process and instrument diagrams* (P&IDs) lie somewhere in the middle between process flow diagrams and loop diagrams. A P&ID shows the layout of all relevant process vessels, pipes, and machinery, but with instruments superimposed on the diagram showing what gets measured and what gets controlled. Here, one can view the flow of the process as well as the "flow" of information between instruments measuring and controlling the process. P&IDs are developed by the collaboration of the process or mechanical engineer and the I&C engineer.

An I&C professional must often switch between different diagrams when troubleshooting a complex control system. There is simply too much detail for any one diagram to show everything. Even if the page were large enough, a "show everything" diagram would be so chock-full of details that it would be difficult to follow any one line of details. The narrowing of scope with the progression from PFD to loop diagram may be visualized as a process of "zooming in," as though one were viewing a process through the lens of a microscope at different powers. First, begin with a PFD or P&ID to get an overview of the process, to see how the major components interact. Then, once the instrument "loop" has been identified, go to the appropriate loop diagram to see the interconnection details of that instrument system to find where to connect test equipment and what signals are expected at the test locations.

Another analogy for this progression of documents is a map, or more precisely, a globe, an atlas, and a city street map. The globe gives the "big picture" of the Earth, countries, and major cities. An atlas allows "zooms in" to see details of particular provinces, states, and principalities, and the

routes of travel connecting them all. A city map shows major and minor roads, canals, alleyways, and perhaps even some addresses. It would be impractical to have a globe large enough to show all the details of every city! Furthermore, a globe comprehensive enough to show all these details would have to be updated *very* frequently to keep up with all cities' road changes. There is a certain economy inherent to the omission of fine details, both in ease of use and in ease of maintenance.

## 2.1 I&C Drawings

### 2.1.1 Process Flow Diagrams

To show a practical process example, let's examine three diagrams for a compressor control system. In this fictitious process, water is being evaporated from a process solution under partial vacuum (provided by the compressor). The compressor then transports the vapors to a "knockout drum" where some of them condense into liquid form. As a typical PFD, this diagram shows the major interconnections of process vessels and equipment, but omits details such as instrument signal lines and auxiliary instruments:

Figure 2.1: A Process Flow Diagram (PFD)

One might guess the instrument interconnections based on the labels of the instruments. For instance, a good guess would be that the level transmitter (LT) on the bottom of the knockout drum might send the signal that eventually controls the level valve (LV) on the bottom of that same vessel. One might also guess that the temperature transmitter (TT) on the top of the evaporator might be part of the temperature control system that lets steam into the heating jacket of that vessel.

Based on this diagram alone, one would be hard-pressed to determine what control system, if any, controls the compressor itself. All the PFD shows relating directly to the compressor is a flow transmitter (FT) on the suction line. This level of uncertainty is perfectly acceptable for a PFD, because its purpose is merely to show the general flow of the process itself, and only a bare minimum of control instrumentation.

### 2.1.2 Process and Instrument Diagrams

The next level of detail is the Process and Instrument Diagram[1], or P&ID. Here, there is a "zooming in" of scope from the whole evaporator process to the compressor as a unit. The evaporator and knockout vessels almost fade into the background, with their associated instruments absent from view:

The International Society of Automation (ISA) does not publish a standard that defines the information shown on a P&ID, but ISA Standard 5.1, *Instrumentation Symbols and Identification* does define the symbols used [1]. The only available standard for P&IDs is produced by the Process Industry Practices (PIP) organization, PIC001-2008, *Piping and Instrumentation Diagram Documentation Criteria* [2]. ISA is currently developing a standard based on the PIP standard.

Figure 2.2: A P&ID shows increased detail over a PFD

[1]Sometimes P&ID stands for *Piping* and Instrument Diagram. Either way, it means the same thing.

It can be seen that there is more instrumentation associated with the compressor than just a flow transmitter. There is also a differential pressure transmitter (PDT), a flow indicating controller (FIC), and a "recycle" control valve that allows some of the vapor coming out of the compressor's discharge line to go back around into the compressor's suction line. Additionally, there are have a pair of temperature transmitters that report suction and discharge line temperatures to an indicating recorder.

Some other noteworthy details emerge in the P&ID as well. Note that the flow transmitter, flow controller, pressure transmitter, and flow valve all bear a common number: 42. This common "loop number" indicates these four instruments are all part of the same control loop. An instrument with any other loop number is part of a different control loop, measuring and/or controlling some other function in the process. Examples of this include the two temperature transmitters and their respective recorders, bearing the loop numbers 41 and 43.

Please note the differences in the instrument "bubbles" as shown on this P&ID. Some of the bubbles are just open circles, where others have lines going through the middle. Each of these symbols has meaning defined by ISA Standard 5.1:

Figure 2.3: Several standard instrument bubble symbols

The type of bubble used for each instrument tells us something about its location. This, obviously, is quite important when working in a facility with many thousands of instruments scattered over acres of facility area, structures, and buildings.

The rectangular box enclosing both temperature recorders shows they are part of the same physical instrument. In other words, this indicates there is really only one temperature recorder instrument, and that it plots both suction and discharge temperatures (most likely on the same trend graph). This suggests that each bubble may not necessarily represent a discrete, physical instrument, but rather an instrument *function* that may reside in a multi-function device.

### 2.1.3 Loop diagrams

Loop diagrams are essentially electrical installation details. P&IDs do not include cable types, wire numbers, terminal blocks, junction boxes, instrument calibration ranges, failure modes, power sources, and the like. This level of detail is shown on the loop diagram. Loop diagrams are defined by International Society of Automation (ISA) Standard 5.4, *Instrument Loop Diagrams* [3]. The

loop diagram (sometimes called a *loop sheet*) for the compressor surge control system (loop number 42) is shown below:

| Tag number | Description | Input cal. | Output cal. | Notes |
|---|---|---|---|---|
| FE 42 | Venturi tube | 0-1500 SCFM | 0-100 "WC | |
| FT 42 | Suction flow transmitter | 0-100 "WC | 4-20 mA | |
| FY 42a | Square root extractor | 4-20 mA | 4-20 mA | |
| FY 42b | Current-to-pressure converter | 4-20 mA | 3-15 PSI | |
| FV 42 | Anti-surge control valve | 3-15 PSI | 100%-0% | Air-to-close |
| PDT 42 | Differential pressure transmitter | 0-200 PSI | 20-4 mA | Reverse action |
| FIC 42 | Anti-surge controller | 4-20 mA | 4-20 mA | |

Figure 2.4: A loop diagram shows increased detail over a P&ID

Comparison of the compressor surge control loop diagram to the compressor P&ID reveals that the P&ID didn't show all of the instruments in this control "loop." Not only are there two transmitters, a controller, and a valve; there are also two signal transducers. Transducer 42a modifies the flow transmitter's signal before it goes into the controller, and transducer 42b converts the electronic 4 to 20 mA signal into a pneumatic 3 to 15 psig air pressure signal. Each instrument "bubble" in a loop diagram represents an individual device, with its own terminals for connecting wires.

Note that dashed lines now represent individual copper wires instead of whole cables. Terminal blocks where these wires connect to are represented by squares with numbers in them. Cable numbers, wire colors, junction block numbers, panel identification, and even grounding points are all shown in loop diagrams. The only type of diagram at a lower level of abstraction than a loop diagram would be an electronic schematic diagram for an individual instrument, which of course would only show details pertaining to that one instrument. Thus, the loop diagram is the most detailed form of diagram for an I&C system as a whole, and thus it must contain all details omitted by PFDs and P&IDs alike.

To the novice it may seem excessive to include such trivia as wire colors in a loop diagram. To

the experienced I&C professional who has had to work on systems lacking such documented detail, this information is highly valued. The more detail shown on a loop diagram, the easier it makes the inevitable job of maintaining, updating, or demolishing that system at some later date. When a loop diagram shows exactly what wire color to expect at exactly what point in an instrumentation system, and exactly what terminal that wire should connect to, it becomes much easier to proceed with any troubleshooting, calibration, or upgrade task.

An interesting detail seen on this loop diagram is an entry specifying "input calibration" and "output calibration" for each and every instrument in the system. This is actually a very important concept to keep in mind when troubleshooting a complex instrumentation system: every instrument has at least one input and at least one output, with some sort of mathematical relationship between the two. Diagnosing where a problem lies within a measurement or control system often reduces to testing various instruments to see if their output responses appropriately match their input conditions.

For example, one way to test the flow transmitter in this system would be to subject it to a number of different pressures within its range (specified in the diagram as 0 to 100 inches of water column differential) and seeing whether or not the current signal output by the transmitter was linearly proportional to the applied pressure (e.g., 4 mA at 0 inches pressure, 20 mA at 100 inches pressure, 12 mA at 50 inches pressure, etc.).

Given the fact that a calibration error or malfunction in any one of these instruments can cause a problem for the control system as a whole, it is nice to know there is a way to determine which instrument is to blame and which instruments are not. This general principle holds true regardless of the instrument's type or technology. The same input-versus-output test procedure can be used to verify the proper operation of a pneumatic (3 to 15 psig) level transmitter or an analog electronic (4 to 20 mA) flow transmitter or a digital (fieldbus) temperature transmitter alike. Each and every instrument has an input and an output, and there is always a predictable (and testable) correlation from one to the other.

Another interesting detail seen on this loop diagram is the *action* of each instrument. Notice the box and arrow (pointing either up or down) next to each instrument bubble. An "up" arrow (↑) represents a *direct-acting* instrument: one whose output signal increases as the input stimulus increases. A "down" arrow (↓) represents a *reverse-acting* instrument: one whose output signal decreases as the input stimulus increases. All the instruments in this loop are direct-acting with the exception of the pressure differential transmitter PDT-42:

Figure 2.5: A reverse-acting instrument bubble from a loop diagram

Here, the "down" arrow tells us the transmitter will output a full-range signal (20 mA) when it senses zero differential pressure, and a 0% signal (4 mA) when sensing a full 200 psig differential. While this calibration may seem confusing and unwarranted, it serves a definite purpose in this

particular control system. Since the transmitter's current signal decreases as pressure increases, and the controller must be correspondingly configured, a decreasing current signal will be interpreted by the controller as a high differential pressure. If any wire connection fails in the 4-20 mA current loop for that transmitter, the resulting 0 mA signal will be naturally "seen" by the controller as a pressure over-range condition. This is considered dangerous in a compressor system because it predicts a condition of surge. Thus, the controller will naturally take action to prevent surge by commanding the anti-surge control valve to open, because it "thinks" the compressor is about to surge. In other words, the transmitter is intentionally calibrated to be reverse-acting such that any break in the signal wiring will naturally bring the system to its safest condition.

### 2.1.4 Mechanical installation details

Mechanical installation details are used to show how the instrument is connected to the process (see Appendix A for instrument connection information). These details call out such information as the location on a pipe where the instrument line is to attach, what fittings are used, whether or not a shut-off (root) valve is provided, how instrument lines are sloped, elevation of an instrument relative to a pipe, distance of straight pipe lengths required for the instrument to provide the best measurement, whether electric or steam heating of instrument lines is required, etc. They usually also contain a small bill of material (BOM) to call out the required components. Additionally, the installation detail often includes a definition of responsibilities indicating which discipline is responsible for various parts of the drawing, e.g., process or instrumentation. The Process Industry Practices (PIP) organization is a consortium of companies that has produced a set of industry-standard guides for engineering design and construction. These guides are known as "practices." Several of the practices provide mechanical instrument installation details for measuring parameters such as flow, pressure, temperature, and level.

### 2.1.5 Location Drawings

#### Plan Drawings

Plan drawings are views looking down on the floor plan of a facility showing the location of equipment relative to the walls of the building. The locations of instruments are shown by instrument bubbles and often accompanied by an indication of elevation. Other items that may appear on plan drawings are control panels, operator interface stations, and conduit routing.

#### Elevation Drawings

Elevation drawings are views looking at the side of a building or piece of equipment relative to floor or ground level. Elevations are often created to show the locations of instruments in vertical space, especially on equipment that extends through multiple stories of a facility. For example, a distillation column may have multiple trays at different heights, and temperature may be measured at certain trays. It is helpful to be able to see an elevation view so that the instruments are installed at the proper vertical locations.

### 2.1.6 Control System Architecture Drawings

*A control system architecture drawing* is a block diagram that shows all of the control cabinets, operator workstations, human-machine interfaces, printers, etc. in the control system and the specific network connections between them.

## 2.2 Instrument and process equipment symbols

This section shows some of the many instrument symbols found in different types of technical diagrams used to document instrument systems.

### 2.2.1 Line types

| No. | Symbol | Application |
|---|---|---|
| 1 | IA | • IA may be replaced by PA [plant air], NS [nitrogen}, or GS [any gas supply].<br>• Indicate supply pressure as required, e.g. PA-70kPa, NS-150psig, etc. |
| 2 | ES | • Instrument electric power supply.<br>• Indicate voltage and type as required, e.g. ES-220VAC.<br>• ES may be replaced by 24VDC or 120VAC. |
| 3 | HS | • Instrument hydraulic power supply.<br>• Indicate pressure as required, e.g. HS-70psig. |
| 4 | | • Undefined signal.<br>• Use for Process Flow Diagrams.<br>• Use for discussions or diagrams where type of signal is not of concern. |
| 5 | | • Pneumatic signal, continuously variable or binary. |
| 6 | | • Electronic or electrical continuously variable or binary signal.<br>• Functional diagram binary signal. |
| 7 | | • Functional diagram continuously variable signal.<br>• Electrical schematic ladder diagram signal and power rails. |
| 8 | | • Hydraulic signal. |
| 9 | | • Filled thermal element capillary tube.<br>• Filled sensing line between pressure seal and instrument. |
| 10 | | • Guided electromagnetic signal.<br>• Guided sonic signal.<br>• Fiber optic cable. |
| 11 | a)<br>b) | • Unguided electromagnetic signals, light, radiation, radio, sound, wireless etc.<br>• Wireless instrumentation signal.<br>• Wireless communication link. |
| 12 | | • Communication link and/or system bus between devices and functions of a shared display, shared control system.<br>• DCS, PLC, or PC communication link and system bus. |
| 13 | | • Communication link or bus connecting two or more independent microprocessor-or computer based systems.<br>• DCS to DCS, DCS to PLC, PLC to PC, and DCS to Fieldbus, etc. connections. |
| 14 | | • Communication link and system bus, between devices and functions of a fieldbus system.<br>• Link from and to “intelligent” devices. |
| 15 | | • Communication link between a device and a remote calibration adjustment device or system.<br>• Link from and to “smart” devices. |
| 16 | | • Mechanical link or connection. |

Figure 2.6: Instrument diagram line types

### 2.2.2 Process/Instrument line connections

| No. | Symbol | Application |
|---|---|---|
| 1 | | • Instrument connections to process and equipment.<br>• Process impulse lines.<br>• Analyzer sample lines. |
| 2 | (ST) | • Heat [cool] traced impulse or sample line from process.<br>• Type of tracing indicated by: [ET] electrical, [ST] steam, [CW] chilled water, etc. |
| 3 | | • Generic instrument connection to process line.<br>• Generic instrument connection to equipment. |
| 4 | | • Heat [cool] traced generic instrument impulse line.<br>• Process line or equipment may or may not be traced. |
| 5 | | • Heat [cool] traced instrument.<br>• Instrument impulse line may or may not be traced. |
| 6 | | • Flanged instrument connection to process line.<br>• Flanged instrument connection to equipment. |
| 7 | | • Threaded instrument connection to process line.<br>• Threaded instrument connection to equipment. |
| 8 | | • Socket welded instrument connection to process line.<br>• Socket welded instrument connection to equipment. |
| 9 | | • Welded instrument connection to process line.<br>• Welded instrument connection to equipment. |

Figure 2.7: Instrument connection types

### 2.2.3 Instrument bubbles

| No. | Shared display, Shared control | | C | D | Location & accessibility |
|---|---|---|---|---|---|
| | A | B | | | |
| | Primary Choice or Basic Process Control System | Alternate Choice or Safety Instrumented System | Computer Systems and Software | Discrete | |
| 1 | | | | | • Located in field.<br>• Not panel, cabinet, or console mounted.<br>• Visible at field location.<br>• Normally operator accessible. |
| 2 | | | | | • Located in or on front of central or main panel or console.<br>• Visible on front of panel or on video display.<br>• Normally operator accessible at panel front or console. |
| 3 | | | | | • Located in rear of central or main panel.<br>• Located in cabinet behind panel.<br>• Not visible on front of panel or on video display.<br>• Not normally operator accessible at panel or console. |
| 4 | | | | | • Located in or on front of secondary or local panel or console.<br>• Visible on front of panel or on video display.<br>• Normally operator accessible at panel front or console. |
| 5 | | | | | • Located in rear of secondary or local panel.<br>• Located in field cabinet.<br>• Not visible on front of panel or on video display.<br>• Not normally operator accessible at panel or console. |

Figure 2.8: Instrument bubble symbols

### 2.2.4 Process valve types

*Valve status may or may not be shown in a process diagram. If you happen to see solid-colored valve symbols anywhere in a diagram, you know that status is being represented. If you see no solid-colored valves anywhere in the diagram, either all valves are shown open or else status is not represented at all.*

Figure 2.9: Valve symbols

### 2.2.5 Valve actuator types

Figure 2.10: Valve actuator symbols

### 2.2.6 Valve failure mode

Figure 2.11: Valve failure symbols

### 2.2.7 Flow measurement devices (flowing left-to-right)

Figure 2.12: Flow measurement symbols

### 2.2.8 Process equipment

Figure 2.13: Process equipment symbols

### 2.2.9 Single-line electrical diagram symbols

Figure 2.14: Electrical one-line symbols

Figure 2.15: Electrical one-line symbols (continued)

### 2.2.10 Instrument identification tags

Up until this point, various types of instrumentation diagrams have been discussed, each one making reference to different instruments by lettered identifiers such as TT (Temperature Transmitter), PDT (Pressure Differential Transmitter), or FV (Flow Valve), without formally defining all the letters used to identify instruments. Part of the ISA 5.1 standard does exactly this, which is described below.

Each instrument within an instrumented facility must have its own unique identifying *tag* consisting of a series of letters describing that instrument's *function*, as well as a number identifying the particular *loop* to which it belongs. An optional numerical prefix typically designates the larger area of the facility in which the loop resides, and an optional alphabetical suffix designates multiple instances of instruments within one loop.

For example, if an instrument bears the tag `FC-135`, it is a *flow controller* (FC) for loop number *135*. In a large manufacturing facility with multiple processing "unit" areas, a tag such as this might be preceded by another number designating the unit area. For example, the hypothetical flow controller might be labeled 12-FC-135 (flow controller for loop #135, located in unit 12). This type of longer tag is always unique, while the shorter tag is often used for brevity and may be the same for identical units since the unit number is omitted. For example, if there were twelve identical processing units, there might only be one drawing associated with them. This drawing would contain short tags only, but it would have a note indicating that the tags shown are abbreviated and that the full tag for each instrument includes a prefix with the corresponding unit number. (this reduces the drawing effort and the related cost to the client). If this loop happened to contain multiple controllers, they would need to be distinguished them from each other by the use of suffix letters appended to the loop number (e.g., 12-FC-135A, 12-FC-135B, 12-FC-135C).

Each and every instrument within a particular loop is first defined by the variable that loop seeks to sense or control, regardless of the physical construction of the instrument itself. The hypothetical flow controller FC-135, for example, may be physically identical to the level controller in loop #72

(LC-72), or to the temperature controller in loop #288 (TC-288). What makes FC-135 a *flow* controller is the fact that the transmitter sensing the main process variable measures *flow*. Likewise, the identifying tag for every other instrument within that loop[2] must begin with the letter "F" as well. This includes the final control element as well: in a level control loop, the transmitter is identified as an "LT" even if the actual sensing element works on *pressure* (because the variable that the loop strives to sense or control is actually level, even if indirectly sensed by pressure), the controller is identified as an "LC", and the control valve throttling fluid *flow* is identified as an "LV": every instrument in that level-controlling loop serves to help control *level*, and so its primary function is to be a "level" instrument.

Valid letters recognized by the ISA for defining the primary process variable of an instrument within a loop are shown in the following table. Please note that the use of a modifier defines a unique variable: for example, a "PT" is a transmitter measuring *pressure* at a single point in a process, whereas a "PDT" is a transmitter measuring a *pressure difference* between two points in a process. Likewise, a "TC" is a controller controlling temperature, whereas a "TKC" is a controller controlling the *rate-of-change of temperature*:

[2]Exceptions do exist to this rule. For example, in a cascade or feedforward loop where multiple transmitters feed into one or more controllers, each transmitter is identified by the type of process variable *it* senses, and each controller's identifying tag follows suit.

| Letter | Variable | Modifier |
|---|---|---|
| A | Analysis (composition) | |
| B | Burner or Combustion | |
| C | *User-defined* | |
| D | *User-defined* | Difference, Differential |
| E | Voltage | |
| F | Flow | Ratio or Fraction |
| G | *User-defined* | |
| H | Hand (manual) | |
| I | Current | |
| J | Power | |
| K | Time or Schedule | Time rate-of-change |
| L | Level | |
| M | *User-defined* | |
| N | *User-defined* | |
| O | *User-defined* | |
| P | Pressure or Vacuum | |
| Q | Quantity | Time-Integral or Total |
| R | Radiation | |
| S | Speed or Frequency | Safety |
| T | Temperature | |
| U | Multi-function | |
| V | Vibration or Mechanical Analysis | |
| W | Weight or Force | |
| X | *Unclassified* | X-axis |
| Y | Event, State, or Presence | Y-axis |
| Z | Position or Dimension | Z-axis or Safety Instrumented System |

A "user-defined" letter represents a non-standard variable used multiple times in an instrumentation system. For example, an engineer designing an instrument system for measuring and controlling the *refractive index* of a liquid might choose to use the letter "C" for this variable. Thus, a refractive-index transmitter would be designated "CT" and a control valve for the refractive-index loop would be designated "CV". The meaning of a user-defined variable need only be defined in one location (e.g., in a legend for the diagram).

An "unclassified" letter represents one or more non-standard variables, each used only once (or a very limited number of times) in an instrumentation system. The meaning of an unclassified variable is best described immediately near the instrument's symbol rather than in a legend.

Succeeding letters in an instrument tag describe the function that instrument performs relative to the process variable. For example, a "PT" is an instrument *transmitting* a signal representing pressure, while a "PI" is an *indicator* for pressure and a "PC" is a *controller* for pressure. Many instruments have multiple functions designated by multiple letters, such as a TRC (Temperature *Recording Controller*). In such cases, the first function letter represents the "passive" function (usually provided to a human operator) while the second letter represents the "active" (automated)

control function. Note: per the ISA 5.1 standard, the second letter "C" should only be used for a device that actively implements the control function. For example, a PV is a valve that controls pressure based on a signal from an external controller, but a PCV is a valve that controls pressure itself without the need for an external controller (i.e., a pressure regulator).

| **Letter** | **Passive function** | **Active function** | **Modifier** |
|---|---|---|---|
| A | Alarm | | |
| B | *User-defined* | *User-defined* | *User-defined* |
| C | | Control | Close |
| D | | | Deviation |
| E | Element (sensing) | | |
| G | Glass, Gauge or Viewport | | |
| H | | | High |
| I | Indicate | | |
| J | Scan | | |
| K | | Control Station | |
| L | Light | | Low |
| M | | | Middle or Intermediate |
| N | *User-defined* | *User-defined* | *User-defined* |
| O | Orifice or Restriction | | Open |
| P | Test point | | |
| Q | Integrate or Totalize | | |
| R | Record | | Run |
| S | | Switch | Stop |
| T | | Transmit | |
| U | Multi-function | Multi-function | Multi-function |
| V | | Valve, Damper, or Louver | |
| W | Well | | |
| X | Accessory Device or *Unclassified* | *Unclassified* | *Unclassified* |
| Y | | Auxiliary Devices | |
| Z | | Driver, Actuator, or unclassified final control element | |

A variety of other letter combinations are often used to identify details not standardized by the ISA. Examples of some instrument tag letter combinations are shown in the following list:

- **AIT** = Analytical Indicating Transmitter *(e.g., an oxygen concentration analyzer with a built-in display of oxygen percentage)*
- **ESL** = Voltage Switch, Low *(e.g., a switch used to detect an under-voltage condition in an electrical power system)*
- **FFI** = Flow Ratio Indicator *(e.g., a device indicating the ratio between air and fuel for a large industrial engine)*
- **FIC** = Flow Indicating Controller *(i.e., a controller designed to indicate flow to a human operator)*

- **HC** = Hand Controller *(i.e., a device allowing a human operator to set a control signal to some desired level, usually to operate a valve or other final control element)*
- **HS** = Hand Switch *(i.e., a switch allowing a human operator to change the operating status of a piece of equipment)*
- **JQR** = Power Totalizing Recorder *(e.g., a watt-hour recorder, tracking total energy used)*
- **LSHH** = Level Switch, High *(e.g., a level-sensing switch designed to detect a high liquid level)*
- **LSHH** = Level Switch, High-High *(e.g., a level-sensing switch designed to detect a higher liquid level than that detected by a high level switch )*
- **LT** = Level Transmitter *(i.e., a device sensing liquid level and reporting that level in some analog or digital form)*
- **PIT** = Pressure Indicating Transmitter *(i.e., pressure transmitter with a built-in display of measured pressure)*
- **PDT** = Pressure Differential Transmitter *(i.e., a pressure transmitter built and installed to sense the difference of pressure between two points in a fluid system)*
- **PV** = Pressure Valve *(i.e., a control valve installed in a loop where the process variable is pressure)*
- **PCV** = Pressure Control Valve *(a control valve installed in a loop where the process variable is pressure and the valve implements the control function itself, e.g., a pressure regulator)*
- **PSV** = Pressure Safety Valve *(a valve installed in a loop where the process variable is pressure and the valve implements a safety function, e.g., a pressure relief)*
- **TE** = Temperature Element *(i.e., a sensing element used to directly detect the temperature of a process material; e.g., a thermocouple, thermistor, filled-bulb, bimetallic spring)*
- **TKAH** = Temperature Rate-of-change Alarm, High *(i.e., a device alarming when the rate of temperature change exceeds a pre-set limit)*
- **TV** = Temperature Valve *(i.e., a control valve installed in a loop where the process variable is temperature)*
- **TY** = Temperature Auxiliary Device *(e.g., an I/P transducer in a temperature loop that converts a current signal to a pressure)*
- **VSH** = Vibration Switch, High *(i.e., a switch used to detect a high level of vibration on a piece of machinery)*
- **ZXI**, **ZYI**, and **ZZI** = Position Indicators for X, Y, and Z axes respectively *(e.g., indicators showing the three axis positions for a CNC machine tool)*

## 2.3 Other I&C Documents

### 2.3.1 Guide Specifications

Construction drawings only carry so much information. *Guide specifications* are written requirements of the the codes and standards, methods, materials, equipment, system performance, and workmanship standards. They accompany the construction drawings and provide essential information required by the contractor so that the job may be constructed in accordance with the design intent.

### 2.3.2 Instrument Specification Sheets

The *instrument specification sheet* (a.k.a. *spec sheet* or *data sheet*) provides the minimum amount of information required to specify an instrument for the application required. These sheets are typically issued for bid to various vendors. Once a vendor has been selected for a given instrument, additional vendor-specific information may be added to a spec sheet prior to issuing it for purchase.

The ISA has produced a set of specification sheets in TR20.00.01, *Specification Forms for Process Measurement and Control Instruments* [4]. The following are generally important questions that get answered on specification sheets:

- Process/Mechanical information:
  - What process materials and conditions are being monitored and what will be the required range of the instrument? Instruments generally work best when the nominal process values are between 1/4 and 3/4 of the range of the instrument.
  - Are the materials of construction compatible with the materials in the process?
  - What type of mechanical connections does the device have? What is the type, size, and rating of the process connection? Does it require threaded, flanged, or socketed fittings? Is there a pilot connection to the process?
  - Are isolation valves needed? What about bypass, vent, or drain valves? Is a valve manifold needed?
  - Does it require hydraulic or pneumatic power? If so, at what pressures and flows?
- Electrical information:
  - What type of electrical connections does the device have?
  - What wire sizes does it accept?
  - What size conduit connection does it have?
  - What voltage is applied? Does it need an external power source (four wire), or is it powered by the signal wiring (two wire, a.k.a. loop powered)?
  - What is the electrical output (e.g., 4-20 mA, 0-10 VDC)?
  - Is an external solenoid valve needed to activate the device?
  - Is the atmosphere in which you are going to install the device hazardous (i.e., flammable)? If so, what type of rating is needed? Explosion-proof? Intrinsically safe?
  - Does the device need to be approved by a national testing laboratory (e.g., most states require electrical equipment to be approved by UL, CSA, ETL, or another third-party approver)?
- Miscellaneous information:
  - What type of environment will it be exposed to (e.g., high ambient temperature or humidity, dry or wet environments, etc.)?
  - How will the device be mounted in the field? Will it clamp to a pipe, be mounted in a duct, get attached to a wall, or mount inside a control panel on DIN rail?
  - Will identification tags or calibration certificates need to be provided?

### 2.3.3 The Instrument Index

An instrument index is used to keep track of the field devices in a project. It is usually either a database (preferred) or a spreadsheet that contains information related to each field device.

An instrument index often includes the following information about each instrument:

- Instrument tag - no duplicates are ever allowed (e.g., LT-101)
- Instrument description (e.g., Boiler 1 Water Level)
- Associated equipment or process number
- P&ID number
- Loop drawing number
- Installation detail drawing number
- Location drawing number
- I/O type (e.g., DI, DO, AI, or AO)
- Controller wiring terminations
- Wire label
- Software address
- Notes about the device

### 2.3.4 Control System Narratives

A *control system narrative*, also called a sequence of operations (SOO), is a written document that describes the detailed workings of each control loop in the system. The SOO provides a basis for the control system integrator to program the controller to acheive the desired functionality. It also acts as a troubleshooting aid to technicians when the system has a fault. An example SOO is included in Appendix B.

# References


[1] Instrumentation Symbols and Identification, International Society of Automation Std. 5.1, 2009.

[2] Piping and Instrumentation Diagram Documentation Criteria, Process Industry Practices Std. PIC001, 2008.

[3] Instrument Loop Diagrams, International Society of Automation Std. 5.4, 1991.

[4] Specification Forms for Process Measurement and Control Instruments, International Society of Automation Std. 20.00.01, 2006.

# Part II

# Instrumentation

# Chapter 3

# Analog Signal Transmission

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Discuss the signal types used in instrumentation loops.
2. Determine the transmitter current value or the process/controller variable given the value of the other.
3. Describe the function and application of the following types of instrument circuits:
   (a) Loop-powered (two-wire)
   (b) Externally-powered (three- or four-wire)
4. Explain the function and application of the following types of accessories:
   (a) Precision dropping resistors
   (b) Signal isolators
   (c) Signal converters
   (d) Signal splitters
5. List and discuss methods used for measuring loop current
6. Troubleshoot current loops by measuring loop current or voltages
7. Use loop calibrators to calibrate instruments
8. State the limits for loop currents set up by NAMUR

## 3.1 Analog electronic communication

An "analog" electronic signal is a voltage or current whose magnitude represents some physical measurement or control quantity. An instrument is often classified as being "analog" simply by virtue of using an analog signal standard to communicate information, even if the internal construction and design of the instrument may be mostly digital in nature. This is to distinguish such instruments from those making use of no analog electronic signals at all (e.g., wireless or Fieldbus instruments).

## 3.2 4 to 20 mA analog current signals

The most popular form of signal transmission used in modern industrial instrumentation systems (as of this writing) is the 4 to 20 milliamp DC standard. This is an *analog* signal standard, meaning that the electric current is used to proportionately represent measurements or command signals. Typically, a 4 milliamp current value represents 0% of scale, a 20 milliamp current value represents 100% of scale, and any current value in between 4 and 20 milliamps represents a commensurate percentage in between 0% and 100%.

For example, if a 4-20 mA temperature transmitter were calibrated for a measurement range of 50 to 250 degrees C, the current and measured temperature values relate to each other on a graph like this:

Figure 3.1: Temperature transmitter with 4-20 mA output

This is not unlike the pneumatic instrument signal standard of 3 to 15 pounds per square inch (psig), where a varying air pressure signal represents some process measurement in an analog (proportional) fashion. Both signal standards are referred to as *live zero* because their ranges begin with a non-zero value (3 psig in the case of the 3-15 psig standard, and 4 milliamps in the case of the

4-20 mA standard). This "live" zero provides a simple means of discriminating between a legitimate 0% signal value and a failed signal (e.g., leaking tube or severed cable)[1].

DC current signals are also used in control systems to command the positioning of a final control element, such as a control valve or a variable-speed motor drive (VSD). In these cases, the milliamp value does not directly represent a process measurement, but rather how the degree to which the final control element influences the process. Typically (but not always!), 4 milliamps commands a closed (shut) control valve or a stopped motor, while 20 milliamps commands a wide-open valve or a motor running at full speed.

Thus, most industrial control systems use at least *two* different 4-20 mA signals: one to represent the process variable (PV) and one to represent the command signal to the final control element (the "manipulated variable" or MV):

Figure 3.2: Block diagram of a process control loop

The relationship between these two signals depends entirely on the response of the controller. There is no reason to ever expect the two current signals to be equal, for they represent entirely different things. In fact, if the controller is reverse-acting, it is entirely normal for the two current signals to be inversely related: as the PV signal increases going to a reverse-acting controller, the output signal will decrease. If the controller is placed into "manual" mode by a human operator, the output signal will have no automatic relation to the PV signal at all, instead being entirely

[1]Not all industrial measurement and control signals are "live zero" like the 3-15 psig and 4-20 mA standards. 0 to 10 volts DC is a common "dead zero" signal standard, although far more common in environmental (building heating and cooling) control systems than industrial control systems. Think of what would happen if an analog control system used -10 volts to +10 volts as its analog signal range, which means 0 volts represents a 50% signal! A failed signal path in such a system will be very misleading indeed, as a 50% signal value is not suspicious in the least.

determined by the operator's whim.

## 3.3 Relating 4 to 20 mA signals to instrument variables

Calculating the equivalent milliamp value for any given percentage of signal range is quite easy. Given the linear relationship between signal percentage and milliamps, the equation takes the form of the standard *slope-intercept* line equation $y = mx + b$. Here, $y$ is the equivalent current in milliamps, $x$ is the desired percentage of signal, $m$ is the span of the 4-20 mA range (16 mA), and $b$ is the offset value, or the "live zero" of 4 mA:

$$\text{current} = (16\text{ mA})\left(\frac{x}{100\%}\right) + (4\text{ mA})$$

This equation form is identical to the one used to calculate pneumatic instrument signal pressures (the 3 to 15 psig standard):

$$\text{pressure} = (12\text{ psid})\left(\frac{x}{100\%}\right) + (3\text{ psig})$$

The same mathematical relationship holds for *any* linear measurement range. Given a percentage of range $x$, the measured variable is equal to:

$$\text{measured variable} = (\text{Span})\left(\frac{x}{100\%}\right) + (\text{LRV})$$

Some practical examples of calculations between milliamp current values and process variable values follow:

**Example 3.1** An electronic loop controller outputs a signal of 8.55 mA to a direct-responding control valve (where 4 mA is shut and 20 mA is wide open). How far open should the control valve be at this MV signal level?

**Answer to Example 3.1** Convert the milliamp signal value into a percentage of valve travel. This means determining the percentage value of the 8.55 mA signal on the 4-20 mA range. First, manipulate the percentage-milliamp formula to solve for percentage ($x$):

$$(16\text{ mA})\left(\frac{x}{100\%}\right) + (4\text{ mA}) = \text{ current}$$

$$(16\text{ mA})\left(\frac{x}{100\%}\right) = \text{ current} - (4\text{ mA})$$

$$\frac{x}{100\%} = \frac{\text{current} - (4\text{ mA})}{(16\text{ mA})}$$

$$x = \left(\frac{\text{current} - (4\text{ mA})}{(16\text{ mA})}\right)100\%$$

Next, substitute in the 8.55 mA signal value and solve for $x$:

$$x = \left(\frac{8.55\text{ mA} - (4\text{ mA})}{(16\text{ mA})}\right)100\%$$

$$x = 28.4\%$$

Therefore, the control valve should be 28.4 % open when the MV signal is at a value of 8.55 mA.

**Example 3.2** A flow transmitter is ranged 0 to 350 gallons per minute, 4-20 mA output, direct-responding. Calculate the current signal value at a flow rate of 204 gpm.

**Answer to Example 3.2**First, convert the flow value of 204 gpm into a percentage of range. This is a simple matter of division, since the flow measurement range is zero-based:

$$\frac{204 \text{ gpm}}{350 \text{ gpm}} = 0.583 = 58.3\%$$

Next, take this percentage value and translate it into a milliamp value using the formula previously shown:

$$(16 \text{ mA}) \left(\frac{x}{100\%}\right) + (4 \text{ mA}) = \text{ current}$$

$$(16 \text{ mA}) \left(\frac{58.3\%}{100\%}\right) + (4 \text{ mA}) = 13.3 \text{ mA}$$

Therefore, the transmitter should output a PV signal of 13.3 mA at a flow rate of 204 gpm.

**Example 3.3** An electronic temperature transmitter is ranged 50 to 140 degrees Fahrenheit and has a 4-20 mA output signal. Calculate the current output by this transmitter if the measured temperature is 79 degrees Fahrenheit.

**Answer to Example 3.3** First, convert the temperature value of 79 degrees into a percentage of range based on the knowledge of the temperature range span (140 degrees − 50 degrees = 90 degrees) and lower-range value (LRV = 50 degrees). to do so, manipulate the general formula for any linear measurement to solve for $x$:

$$\text{measured variable} = (\text{Span}) \left(\frac{x}{100\%}\right) + (\text{LRV})$$

$$\text{measured variable} - (\text{LRV}) = (\text{Span}) \left(\frac{x}{100\%}\right)$$

$$\frac{\text{measured variable} - (\text{LRV})}{(\text{Span})} = \frac{x}{100\%}$$

$$x = \left(\frac{\text{measured variable} - (\text{LRV})}{(\text{Span})}\right) 100\%$$

$$x = \left(\frac{79^o\text{F} - 50^o\text{F}}{90^o\text{F}}\right) 100\%$$

$$x = 32.2\%$$

Next, take this percentage value and translate it into a 4-20 mA current value using the formula previously shown:

$$(16 \text{ mA})\left(\frac{x}{100\%}\right) + (4 \text{ mA}) = \text{ current}$$

$$(16 \text{ mA})\left(\frac{32.2\%}{100\%}\right) + (4 \text{ mA}) = 9.16 \text{ mA}$$

Therefore, the transmitter should output a PV signal of 9.16 mA at a temperature of 79 $^o$F.

**Example 3.4** A pH transmitter has a calibrated range of 4 pH to 10 pH, with a 4-20 mA output signal. Calculate the pH sensed by the transmitter if its output signal is 11.3 mA.

**Answer to Example 3.4** First, convert the milliamp value into a percentage. Following the same technique used for the control valve problem:

$$\left(\frac{\text{current} - (4 \text{ mA})}{(16 \text{ mA})}\right) 100\% = \text{ percent of range}$$

$$\left(\frac{11.3 \text{ mA} - (4 \text{ mA})}{(16 \text{ mA})}\right) 100\% = 0.456 = 45.6\%$$

Next, take this percentage value and translate it into a pH value, given the transmitter's measurement span of 6 pH (10 pH − 4 pH) and offset of 4 pH:

$$(10 \text{ pH} - 4 \text{ pH})\left(\frac{x}{100\%}\right) + (4 \text{ pH}) = \text{ pH value}$$

$$(6 \text{ pH})\left(\frac{45.6\%}{100\%}\right) + (4 \text{ pH}) = 6.74 \text{ pH}$$

Therefore, the transmitter's 11.3 mA output signal reflects a measured pH value of 6.74 pH.

**Example 3.5** A current-to-pressure transducer is used to convert a 4-20 mA electronic signal into a 3-15 psig pneumatic signal. This particular transducer is configured for **reverse action** instead of direct, meaning that its pressure output at 4 mA should be 15 psig and its pressure output at 20 mA should be 3 psig. Calculate the necessary current signal value to produce an output pressure of 12.7 psig.

**Answer to Example 3.5** Reverse-acting instruments are still linear, and therefore still follow the slope-intercept line formula $y = mx + b$. The only differences are a negative slope and a different intercept value. Instead of $y = 16x+4$ as is the case for direct-acting instruments, this reverse-acting instrument follows the linear equation $y = -16x + 20$:

$$(-16 \text{ mA})\left(\frac{x}{100\%}\right) + (20 \text{ mA}) = \text{ current}$$

First, convert the pressure signal value of 12.7 psig into a percentage of 3-15 psig range. To do this, manipulate the percentage-pressure formula to solve for $x$:

$$(12 \text{ psid})\left(\frac{x}{100\%}\right) + (3 \text{ psig}) = \text{ pressure}$$

$$(12 \text{ psid})\left(\frac{x}{100\%}\right) = \text{ pressure} - (3 \text{ psig})$$

$$\frac{x}{100\%} = \frac{\text{pressure} - (3 \text{ psig})}{(12 \text{ psid})}$$

$$x = \left(\frac{\text{pressure} - (3 \text{ psid})}{(12 \text{ psid})}\right) 100\%$$

Next, substitute in the 12.7 psig signal value and solve for $x$:

$$x = \left(\frac{12.7 \text{ psig} - (3 \text{ psig})}{(12 \text{ psid})}\right) 100\%$$

$$x = 80.8\%$$

This says that 12.7 psig represents 80.8 % of the 3-15 psig signal range. Substituting this percentage value into the modified (negative-slope) percentage-current formula will determine how much current is necessary to generate this 12.7 psig pneumatic output:

$$(-16 \text{ mA})\left(\frac{x}{100\%}\right) + (20 \text{ mA}) = \text{ current}$$

$$(-16 \text{ mA})\left(\frac{80.8\%}{100\%}\right) + (20 \text{ mA}) = 7.07 \text{ mA}$$

Therefore, a current signal of 7.07 mA is necessary to drive the output of this reverse-acting I/P transducer to a pressure of 12.7 psig.

### 3.3.1 Graphical interpretation of signal ranges

A helpful illustration for students in understanding analog signal ranges is to consider the signal range to be expressed as a length on a number line. For example, the common 4-20 mA analog current signal range would appear as such:

Figure 3.3: The 4-20 mA signal range

If one were to ask the percentage corresponding to a 14.4 mA signal on a 4-20 mA range, it would be as simple as determining the length of a line segment stretching from the 4 mA mark to the 14.4 mA mark:

Figure 3.4: The 4-20 mA signal range with an example value indicated in mA and %

As a percentage, this thick line is 10.4 mA long (the distance between 14.4 mA and 4 mA) over a total (possible) length of 16 mA (the total span between 20 mA and 4 mA). Thus:

$$\text{Percentage} = \left(\frac{14.4\ \text{mA} - 4\ \text{mA}}{20\ \text{mA} - 4\ \text{mA}}\right) 100\%$$

$$\text{Percentage} = 65\%$$

This same "number line" approach may be used to visualize any conversion from one analog scale to another. Consider the case of an electronic pressure transmitter calibrated to a pressure range of -5 to +25 psig, having an (obsolete) current signal output range of 10 to 50 mA. The appropriate current signal value for an applied pressure of +12 psig would be represented on the number line as such:

Figure 3.5: Conversion of a pressure input to a 10-50 mA signal

Finding the "length" of this line segment in units of milliamps is as simple as setting up a proportion between the length of the line in units of psig over the total (span) in psig, to the length of the line in units of mA over the total (span) in mA:

$$\frac{17 \text{ psig}}{30 \text{ psig}} = \frac{?\ \text{mA}}{40 \text{ mA}}$$

Solving for the unknown (?) current by cross-multiplication and division yields a value of 22.67 mA. Of course, this value of 22.67 mA is only the length of the line segment on the number line; it does not directly give the current signal value. To find that, add the "live zero" offset of 10 mA, for a final result of 32.67 mA.

Figure 3.6: Conversion of a pressure input to a mA signal

Thus, an applied pressure of +12 psig to this transmitter should result in a 32.67 mA output signal.

## 3.4 2-wire transmitter current loops

DC electric current signals may also be used to communicate process measurement information from transmitters to controllers, indicators, recorders, alarms, and other input devices. The simplest form of 4-20 mA measurement loop is one where the transmitter has two terminals for the 4-20 mA signal wires to connect. It is possible to convey electrical power *and* communicate analog information over the same two wires using 4 to 20 milliamps DC, if the transmitter is designed to be *loop-powered*. A loop-powered transmitter connects to a process controller in the following manner:

Figure 3.7: Diagram of a two wire transmitter loop

A 2-wire transmitter's circuitry is designed to act as a current *regulator*, limiting current in the series loop to a value representing the process measurement, while relying on a remote source of power to motivate current to flow. Thus, this is a passive form of generating a 4-20 mA signal. Please note the direction of the arrow in the transmitter's dependent current source symbol, and how it relates to the voltage polarity marks. This means the current source is actually acting more like a variable resistor than a supply.

A loop-powered transmitter gets its operating power from the minimum terminal voltage and current available at its two terminals. With the typical source voltage being 24 volts DC, and the maximum voltage dropped across the controller's 250 ohm resistor being 5 volts DC, the transmitter should always have at least 19 volts available at its terminals when a single receiver is connected[2]. Given the lower end of the 4-20 mA signal range, the transmitter should always have at least 4 mA of current to run on. Thus, the transmitter will always have a certain minimum amount of electrical power available on which to operate, while regulating current to signal the process measurement.

Internally, the loop-powered transmitter circuitry looks something like this:

[2]ISA Standard 50.00.01 requires a transmitter to be able to deliver rated current into a loop resistance of at least 600 Ω, and the standard receiver resistance is 250±0.1% Ω

Figure 3.8: Example circuitry in a two wire transmitter

All sensing, scaling, and output conditioning circuitry inside the transmitter must be designed to run on less then 4 mA of DC current, and at a modest terminal voltage. In order to create loop currents exceeding 4 mA – as the transmitter must do in order to span the entire 4 to 20 milliamp signal range – the transmitter circuitry uses a transistor to shunt (bypass) extra current from one terminal to the other as needed to make the total current indicative of the process measurement. For example, if the transmitter's internal operating current is only 3.8 mA, and it must regulate loop current at a value of 16 mA to represent a condition of 75% process measurement, the transistor will bypass 12.2 mA of current.

Early current-based industrial transmitters were not capable of operating on such low levels of electrical power, and so used a different current signal standard: 10 to 50 milliamps DC. Loop power supplies for these transmitters ranged upwards of 90 volts to provide enough power for the transmitter. Safety concerns made the 10-50 mA standard unsuitable for some industrial installations, and modern microelectronic circuitry with its reduced power consumption made the 4-20 mA standard practical for nearly all types of process transmitters.

## 3.5 3- and 4-wire transmitter current loops

Some transmitters have two terminals for the 4-20 mA signal wires to connect, and either one or two more terminals where a power source connects. This type of transmitter has it's own current *source* to drive the loop, but receives power from an external power supply. These transmitters are called "3-wire" or "4-wire" as appropriate, or externally-powered. Since 3-wire transmitters are not as common as 4-wire, the remainder of this discussion focuses on 4-wire devices only. The current signal from the transmitter connects to the *process variable input* terminals of the controller to

complete the loop:

Figure 3.9: Diagram of a four wire transmitter loop

Refer back to the illustration of the 2-wire transmitter circuit for comparison. The current "source" in a 2-wire loop-powered transmitter actually behaves as an electrical *load*, while the current source in the 4-wire transmitter functions as a true electrical source.

Most process instruments such as pressure, level, and temperature are 2-wire tranmitter, not 4-wire transmitters. Flow and analytical transmitters often are externally-powered, however, since these generally require more operating power than a 2-wire device has available in its current loop. In some installations, transmitter power is supplied through additional wires in the cable from a power source located in the same panel as the controller:

Figure 3.10: Diagram of a four wire transmitter with a power source in the panel

One obvious disadvantage of this scheme is the requirement of two more conductors in the cable. More conductors means the cable will be larger in diameter and more expensive for a given length. Cables with more conductors will require larger electrical conduit, and all field wiring panels will have to contain more terminal blocks to marshal the additional conductors. If no suitable electrical power source exists at the transmitter location, though, a 4-wire cable is necessary to service a 4-wire transmitter.

A second disadvantage of this scheme is that different power sources must be isolated from one another. If the power source of a transmitter is different than the power source of the receiving device, the systems can affect each other, especially if a ground loop is created. This typically means that *signal isolators* must be used between 4-wire transmitters and controllers, adding additional cost and complexity. Alternatively, controllers can be selected with isolated (differential) inputs, which generally cut the number of inputs available on a particular card in half, meaning more input cards are needed to serve a given number of transmitters. Again, this adds cost.

## 3.6 4 to 20 mA loop accessories

### 3.6.1 Resistors

Typically, process controllers are not equipped to directly accept milliamp input signals, but rather voltage signals. For this reason a precision resistor must be connected across the input terminals to convert the 4-20 mA signal into a standardized analog voltage signal that the controller can understand. A voltage signal range of 1 to 5 volts is standard, although some models of controller use different voltage ranges and therefore require different precision resistor values. If the voltage range is 1-5 volts and the current range is 4-20 mA, the precision resistor value must be 250 Ω (typically with 0.1% tolerance or better).

Since this is a digital controller, the input voltage at the controller terminals is interpreted by an analog-to-digital converter (ADC) circuit, which converts the measured voltage into a digital number that the controller's microprocessor can work with.

### 3.6.2 Signal conditioners

Signal isolators provide electrical isolation between devices with different power supplies. They typically are used with 4-wire devices as mentioned above, with the 4-wire device current source on the input of the signal isolator and a loop-powered circuit on the output of the isolator. However, they can also be operated in the other direction, where the input is loop-powered and the output is externally-powered.

Signal converters are used to convert a signal from one electrical level to another. For example, a converter may have available ranges for either circuit of 0-5 V, 0-10 V, 0-20 mA, or 4-20 mA. By setting up the converter properly (either via dip switches or by computer configuration software), any combination of input and output may be selected, e.g., a 0-10 V input and a 4-20 mA output.

Signal splitters are used when a signal needs to be delivered to more than one receiving device. Since 4-20 mA current may be delivered to multiple devices connected in series, if any one receiver fails, the entire loop is lost. A splitter will allow a receiver to fail or be removed from the circuit without affecting any of the other receivers connected to it. Also, transmitters typically only have a compliance of 600 Ω. Thus, only two receivers that use 250 Ω resistors may be connected as loads in series in a single 4-20 mA loop. A splitter can be used to allow more devices to be loads on a single transmitter without exceeding the compliance. Splitters are often necessary when two controllers are to receive a signal from one transmitter, since controllers often expect to provide the ground reference for a signal. If the controllers do not share a common ground, problems occur with one of them.

Signal isolators, converters, and splitters may receive power from their input circuit, their output circuit, or from an external source, so it is important to understand their power requirements during design so that the circuit works properly.

## 3.7 Controller output current loops

The simplest form of 4-20 mA current loop is the type used to represent the output of a process controller, sending a command signal to a final control element. Here, the controller both supplies the electrical power and regulates the DC current to the final control element, which acts as an electrical load. To illustrate, consider the example of a controller sending a 4-20 mA signal to an I/P (current-to-pressure) signal converter, which then pneumatically drives a control valve:

Figure 3.11: The 4-20 mA controller output loop used with an I/P converter to position a valve

This particular controller has two digital displays, one for process variable (PV) and one for setpoint (SP), with a bargraph for displaying the output value (Out). One pushbutton provides the operator with a way to switch between Automatic and Manual modes (A/M), while two other pushbuttons provide means to decrement and increment either the setpoint value (in Automatic mode) or the Output value (in Manual mode).

Inside the controller, a *dependent current source* provides the 4-20 mA DC current signal to the I/P transducer. Like all current sources, its purpose is to maintain current in the "loop" circuit regardless of circuit resistance or any external voltage sources. Unlike a constant current source, a "dependent" current source (represented by a diamond shape instead of a circle shape) varies its current value according to the dictates of some external stimulus. In this case, either the mathematical function of the controller (Automatic mode) or the arbitrary setting of the human operator (Manual mode) tells the current source how much DC current it should maintain in the circuit.

For example, if the operator happened to switch the controller into Manual mode and set the output value at 50%, the proper amount of DC current for this signal percentage would be 12 mA (exactly half-way between 4 mA and 20 mA). If everything is working properly, the current in the "loop" circuit to the I/P transducer should remain exactly at 12 mA regardless of slight changes in wire resistance, I/P coil resistance, or anything else: the current source inside the controller will "fight" as hard as it has to in order to maintain this set amount of current. This current, as it flows through the wire coil of the I/P transducer mechanism, creates a magnetic field inside the I/P to actuate the pneumatic mechanism and produce a 9 psig pressure signal output to the control valve (9 psig being exactly half-way between 3 psig and 15 psig in the 3-15 psig signal standard range). This should move the control valve to the half-way position.

The details of the controller's internal current source are not terribly important. Usually, it takes the form of an operational amplifier circuit driven by the voltage output of a DAC (Digital-to-Analog

Converter). The DAC converts a binary number (either from the controller's automatic calculations, or from the human operator's manual setting) into a small DC voltage, which then commands the op-amp circuit to regulate output current at a proportional value.

The scenario is much the same if a variable-speed motor drive replaces the I/P and control valve. From the controller's perspective, the only difference it sees is a resistive load instead of an inductive load. The input resistance of the motor drive circuit converts the 4-20 mA signal into an analog voltage signal (typically 1-5 V, but not always). This voltage signal then constitutes a command to the rest of the drive circuitry, telling it to modulate the power going to the electric motor in order to drive it at the desired speed:

Figure 3.12: The 4-20 mA controller output loop used with a motor drive

Here, the variable-speed motor drive is a high-power electronic circuit that takes 3-phase AC power in and converts it to DC power at a variable voltage (that voltage level controlled by the 4-20 mA signal from the controller output). Another style of motor "drive" is one that takes in AC power and outputs 3-phase AC power at variable voltage and frequency to power an AC induction motor. This latter style is usually called a *variable frequency drive* or *VFD*.

## 3.8 Troubleshooting current loops

A fundamental principle in instrumentation system troubleshooting is that every instrument has at least one input and at least one output, and that the output(s) should accurately correspond to the input(s). If an instrument's output is not properly corresponding to its input according to the instrument's design function, there must be something wrong with that instrument.

Consider the inputs and outputs of several common instruments: transmitters, controllers, indicators, and control valves. Each of these instruments takes in (input) data in some form, and

generates (output) data in some form. In any instrument "loop," the output of one instrument feeds into the input of the next, such that information is passed from one instrument to another. By intercepting the data communicated between components of an instrument system, it is possible to locate and isolate faults. In order to properly understand the intercepted data, the inputs and outputs of the respective instruments and the basic functions of those instruments must be understood.

The following illustrations highlight inputs and outputs for instruments commonly found in control systems:

Figure 3.13: Inputs and outputs associated with types of transmitters

Figure 3.14: Inputs and outputs associated with a controller

In order to check for proper correspondence between instrument inputs and outputs, the appropriate test equipment must be used to intercept the signals going into and out of those instruments. For 4-20 mA analog signal-based instruments, this means using electrical meters capable of accurately measuring current and voltage.

### 3.8.1 Using a standard milliammeter to measure loop current

Since the signal of interest is represented by an electric current in an instrumentation current "loop" circuit, the obvious tool to use for troubleshooting is a multimeter capable of accurately measuring DC milliamperes. Unfortunately, though, there is a major disadvantage to the use of a milliammeter: the circuit must be "broken" at some point to connect the meter in series with the current, and this means the current will fall to 0 mA until the meter is connected (then fall to 0 mA when the meter is removed from the circuit). Interrupting the current means interrupting the flow of information conveyed by that current, be it a process measurement or a command signal to a final control element. This *will* have adverse effects on a control system unless certain preparatory steps are taken.

Before "breaking the loop" to connect a meter, first warn all appropriate personnel that the signal will be interrupted at least twice, falling to a value of -25% each time. If the signal to be interrupted is coming from a process transmitter to a controller, the controller should be placed in manual mode so it will not cause an upset in the process (by moving the final control element in response to the sudden loss of PV signal). Also, process alarms should be temporarily disabled so they do not cause panic. If this current signal also drives process shutdown alarms, these should be

temporarily disabled so that nothing shuts down upon interruption of the signal.

If the current signal to be interrupted is a command signal from a controller to a final control element, the final control element either needs to be manually overridden so as to hold a fixed setting while the signal varies, or it needs to be bypasses completely by some other device(s). If the final control element is a control valve, this typically takes the form of opening a bypass valve and closing at least one block valve:

Figure 3.15: Diagram of a typical control valve piping arrangement

Since the manually-operated bypass valve now performs the job that the automatic control valve used to, a human operator must remain posted at the bypass valve to carefully throttle it and maintain control of the process.

Block and bypass valves for a large gas flow control valve may be seen in the following photograph:

Figure 3.16: A typical control valve piping arrangement in an industrial installation

In consideration of the labor necessary to safely interrupt the current signal to a control valve in a live process, the seemingly simple task of connecting a milliammeter in series with a 4-20 mA current signal is not as easy as it may first appear. Better ways must exist, no?

### 3.8.2 Using a clamp-on milliammeter to measure loop current

One better way to measure a 4-20 mA signal without interrupting it is to do so magnetically, using a clamp-on milliammeter. Modern Hall-effect sensors are sensitive and accurate enough to monitor the weak magnetic fields created by the passage of small DC currents in wires. Ammeters using Hall-effect sensors have are completely non-intrusive because they merely clamp around the wire, with no need to "break" the circuit. An example of a such a clamp-on current meter is the Fluke model 771, shown in this photograph:

ZERO
HOLD
WAKE UP
mA
3.98
FLUKE 771
MILLIAMP PROCESS CLAMP METER

Note how this milliammeter not only registers loop current (3.98 mA as shown in the photograph), but it also converts the milliamp value into a percentage of range, following the 4 to 20 mA signal standard. One disadvantage to be aware of for clamp-on milliammeters is the susceptibility to error from strong external magnetic fields. Steady magnetic fields (from permanent magnets or DC-powered electromagnets) may be compensated for by performing a "zero" adjustment with the instrument held in a similar orientation prior to measuring loop current through a wire.

### 3.8.3 Using "test" diodes to measure loop current

Another way to measure a 4-20 mA signal without interrupting it involves the use of a rectifying diode, originally installed in the loop circuit when it was constructed. A "test" diode may be placed anywhere in series within the loop in such a way that it will be forward-biased. During normal operation, the diode will drop approximately 0.7 volts, as is typical for any silicon rectifying diode when forward biased. The following schematic diagram shows such a diode installed in a 2-wire transmitter loop circuit:

Figure 3.18: Schematic diagram showing a test diode used in a two-wire loop

If someone connects a milliammeter in parallel with this diode, however, the very low input resistance of the ammeters "shorts past" the diode and prevents any substantial voltage drop from forming across it. Without the necessary forward voltage drop, the diode effectively turns off and conducts 0 mA, leaving the entire loop current to pass through the ammeter:

Figure 3.19: Schematic diagram showing an ammeter connected across a test diode

When the milliammeter is disconnected, the requisite 0.7 volt drop appears to turn on the diode, and all loop current flows through the diode again. At no time is the loop current ever interrupted,

which means a technician may take current measurements this way and never have to worry about generating false process variable indications, setting off alarms, or upsetting the process.

Such a diode may be installed at the nearest junction box, between terminals on a terminal strip, or even incorporated into the transmitter itself. Some process transmitters have an extra pair of terminals labeled "Test" for this exact purpose. A diode is already installed in the transmitter, and these "test" terminals serve as points to connect the milliammeter across.

The following photograph shows an example of this on a Rosemount model 3051 differential pressure transmitter:

Figure 3.20: An industrial transmitter with test points for ammeter connection

Note the two test points labeled "TEST" below and to the right of the main screw terminals where the loop wiring attaches. Connecting an ammeter to these two test points allows for direct measurement of the 4-20 mA current signal without having to un-do any wire connections in the circuit.

Transmitters equipped with analog meter movements for direct visual indication of the 4-20 mA signal usually connect the analog milliammeter in parallel with just such a diode. The reason for doing this is to maintain loop continuity in the event that the fine-wire coil inside the milliammeter movement were to accidently break open.

### 3.8.4 Using shunt resistors to measure loop current

A similar method for non-invasively measuring current in a 4-20 mA instrumentation circuit is to install a precision resistor in series. If the resistance value is precisely known, the technician merely needs to measure voltage across it with a voltmeter and use Ohm's Law to calculate current:

Figure 3.21: Current can be indirectly measured using a sense resistor and a voltmeter

In electronics, such a precision resistor used for measuring current is often referred to as a *sense* or *shunt* resistor. Shunt resistor values are commonly very small, for their purpose is to assist in current measurement without imposing undue voltage drop within a circuit. It is rare to find a 250 ohm resistor used strictly as a diagnostic shunt resistor, because the extra voltage drop (1 to 5 volts, depending on the current signal level) may "starve" loop-powered instruments of voltage necessary to operate. Shunt resistor values as low as 1 ohm may be found installed in 4-20 mA current loops at strategic locations where technicians may need to measure loop current[3].

### 3.8.5 Troubleshooting current loops with voltage measurements

If neither component (diode nor shunt resistor) is pre-installed in the circuit, and if a Hall-effect (clamp-on) precision milliammeter is unavailable, a technician may still perform useful troubleshooting measurements using nothing but a DC voltmeter. Here, however, one must be careful of how to interpret these voltage measurements, for they may not directly correspond to the loop current as was the case with measurements taken in parallel with the precision resistor.

Take for example this 4-20 mA loop where a controller sends a command signal to an I/P transducer shown in Figure 3.11:

There is no standardized resistance value for I/P transducer coils, and so the amount of voltage dropped across the I/P terminals for any given amount of loop current will be unique for every different model of I/P. The Fisher model 567 I/P transducer built for 4-20 mA signals has a nominal coil resistance of 176 ohms. Thus, a voltage drop can be expected of approximately 0.7 volts at 4 mA and a drop of approximately 3.5 volts at 20 mA across the I/P terminals. Since the controller output terminals are directly in parallel with the I/P terminals, approximately the same voltage should be measured there as well (slightly greater due to wire resistance). The lack of known precision in the I/P coil resistance makes it difficult to tell exactly how much current is in the loop for any given voltage measurement taken with a voltmeter. However, the approximate coil resistance of the I/P is known, an estimate of loop current can be obtained, which is usually good enough for diagnostic purposes.

If the I/P coil resistance is completely unknown, voltage measurements become useless for quantitative determination of loop current. Voltage measurements would be useful only for qualitatively determining loop continuity (i.e., whether there is a break in the wiring between the controller and I/P).

[3]Of course, a 1 ohm resistor would drop 4 mV at 4 mA loop current, and drop 20 mV at 20 mA loop current. These small voltage values necessitate a highly accurate DC voltmeter for field measurement!

Another example for consideration is this loop-powered 4-20 mA transmitter and controller circuit, where the controller supplies DC power for the loop:

Figure 3.22: Current can be indirectly measured using a voltmeter and a dropping resistor

It is very common to find controllers with their own built-in loop power supplies, due to the popularity of loop-powered (2-wire) 4-20 mA transmitters. If the transmitter requires a DC voltage source somewhere in the circuit to power it up, it makes sense to include one in the controller, right?

The only voltage measurement that directly and accurately correlates to loop current is the voltage directly across the 250 ohm precision resistor. A loop current of 4 mA will yield a voltage drop of 1 volt, 12 mA will drop 3 volts, 20 mA will drop 5 volts, etc.

A voltage measurement across the transmitter terminals will show the *difference* in voltage between the 26 volt power supply and the voltage dropped across the 250 ohm resistor. In other words, the transmitter's terminal voltage is simply what is left over from the source voltage of 26 volts after subtracting the resistor's voltage drop. This makes the transmitter terminal voltage inversely proportional to loop current: the transmitter sees approximately 25 volts at 4 mA loop current (0% signal) and approximately 21 volts at 20 mA loop current (100% signal).

The use of the word "approximate" is very intentional here, for loop power supplies are usually non-regulated. In other words, the "26 volt" rating is approximate and subject to change! One of the advantages of the loop-powered transmitter circuit is that the source voltage is largely irrelevant, so long as it exceeds the minimum value necessary to ensure adequate power to the transmitter. If the source voltage drifts for any reason, it will have no impact on the measurement signal at all, because the transmitter is built as a *current regulator*, regulating current in the loop to whatever value represents the process measurement, regardless of slight changes in loop source voltage, wire resistance, etc. This rejection of power supply voltage changes means that the loop power supply need not be regulated, and so in practice it rarely is.

This identifies a common problem in loop-powered 4-20 mA transmitter circuits: maintaining

sufficient operating voltage at the transmitter terminals. Recall that a loop-powered transmitter relies on the voltage dropped across its terminals (combined with a current of less than 4 mA) to power its internal workings. This means the terminal voltage must not be allowed to dip below a certain minimum value, or else the transmitter will not have enough electrical power to continue its normal operation. This makes it possible to "starve" the transmitter of voltage if the loop power supply voltage is insufficient, and/or if the loop resistance is excessive.

To illustrate how this can be a problem, consider the following 4-20 mA measurement loop, where the controller supplies only 20 volts DC to power the loop, and an indicator is included in the circuit to provide operators with field-located indication of the transmitter's measurement:

Figure 3.23: Loop circuits can be overloaded by too much loop resistance

The indicator contains its own 250 ohm resistor to provide a 1-5 volt signal for the meter mechanism to sense. This means the total loop resistance is now 500 ohms (plus any wire resistance). At full current (20 mA), this total resistance will drop (at least) 10 volts, leaving 10 volts or less at the transmitter terminals to power the transmitter's internal workings. 10 volts may not be enough for the transmitter to successfully operate, though. The Rosemount model 3051 pressure transmitter, for example, requires a minimum of 10.5 volts at the terminals to operate.

However, the transmitter *will* operate just fine at lower loop current levels. When the loop current is only 4 mA, for example, the combined voltage drop across the two 250 ohm resistors will be only 2 volts, leaving about 18 volts at the transmitter terminals: more than enough for practically any model of 4-20 mA loop-powered transmitter to successfully operate. Thus, the problem of insufficient supply voltage only manifests itself when the process measurement nears 100% of range. This could be a difficult problem to diagnose, since it appears only during certain process conditions. A technician looking only for wiring faults (loose connections, corroded terminals, etc.) would never find the problem.

When a loop-powered transmitter is starved of voltage, its behavior becomes erratic. This is especially true of "smart" transmitters with built-in microprocessor circuitry. If the terminal voltage dips below the required minimum, the microprocessor circuit shuts down. When the circuit shuts down, the current draw decreases accordingly. This causes the terminal voltage to rise again, at which point the microprocessor has enough voltage to start up. As the microprocessor "boots" back up again, it increases loop current to reflect the near-100% process measurement. This causes the terminal voltage to sag, which subsequently causes the microprocessor to shut down again. The result is a slow on/off cycling of the transmitter's current, which makes the process controller think the process variable is surging wildly. The problem disappears, though, as soon as the process measurement decreases enough that the transmitter is allowed enough terminal voltage to operate normally.

The amount of resistance allowed in a loop is governed by ANSI/ISA-50.00.01-1975(R2002), "Compatibility of Analog Signals for Electronic Industrial Process Instruments". A standard current output signal of 4-20 mA must be able to be delivered into a load of at least 600 Ω. Exceeding this amount of resistance may make a loop behave abnormally. This standard also specifies other conditions for both current and voltage transmitters.

### 3.8.6 Using loop calibrators

Special-purpose electronic test instruments called *loop calibrators* are manufactured for the express purpose of 4-20 mA current loop circuit troubleshooting. These versatile instruments are generally capable of not only measuring current, but also *sourcing* current to unpowered devices in a loop, and also *simulating* the operation of loop-powered 4-20 mA transmitters.

A very popular loop calibrator unit is the Altek model 334A, a battery-powered, hand-held unit with a rotary knob for current adjustment and toggle switches for mode setting. The following illustration shows how this calibrator would be used to measure current in a functioning input signal loop[4]:

[4] In the following illustrated examples, the transmitter is assumed to be a pressure transmitter with a calibrated range of 0 to 750 inches of water column, 4-20 mA. The controller's PV (process variable) display is appropriately ranged to display 0 to 750 as well.

Figure 3.24: Diagram illustrating use of a loop calibrator to read a current loop signal

Here, the loop wiring is broken at the negative terminal of the loop-powered transmitter, and the calibrator connected in series to measure current. If this loop had a test diode installed, the calibrator could be connected in parallel with the diode to achieve the same function. Note the polarity of the calibrator's test leads in relation to the circuit being tested: the calibrator is acting as an unpowered device (a load rather than a source), with the more positive loop terminal connected to the calibrator's red test lead and the more negative terminal connected to the black test lead.

The same loop calibrator may be used to *source* (or *drive*) a 4-20 mA signal into an indicating instrument to test the function of that instrument independently. Here, the Altek calibrator is used as a current source to send a 16.00 mA signal to the PV (process variable) input of the controller:

Figure 3.25: Diagram illustrating use of a loop calibrator to source a current loop signal

No transmitter need be included in this illustration, because the calibrator takes its place. Note how the calibrator is used here as an active *source* of current rather than a passive load as it was in the last example. The calibrator's red test lead connects to the controller's positive input terminal, while the black test lead connects to the negative terminal. The DC power source inside the controller is not used for loop power, because the calibrator in "source" mode provides the necessary power to drive current through the 250 ohm resistor.

An alternative method of sourcing a known current signal into an indicating instrument that provides loop power is to set the loop calibrator to a mode where it mimics the electrical behavior of a loop-powered 2-wire transmitter. In this mode, the calibrator serves to regulate loop current at a user-determined value, but it provides no motivating voltage to drive this current. Instead, it passively relies on some external voltage source in the loop circuit to provide the necessary electromotive force:

Figure 3.26: Diagram illustrating use of a loop calibrator to simulate a current loop signal using an external voltage source

Note the polarity of the calibrator's test leads in relation to the controller: the red test lead connects to the positive loop power terminal while the black lead connects to the positive input terminal. Here, the calibrator acts as a load, just as a loop-powered transmitter acts as an electrical load. The only source of electrical power in this test circuit is the 24 VDC source inside the controller: the same one normally providing energy to the circuit when a loop-powered transmitter is connected.

This *simulate transmitter* mode is especially useful for testing a 4-20 mA loop at the end of the cable where the transmitter is physically located. After disconnecting the cable wires from the transmitter and re-connecting them to the loop calibrator (set to "simulate" mode), the calibrator may be used to simulate a transmitter measuring any value within its calibrated range.

A legacy loop calibrator still familiar to many I&C professionals at the time of this writing is the classic Transmation model 1040:

Figure 3.27: A legacy loop calibrator

A modern loop calibrator manufactured by Fluke is the model 705:

Figure 3.28: A modern loop calibrator

With this calibrator, the *measure*, *source*, and *simulate* modes are accessed by repeatedly pushing a button, with the current mode displayed on the screen as shown in the subfigures in Figure 3.29.

(a) Measure

(b) Source

(c) Simulate

Figure 3.29: A calibrator screen displaying various function modes

### 3.8.7 NAMUR signal levels

4-20 mA process transmitters compliant with the NAMUR recommendations for fault signal levels will limit their output signals between 3.8 mA and less than 21 mA when functioning properly.

Signals lying outside this range indicate some form of failure has occurred within the transmitter[5].

| **Signal level** | **Fault condition** |
|---|---|
| Output $\leq$ 3.6 mA | Sensing transducer failed low |
| 3.6 mA $<$ Output $<$ 3.8 mA | Sensing transducer failed (detected) low |
| 3.8 mA $\leq$ Output $<$ 4.0 mA | Measurement under-range |
| 21.0 $>$ Output $\geq$ 20.5 mA | Measurement over-range |
| Output $\geq$ 21.0 mA | Sensing transducer failed high |

Any control system programmed to respond to these specific fault-induced current levels may act upon them by forcing controllers into manual mode, initiating shutdown procedures, or taking some other form of safe action appropriate to the knowledge of a failed process transmitter.

## 3.9 Analog pneumatic communication

Air or gas pressure may be used as an alternative signaling medium to electricity [1]. Since compressed air is not likely to ignite a cloud of hydrocarbon vapor or flammable dust, pneumatics are often used in locations that contain these hazards.

Imagine a pressure transmitter designed to output a variable air pressure according to its calibration rather than a variable electric current. Such a transmitter would have to be supplied with a source of constant-pressure compressed air instead of an electric voltage, and the resulting output signal would be conveyed to the indicator via tubing instead of wires:

Figure 3.30: Analog signal transmission using pneumatic pressure

The indicator in this case would be a special pressure gauge, calibrated to read in units of process pressure although actuated by the pressure of clean compressed air from the transmitter instead of directly by process fluid. The most common range of air pressure for industrial pneumatic instruments is 3 to 15 psig [2]. An output pressure of 3 psig represents the low end of the process measurement scale and an output pressure of 15 psig represents the high end of the measurement scale. Applied to the previous example of a transmitter calibrated to a range of 0 to 250 psig, a lack of process pressure would result in the transmitter outputting a 3 psig air signal and full process pressure would result in an air signal of 15 psig. The face of this special "receiver" gauge would be labeled from 0 to 250 psig, while the actual mechanism would operate on the 3 to 15 psig

[5] Of course, an open or shorted fault in the wiring connecting the transmitter with the rest of the system could easily force the current beyond this range, through no fault of the transmitter itself.

range output by the transmitter. Just like the 4-20 mA loop, the end-user need not know how the information gets transmitted from the process to the indicator. The 3-15 psig signal medium is once again transparent to the operator.

Pneumatic temperature, flow, and level control systems have all been manufactured to use the same principle of 3-15 psig air pressure signaling. In each case, the transmitter and controller are both supplied clean compressed air at some nominal pressure (20 to 25 psig, usually) and the instrument signals travel via tubing. The following illustrations show what some of these applications look like:

Figure 3.31: Pneumatic control of biodiesel temperature

Figure 3.32: Pneumatic control of flow

Figure 3.33: Pneumatic control of boiler water level

Instruments functioning on compressed air, and process measurement signals transmitted as air pressures through long runs of metal tubing, was the norm for industrial instrumentation prior to the advent of reliable electronics. In honor of this paradigm, instrument technicians were often referred to as *instrument mechanics*, for these air-powered devices were mechanically complex and in frequent need of adjustment to maintain high accuracy.

Back in the days of control room panels populated by rows and rows of pneumatic indicators, recorders, and controllers, clean and organized routing of all the instrument signal tubes was a significant concern. By contrast, electrical wires are relatively easy to organize through the use of marshaling panels and terminal blocks – bundles of tubes (especially metal tubes!) are not. A photograph taken of the upper rear portion of an old control room panel shows a portion of a marshaling board where dozens of bulkhead-style 1/4 inch instrument tube fittings are organized in neat rows[6], where a multitude of pneumatic instrument signal lines once attached:

[6]Note the staggered layout of the tube fittings, intended to improve access to each one. Remember that the technician used a 9/16 inch wrench to loosen and tighten the tube fitting nuts, so it was important to have working room between fittings in which to maneuver a wrench.

Figure 3.34: A bulkhead with pneumatic tube fittings

Each bulkhead fitting bears a numbered tag[7], for easy identification and documentation of tube connections. Loop diagrams of pneumatic control systems documented each bulkhead fitting where an instrument signal passed, in the same way that modern loop diagrams document each terminal block where an electrical signal connection is made.

Pneumatic instruments still find wide application in industry, although it is increasingly rare to encounter completely pneumatic control loops. One of the most common applications for pneumatic control system components is control valve actuation, where pneumatic technology still dominates. Not only is compressed air used to create the actuation force in many control valve mechanisms, it is still often the signal medium employed to command the valve's position. Quite often this pneumatic signal originates from a device called an *f/P transducer*, or *current-to-pressure converter*, taking a 4-20 mA control signal from the output of an electronic controller and translating that information as a pneumatic 3-15 psig signal to the control valve's positioner or actuator.

### 3.9.1 Foxboro model E69 electro-pneumatic transducer

Often it is necessary to convert an electrical current signal into a corresponding pneumatic signal. Devices that perform this function are called I/P transducers (or I/P converters) since the input

[7] The numbers are difficult to see here, because the entire panel has been painted in a thick coat of grey paint. This particular panel was stripped of all pneumatic instruments and outfitted with electronic instruments, so the rows of bulkhead fittings no longer serve a purpose, but to remind us of legacy technology. Some day in the future there may be a photograph of an empty terminal strip in another chapter of this book to explain how wired "legacy" instruments have all but been replaced by wireless (radio) instruments! Let the ghosts of the past speak, dear reader, testifying to the endless march of technological evolution.

current (I) is scaled into an output pressure (P), [8]. In most cases, this means an input of 4-20 mA DC and an output of 3-15 psig, but alternative ranges do exist.

An example of an I/P transducer manufactured by Foxboro is the model E69, shown in Figure 3.35.

Figure 3.35: An industrial I/P transducer

Two pressure gauges indicate supply and output pressure, respectively. Wires convey the 4-20 mA electrical signal into the coil unit inside the transducer.

Figure 3.36 is view with the cover removed that shows the balancing mechanism used to generate a pneumatic pressure signal from the electric current input. The baffle/nozzle may be seen at the left of the mechanism, the nozzle located at the end of a bent tube, facing the flat baffle on the surface of the circular coil unit.

[8] P/I, E/P, and P/E converters also exist, where E stands for voltage

Figure 3.36: Internals of an industrial I/P transducer

As electric current passes through the coil, it produces a magnetic field which reacts against a permanent magnet's field to generate a torque. This torque causes rotation against the restraint of a spring, with the baffle connected to the rotating assembly. Thus, the baffle moves like the needle of an analog electric meter movement in response to current: the more current through the coil, the more the coil assembly moves (and the baffle moves with it).

The nozzle faces this baffle, so when the baffle begins to move toward the nozzle, backpressure within the nozzle rises. This rising pressure is amplified by the relay, with the output pressure applied to a bellows. As the bellows expands, it draws the nozzle away from the advancing baffle, achieving balance by matching one motion (the baffle's) with another motion (the nozzle's).

Figure 3.37 is a closer view that shows the baffle and nozzle in detail.

Figure 3.37: Baffle and nozzle of an industrial I/P transducer

Thus, the self-balancing mechanism of the Foxboro model E69 transducer uses the *motion-balance* principle rather than the force-balance principle as applied in the Foxboro model 13 differential pressure transmitter. Instead of an input force precisely balancing an output force, constraining motion to a negligible degree, this mechanism allows the input freely move, matching that motion with a corresponding output motion to maintain a near-constant baffle/nozzle gap.

Interestingly the model E69 transducer employs the same pneumatic amplifying relay used in virtually every Foxboro pneumatic instrument (shown in Figure 3.38).

Figure 3.38: Industrial pneumatic amplifying relay

As in all instruments, the zero adjustment works by *adding or subtracting* a quantity, while the span adjustment works by *multiplying or dividing* a quantity. In the Foxboro model E69 transducer, the quantity in question is motion, since this is a motion-balance mechanism. The zero adjustment adds or subtracts motion by offsetting the position of the nozzle closer to or further away from baffle. A close-up photograph of the zero adjustment screw in 3.39 shows it pressing against a tab to rotate the mounting baseplate upon which the coil unit is fixed. Rotating this baseplate adds or subtracts angular displacement to/from the baffle's motion.

Figure 3.39: Industrial pneumatic amplifying relay zero adjustment nut

The span adjustment consists of changing the position of the nozzle relative to the baffle's center of rotation, so that a given amount of rotation equates to a different amount of balancing motion required of the nozzle. Figure 3.40 shows that this adjustment consists of a pair of nuts locking the base of the bellows unit at a fixed distance from the baffle's center of rotation. Changing this distance alters the effective radius of the baffle as it swings around its center, therefore altering the gain (or span) of the motion-balance system.

Figure 3.40: Industrial pneumatic amplifying relay span adjustment nuts

### 3.9.2 Fisher model 546 electro-pneumatic transducer

The Fisher model 546 I/P transducer performs the same signal-conversion function (mA into psig) as the Foxboro model E69, but it does so differently [4]. Figure 3.41 shows the internal mechanism of the model 546 transducer with its cover removed.

Figure 3.41: Another industrial I/P transducer

This particular instrument's construction tends to obscure its function, so Figure 3.42 is an illustrative diagram to describe its operation.

Figure 3.42: Diagram of internals of an I/P transducer

The heart of this mechanism is an iron beam, located between the poles of a permanent magnet assembly, and centered within an electromagnet coil (solenoid). Current passing through the electromagnet coil imparts magnetic poles to the ends of the beam. Following the arrow head/tail convention shown in the coil windings (the dots versus X marks) representing conventional flow vectors pointing out of the page (top) and going into the page (bottom) for the coil wrapped around the beam, the right-hand rule tells us that the beam will magnetize with the right-hand side being "North" and the left-hand side being "South." This will torque the beam clockwise around its pivot point (fulcrum), pushing the right-hand side down toward the nozzle.

Any advance of the beam toward the nozzle will increase nozzle backpressure, which is then fed to the balancing bellows at the other end of the beam. That bellows provides a restoring force to the beam to return it (nearly) to its original position. The phenomenon of an input force being counter-acted by a balancing force to ensure minimum motion is the defining characteristic of a

*force-balance* system. This is the same basic principle applied in the Foxboro model 13 differential pressure transmitter: an input force countered by an output force.

By examining the diagram carefully, it can be seen that this instrument's amplifying relay is not located within the force-balance feedback loop. The nozzle's backpressure is *directly* fed back to the balancing bellows with no amplification at all. A relay does exist, but its purpose is to provide a modest (approximately 2:1) pressure gain to raise the nozzle backpressure to standard levels (3-15 psig, or 6-30 psig).

Figure 3.43 shows the solenoid coil, force beam, and nozzle. With close inspection, the copper-colored windings of the coil buried within the mechanism can be seen. The zero-adjustment spring is located above the beam, centered with the nozzle (below the beam).

Figure 3.43: Internals of an I/P transducer

Fisher manufactured these I/P transducers with two different pneumatic ranges: 3-15 psig and 6-30 psig. The mechanical difference between the two models was the size of feedback bellows used in each. In order to achieve the greater pressure range (6-30 psig), a *smaller* feedback bellows was used. This may seem backward at first, but it makes perfect sense when examining the operation of the force-balance mechanism. In order to generate a greater air pressure for a given electric current through the coil, the air pressure must be placed at a mechanical *disadvantage* to force it to rise higher than it ordinarily would in achieving balance. One way to do this is to decrease the effective area of the bellows[9], so that it takes a greater air pressure to generate the same amount of balancing force on the beam.

[9]Recall the mathematical relationship between force, pressure, and area: $F = PA$. If a greater pressure ($P$) is desired to generate the same force ($F$) as before, the area ($A$) must decrease upon which that pressure acts.

A 3-15 psig bellows (left) is contrasted against a 6-30 psig bellows (right) in this pair of photographs in Figure 3.44.

(a) A 3-15 psig bellows (b) A 6-30 psig bellows

Figure 3.44: Bellows assemblies

The span adjustment for this I/P transducer functions by varying the permanent-magnetic field strength acting against the beam's electro-magnetic field. Adjustment occurs through the use of a magnetic *shunt*: an iron plate moved closer to or further away from the permanent magnets, providing an alternate (shunt, or bypass) path for magnetic flux away from the force beam. Moving the shunt further away from the magnets strengthens the magnetic field "seen" by the beam, resulting in a multiplication of force on the beam and therefore a multiplication of output pressure. Moving the shunt closer to the magnets weakens the magnetic field "seen" by the beam, thereby dividing the reaction force and also the output pressure.

A view of the mechanism's other side in Figure 3.45 reveals the magnetic shunt plate, complete with an instructional arrow showing the correct direction to turn the adjustment screw to increase output span.

Figure 3.45: Internals of an I/P transducer showing span adjustment screw

### 3.9.3 Fisher-Rosemount model 846 electro-pneumatic transducer

The Fisher-Rosemount model 846 is a more modern I/P transducer than either the Foxboro model E69 or the Fisher model 546. It employs neither the force-balance nor the motion-balance principle in its operation, which makes it unique to analyze. This I/P unit is also unique in that it features a modular design allowing very convenient replacement of internal components when in service.

Figure 3.46 shows three model 846 I/P transducers attached to a metal panel, below a set of five Rosemount model 1151 pressure transmitters.

Figure 3.46: An array of industrial I/P transducers below an array of pressure transmitters

A closer photograph in Figure 3.47 reveals the unit in more detail.

Figure 3.47: An industrial I/P transducer

When one of the end-covers is unscrewed, the internal workings of the I/P may be removed as a single module. Both the removed module and the housing are shown in Figure 3.48.

Figure 3.48: An industrial I/P transducer's internal electronics

Shown separately in Figure 3.49, the module's current input terminals connect with matching pins in the housing. Even the zero and span adjustment potentiometers on the module circuit board are equipped with Velcro (hook and loop) pads, matching with pads attached to calibration screws on the housing. This simple yet effective mechanical coupling allows screws located on the exterior housing to adjust resistances on the module's circuit board for zero and span calibration, yet without exposing those delicate potentiometers to ambient weather conditions.

(a) Fisher 846 housing (b) Fisher 846 circuit board

Figure 3.49: Internals of a Fisher 846 I/P converter

Pneumatic (air) connections are made to the housing through standard 1/4 inch female NPT pipe threads. Compressed air is passed to the module (and from the module back out to the housing) through ports, sealed from each other by O-rings[10] located on the module.

The primary benefit of this modular design is ease of maintenance in the field. If a module fails for any reason, it may be very quickly removed and replaced, with no disconnection and re-connection of signal wires or pneumatic tubes necessary.

As mentioned before, the feedback mechanism for this particular I/P transducer employs neither the force-balance nor the motion-balance principle. Rather, the negative feedback and balancing of this unit is done electronically rather than mechanically. Figure 3.50 shows how this works.

Figure 3.50: Diagram of the workings of an I/P converter

An electronic pressure sensor continuously monitors the output pressure, with its signal being electronically compared to the input (4-20 mA) signal by the control circuit to check for equivalence. If the output does not match the input, the control circuit drives the deflector motor with more or

[10] It is quite easy to dislodge these small-section, large-diameter O-rings from their respective grooves during re-assembly of the unit. Be very careful when inserting the module back into the housing!

less current as needed, to deflect the air jet more or less as it exits one nozzle and is intercepted by the other to stimulate the pneumatic amplifying relay. Thus, the "balancing" internal to this I/P is done electronically rather than mechanically as it was in the other I/P relays (Foxboro model E69, Fisher model 546) explored in this section.

Electronic components are less likely to drift in their calibration, and are less susceptible to the effects of mechanical vibration and mounting orientation, than mechanical balancing components.

## 3.10 Proper care and feeding of pneumatic instruments

Perhaps the most important rule to obey when using pneumatic instruments is to *maintain clean and dry instrument air.* Compressed air containing dirt, rust, oil, water, or other contaminants will cause operational problems for pneumatic instruments. First and foremost is the concern that tiny orifices and nozzles inside the pneumatic mechanisms will clog over time. Clogged orifices tend to result in decreased output pressure, while clogged nozzles tend to result in increased output pressure. In either case, the "first aid" repair is to pass a welding torch tip cleaner through the plugged hole to break loose the residue or debris plugging it.

Moisture in compressed air tends to corrode metal parts inside pneumatic mechanisms. This corrosion may break loose to form debris that plugs orifices and nozzles, or it may simply eat through thin diaphragms and bellows until air leaks develop. Grossly excessive moisture will cause erratic operation as "plugs" of liquid travel through thin tubes, orifices, and nozzles designed only for air passage.

A common mistake made when installing pneumatic instruments is to connect them to a general-service ("utility") compressed air supply instead of a dedicated instrument-service compressed air system. Utility air systems are designed to supply air tools and large air-powered actuators with pneumatic power. These high-flow compressed air systems are often seeded with antifreeze and/or lubricating chemicals to prolong the operating life of the piping and air-consuming devices, but the same liquids will wreak havoc on sensitive instrumentation. Instrument air supplies should be sourced by their own dedicated air compressor(s), complete with automatic air-dryer equipment, and distributed through stainless steel, copper, or plastic tubing (never black iron or galvanized iron pipe!).

Moisture in an instrument air system was the root cause of an event that happened at one oil refinery. Someone on the operations staff decided they would use 100 psig instrument air to purge a process pipe filled with acid. Unfortunately, the acid pressure in the process pipe exceeded 100 psig, and as a result acid flushed backward into the instrument air system. Within days most of the pneumatic instruments in that section of the refinery failed due to accelerated corrosion of metal components within the instruments. The total failure of multiple instruments over such a short time could have easily resulted in a disaster, but fortunately the crisis was minimal. Once the first couple of faulty instruments were disassembled after removal, the cause of failure became evident and the technicians took action to flush the lines of acid before too many more instruments suffered the same fate.

Pneumatic instruments must be fed compressed air of the proper pressure as well. Just like electronic circuits which require power supply voltages within specified limits, pneumatic instruments do not operate well if their air supply pressure is too low or too high. If the supply pressure is too low, the instrument cannot generate a full-scale output signal. If the supply pressure is too high, internal failure may result from ruptured diaphragms, seals, or bellows. Many pneumatic instruments

are equipped with their own local pressure regulators directly attached to ensure each instrument receives the correct pressure despite pressure fluctuations in the supply line.

Another "killer" of pneumatic instruments is mechanical vibration. These are precision mechanical devices, so they do not generally respond well to repeated shaking. At the very least, calibration adjustments may loosen and shift, causing the instrument's accuracy to suffer. At worst, actual failure may result from component breakage[11].

## 3.11 Trade-offs of pneumatic instruments

The disadvantages of pneumatic instruments are painfully evident to anyone familiar with both pneumatic and electronic instruments. Sensitivity to vibration, changes in temperature, mounting position, and the like affect calibration accuracy to a far greater degree for pneumatic instruments than electronic instruments. Compressed air is an expensive utility – much more expensive per equivalent watt-hour than electricity – making the operational cost of pneumatic instruments far greater than electronic. The installed cost of pneumatic instruments can be quite high as well, given the need for special (stainless steel, copper, or tough plastic) tubes to carry supply air and pneumatic signals to distant locations. The volume of air tubes used to convey pneumatic signals over distances acts as a low-pass filter, naturally damping the instrument's response and thereby reducing its ability to respond quickly to changing process conditions. Pneumatic instruments cannot be made "smart" like electronic instruments, either. With all these disadvantages, one might wonder why pneumatic instruments are still used at all in modern industry.

Part of the answer is legacy. For an industrial facility built decades ago, it makes little sense to replace instruments that still work just fine. The cost of labor to remove old tubing, install new conduit and wires, and configure new (expensive) electronic instruments often is not worth the benefits.

However, pneumatic instruments actually enjoy some definite technical advantages which secure their continued use in certain applications even in the $21^{st}$ century. One decided advantage is the *intrinsic safety* of pneumatic field instruments. Instruments that do not run on electricity cannot generate electrical sparks. This is of utmost importance in "classified" industrial environments where explosive gases, liquids, dusts, and powders exist. Pneumatic instruments are also self-purging. Their continual bleeding of compressed air from vent ports in pneumatic relays and nozzles acts as a natural clean-air purge for the inside of the instrument, preventing the intrusion of dust and vapor from the outside with a slight positive pressure inside the instrument case. It is not uncommon to find a field-mounted pneumatic instrument encrusted with corrosion and filth on the outside, but factory-clean on the inside due to this continual purge of clean air. Pneumatic instruments mounted inside larger enclosures with other devices tend to protect them all by providing a positive-pressure air purge for the entire enclosure.

Some pneumatic instruments can also function in high-temperature and high-radiation environments that would damage electronic instruments. Although it is often possible to "harden"

[11] Having said this, pneumatic instruments can be remarkably rugged devices. A field-mounted pneumatic controller was once attached to the same support as a badly cavitating control valve. The vibrations of the control valve transferred to the controller through the support, causing the baffle to hammer repeatedly against the nozzle until *the nozzle's tip had been worn down to a flattened shape.* Remarkably, the only indication of this problem was the fact the controller was having some difficulty maintaining setpoint. Other than that, it seemed to operate adequately! Electronic devices may have not fared as well, unless completely "potted" in epoxy.

electronic field instruments to such harsh conditions, pneumatic instruments are practically immune by nature.

An interesting feature of pneumatic instruments is that they may operate on compressed gases other than air. This is an advantage in remote natural gas installations, where the natural gas itself is sometimes used as a source of pneumatic "power" for instruments. So long as there is compressed natural gas in the pipeline to measure and to control, the instruments will operate. No air compressor or electrical power source is needed in these installations. What *is* needed, however, is good filtering equipment to prevent contaminants in the natural gas (dirt, debris, liquids) from causing problems within the sensitive instrument mechanisms.

# References


[1] D. R. Patrick and S. R. Patrick, Pneumatic Instrumentation. Albany, NY: Delmar Publishers, Inc., 1993.

[2] Compatability of Analog Signals for Electronic Industrial Process Instruments, International Society of Automation Std. 50.00.01, 2010.

[3] E69 Current-to-Pneumatic Signal Converter, MI 018-430, The Foxboro Company, Foxboro, MA, 1995.

[4] Type 546, 546S, and 546NS Electro-Pneumatic Transducers, Form 1783, Fisher Controls International, Marshalltown, IA, 2005.

[5] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1.

[6] ——, Instrument Engineers Handbook: Process Control and Optimization, 4th ed. Boca Raton, FL: CRC Press, 2006, vol. 2.

[7] ——, Instrument Engineers Handbook: Process Software and Digital Networks, 3rd ed. Boca Raton, FL: CRC Press, 2002, vol. 3.

[8] Type 846 Current-to-Pressure Transducers Instruction Manual, D102005X012, Fisher Controls International, Marshalltown, IA, 2008.

# Chapter 4

# Digital Networks

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Represent numeric and text data in the following different digital representations: binary, octal, hexadecimal, binary-coded decimal, ASCII
2. Convert between the following number systems: decimal, binary, octal, hexadecimal, binary-coded decimal
3. Given a particular ADC or DAC, determine the output for a given input
4. Explain how sampling rate of a converter affects system performance
5. Describe the function of each of the seven levels of the Open Systems Interconnect (OSI) model:
   (a) Application
   (b) Presentation
   (c) Session
   (d) Transport
   (e) Network
   (f) Data Link
   (g) Physical
6. Discuss the function and application of the following types of digital communication methods from the OSI model perspective:
   (a) Serial communication
      i. TIA-232
      ii. TIA-422
      iii. TIA-485
   (b) HART
   (c) Foundation Fieldbus
   (d) PROFIBUS
   (e) Modbus
   (f) ControlNet

(g) DeviceNet
(h) AS-i
(i) Ethernet

7. Discuss the need for bus terminators and explain their application.

## 4.1 Digital communication

The advent of digital electronic circuitry has brought a steady stream of technological progress to industrial instrumentation. From early applications of digital computing in the 1960's to the first distributed control systems (DCS) in the 1970's to the "smart" transmitter revolution of the 1980's, digital technology has expanded on the capabilities and information-sharing ability of measuring and control instruments. It is the purpose of this chapter to give a general overview of digital technology as it applies to data acquisition (measuring and recording process data) and digital communication, highlighting some of the more common standards used in industry. Subsequent chapters will be devoted to more in-depth discussions of specific digital instrumentation standards.

One of the greatest advantages of digital technology over analog is the ability to communicate vast amounts of data over a limited number of data channels. In the world of 4-20 mA signaling (or 3-15 PSI signaling, for that matter!) each pair of wires could communicate only *one* variable. In the world of digital networks, one pair of wires can communicate a nearly limitless number of variables, the only limit being the *speed* of that data communication[1].

This one-signal-per-channel limit of 4-20 mA analog signals represents a technological "bottleneck" restricting data transfer between instruments and control systems. While it certainly is possible to devote a dedicated wire pair to each and every variable in an instrument system, this is very expensive to do. It is particularly cumbersome for instruments generating multiple variables of measurement, such as Coriolis flowmeters which simultaneously measure process fluid mass flow rate, fluid density, and fluid temperature; or "smart" valve positioners which continuously measure the stem position, actuator pressure(s), supply pressure, and temperature of a control valve. The data-rich capabilities of digital field instruments demands a digital form of communication to overcome the "bottleneck" of analog 4-20 mA signals.

Rosemount's HART standard was a valiant attempt to provide the "best of both worlds" in industrial instrumentation. With HART digital signals superimposed on 4-20 mA analog signals, one could retain the simplicity and certainty of analog signaling while enjoying the multi-variable communication benefits that digital had to offer. However, HART communication is rather slow by any standard, restricting its use to maintenance (range changes, diagnostic data polling) and process control for slow processes only.

There exist many different digital communication standards (generally called "fieldbuses") designed to interconnect industrial instruments. An incomplete list is shown here:

- HART
- Modbus
- FOUNDATION Fieldbus
- Profibus PA
- Profibus DP
- Profibus FMS
- AS-i
- CANbus
- ControlNET
- DeviceNet
- Ethernet/IP (Industrial Protocol)

[1] The technical term for the "speed limit" of any data communications channel is *bandwidth*, usually expressed as a frequency (in Hertz).

- LonWorks
- BACnet

The utility of digital "fieldbus" instruments becomes apparent through the host system these instruments are connected to (typically a *distributed control system*, or *DCS*). Fieldbus-aware host systems usually have means to provide instrument information (including diagnostics) in very easy-to-navigate formats.

For example, the following screenshot shows the field instrument devices connected to a small-scale DCS used in an educational lab. Each instrument appears as an icon, which may be explored further simply by pointing-and-clicking with the mouse[2]:

Figure 4.1: Device Manager layout for a fieldbus network

Another application of digital communication technology to industrial measurement and control is what is generally referred to as a *SCADA* ("Supervisory Control And Data Acquisition") system. A SCADA system might be thought of as a distributed control system (DCS) spread over a

[2]The host system in this case is an Emerson DeltaV DCS, and the device manager software is Emerson AMS.

geographically large area, such as across the span of a city or even across national borders. Typical applications of SCADA technology include:

- Electric power generation and distribution (power line, substation) systems
- Water and wastewater treatment and distribution (water line, pumping stations) systems
- Gas and oil exploration and distribution (pipeline) systems
- Large-scale agricultural (irrigation, harvesting) systems
- Storage tank monitoring systems

Process data in a SCADA system is sensed by various measurement devices (transmitters), converted to digital form by a device called an *RTU* ("Remote Terminal Unit"), and communicated to one or more *MTUs* ("Master Terminal Units") at a central location where human operators may monitor the data and make command decisions.

If the flow of information is one-way (simplex, from measurement devices to human operators), the system is more properly referred to as a *telemetry* system rather than a SCADA system. "SCADA" implies two-way (duplex) information flow, where human operators not only monitor process data but also issue commands back to the remote terminal units to effect change.

The saying "necessity is the mother of invention" certainly holds true for the development of remote telemetry and SCADA systems. The need for remote monitoring and control of electric power distribution systems led to the development of "power line carrier" analog telemetry systems as far back in time as the 1940's. These systems superimposed high-frequency (50 kHz to 150 kHz) "carrier" signals on low-frequency (50 Hz and 60 Hz) power line conductors to communicate such basic information as human voice (like a telephone network, only dedicated for power system operators), power flow (wattmeter, MVAR meter) monitoring, and protective relay (automatic trip) controls. These telemetry systems were among the first to enjoy the benefits of digital technology in the 1960's. Large-scale electric power systems simply cannot be operated safely and effectively without remote data monitoring and control, and this operational necessity pushed technological development of telemetry and SCADA systems beyond their small-scale (industrial manufacturing) counterparts.

Whether it is a "smart" temperature transmitter, a panel-mounted process controller with Ethernet communication capability, a variable-speed electric motor drive with Modbus signaling, a large-scale DCS controlling an oil refinery, or a SCADA system monitoring a power distribution system spanning across national borders, digital measurement and communication is an essential part of modern measurement and control systems. This chapter focuses on some of the basic principles of digital data formatting and communication, referencing practical applications wherever possible.

## 4.2 Digital representation of numerical data

Process measurements are usually of an analog nature: the temperature of a furnace, the rate of fluid flow through a pipe, the pressure of a fluid, etc. These data are all analog quantities: infinitely variable, not discrete. Although some process measurements may be discrete (e.g., counting the number of units passed by on a conveyor belt), the majority of measurements in the industrial world are analog.

In order for any digital device to successfully interface with an analog signal, that signal must be *digitized* by means of an *analog-to-digital converter* or *ADC*. This section will not endeavor to explore the intricate details of ADC circuitry, but merely to discuss ADC performance in the context

of process measurements.

Many of the concerns discussed in this section are relevant to circuits converting digital values into analog signals as well. These *digital-to-analog converters*, or *DAC*s, are generally used to produce the analog drive signals required of final control elements (e.g., the output of a digital PID controller driving a 4-20 mA analog signal to a control valve positioner).

### 4.2.1 Integer number formats

An *integer* is any whole number or its negative counterpart (e.g., 2, 7, 241, 0, -35, -1022). All modern digital systems represent integer quantities using *binary* numeration, where integer numbers consists of strings of "bits," each bit having two possible values: 0 or 1. Unlike the *decimal* numeration system most people are familiar with where the positional values are powers of ten, the positional values of binary numeration are powers of two. The following example shows how the integer number six hundred and three is represented in both decimal and binary formats:

**Six hundred and three**

$$\underset{10^2}{\underline{6}}\ \underset{10^1}{\underline{0}}\ \underset{10^0}{\underline{3}} = (6 \times 10^2) + (3 \times 10^0) \qquad \textit{Decimal representation}$$

$$\underset{2^9}{\underline{1}}\ \underset{2^8}{\underline{0}}\ \underset{2^7}{\underline{0}}\ \underset{2^6}{\underline{1}}\ \underset{2^5}{\underline{0}}\ \underset{2^4}{\underline{1}}\ \underset{2^3}{\underline{1}}\ \underset{2^2}{\underline{0}}\ \underset{2^1}{\underline{1}}\ \underset{2^0}{\underline{1}} = 2^9 + 2^6 + 2^4 + 2^3 + 2^1 + 2^0 \qquad \textit{Binary representation}$$

Figure 4.2: Decimal and binary number formats

The largest integer value representable in this format where each place has a positive weight value is the base raised to the power of the number of places, minus one. So, for a three-digit decimal number, the largest possible value is $10^3 - 1$ ($1000 - 1 = 999$). For a ten-bit binary number, the largest possible value is $2^{10} - 1$ ($1024 - 1 = 1023$).

The beauty of binary representation is that the individual "bits" are easy to encode in physical form. "1" and "0" values may take the form of voltage levels ("high" or "low"), magnetization states, AC signal frequencies ("high" or "low"), pulses of light, or any other medium capable of representing diametric states. This makes binary a natural form of numerical representation for computers.

When representing negative integers, the convention for written numerals is to prepend a "negative" symbol to the left-hand side of the number. In digital machines, this may be emulated by including an extra "sign bit" having a value of either 0 or 1 indicating whether the number is positive or negative. A more common method for negative value representation in digital machines, however, is called *two's complement* notation.

Two's complement notation works by assigning the most significant digit a negative weight value. To use a ten-bit binary example showing the number negative three hundred and ninety six:

**Negative three hundred and ninety six**
*(Two's complement binary notation)*

$$\underset{-2^9}{1}\ \underset{2^8}{0}\ \underset{2^7}{0}\ \underset{2^6}{1}\ \underset{2^5}{1}\ \underset{2^4}{1}\ \underset{2^3}{0}\ \underset{2^2}{1}\ \underset{2^1}{0}\ \underset{2^0}{0} = -2^9 + 2^6 + 2^5 + 2^4 + 2^2$$

Figure 4.3: Two's complement negative number binary format

Using this system to represent integer values, the greatest positive number possible is when all bits are "1" with the exception of the most significant bit (MSB). In the case of this ten-bit field, this number would be 0111111111, equaling $2^9 - 1$ or 511. The greatest negative number possible is when the only "1" bit is the MSB. In the case of this ten-bit integer, this number would be 1000000000, equaling $-2^9$ or -512. Thus, a ten-bit two's complement integer may represent values ranging from -512 to +511, inclusive.

Whether binary numbers in a computer use the two's complement system of notation or not is often the choice of the human being programming the computer. For this reason, it is important to define whether a binary integer used within a computer system to represent data is *signed* or *unsigned*. If the integer in question is "signed," it means the binary number is capable of representing both positive and negative integer values (usually via two's complement notation). If the integer in question is "unsigned," it means the binary number is capable only of representing positive values.

For example, an "unsigned" 16-bit integer has a possible numerical value range of 0 to 65535. However, the same 16-bit field – if designated as "signed" using two's complement representation – has a range of -32768 to +32767.

Digital computers usually organize binary numbers as clusters of bits with a set number of bits. A cluster of eight bits is called a *byte*. The standard number of bits for most modern computers exceeds one byte, with 16, 32, or even 64 bits' length being common. Whatever the standard bit-length of a computer's binary number field may be, it is defined for that computer as a *word*. Thus, if a digital computer is built to handle binary data in 32-bit clusters, one "word" of data within that computer will be 32 bits in length. A "double-word" value for that computer is one where the binary number consists of 64 bits.

Given the typically large bit-lengths of numbers in modern computers, it is conventional to express such quantities in "shorthand" form using *octal* or *hexadecimal* numeration[3]. The positional values of octal numeration are powers of eight. The basic idea of octal is that every three bits of a binary number are represented by a single octal character ranging in value from 0 through 7 (0,1,2,3,4,5,6,7). The positional values of hexadecimal numeration are powers of sixteen. For example, the 16-bit binary number $1001001111011110_2$ is equivalent to the six-digit octal number $111736_8$. The basic idea of hexadecimal is that every four bits (half of a byte, or one *nibble*) of a binary number are represented by a single hex character ranging in value from 0 through F (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F). For example, the 16-bit binary number $1001001111011110_2$ is equivalent to the four-digit hexadecimal number $93DE_{16}$::

[3]Note that the radix, or base, of the number is always indicated as a subscript or additional character for numbers in every system other than decimal to avoid confusion. Octal numbers may be represented by the subscript "8" or using a "o" character either before or after the numerals. Hexadecimal numbers may be represented by the subscript "16" or using a "h" character either before or after the numerals. Though a subscript or additional character may be used with decimal numbers, it is often omitted.

Figure 4.4: Binary and hexadecimal number formats

Quite obviously, the octal number $111736_8$ or the hexadecimal number $93DE_{16}$ is a lot easier for human beings to manage than the binary number $1001001111011110_2$. Computer systems providing users with live "views" of numerical values often format those values using octal or hexadecimal numeration, just to simplify the task of interpretation. Conversion from binary to octal integers is simple: starting at the right end of the number and working to the left, separate the number into groups of three binary digits, adding leading zeros at the left end as appropriate. Then, simply convert each group of three binary digits to its equivalent octal character. To convert from hexadecimal to binary, write the equivalent four-bit binary representation for each hexadecimal character. Conversion from binary to hexadecimal integers is simple: starting at the right end of the number and working to the left, separate the number into groups of four binary digits, adding leading zeros at the left end as appropriate. Then, simply convert each group of four binary digits to its equivalent hexadecimal character. To convert from hexadecimal to binary, write the equivalent four-bit binary representation for each hexadecimal character.

Another type of representation uses the binary equivalent of the decimal numeric values to encode the numerals. Appropriately, this is called *binary-coded decimal (BCD)*[4]. Since the decimal numeric values of 0 through 9 can be represented by the binary four-bit words $0000_2$ through $1001_2$, these values are used in a digital representation. For example, the number 3275 would be represented as $0011001001110101_{BCD}$. Each of the four-bit words within the 16-bit representation corresponds to one digit of the original number.

### 4.2.2 Fixed-point number formats

The main limitation of integer numbers is their inability to represent quantities between whole numbers such as fractions. For example, a 4-bit integer number may have one of sixteen possible values (zero through fifteen), but it can never represent two-*and-a-half*.

One solution to this limitation is to re-assign positional values to the bits of a binary number, giving some of those bits fractional weights. For example, here is a ten-bit binary number field with a "binary point" separating whole-weighted bits from fractional-weighted bits, representing the quantity 70.375:

[4]Binary-coded decimal numbers are represented by the subscript "BCD" to avoid confusion with binary

Figure 4.5: Non-integer binary number format

This form of binary notation is referred to as *fixed-point*, because the "binary point" is fixed in its location. Like any integer variable, this 10-bit binary number consisting of seven whole-weighted bits and three fractional-weighted bits has a definite range of possible values: in this particular case, from 0 to 127.875.

While fixing a "binary point" in the bit field escapes the limitation of whole-number resolution, a more sophisticated approach known as *floating-point* representation is a far more flexible way of representing fractional quantities in binary form. This is how most digital computers represent real-world analog quantities in digital form, and is the topic of the next subsection.

### 4.2.3 Floating-point number formats

Integer numbers, while very commonly used in computers for representing data, are insufficient for tasks requiring the representation of fractional (non-whole) values. Most industrial data points fall into this category, whether it be a temperature measurement (e.g., 428.21 $^o$C), a gas concentration (e.g., 5.09 ppm of carbon monoxide), a control valve position (e.g., 27.58% open), or a motor speed (e.g., 1592.1 RPM). What is needed is a form of binary representation capable of expressing fractional quantities.

One way to do this is to use *floating-point* number format, where binary quantities are represented using a form resembling scientific notation. Recall how scientific notation is used to represent decimal numbers:

$$1002349 = 1.002349 \times 10^6$$

$$678.331 = 6.78331 \times 10^2$$

$$0.00004532 = 4.523 \times 10^{-5}$$

The power-of-ten multiplier has the effect of "floating" the decimal point to any position desired, so that the significant digits of the number may be expressed in "normalized" form (between 1 and 10). This is where the term "floating-point" comes from.

In binary floating-point notation, a string of bits is used to represent the significant bits of the number ($m$), and a second string of bits to represent the power-of-two exponent ($E$):

$$\pm 1.m \times 2^E$$

The IEEE has standardized[5] floating-point formats, including one that is 32 bits in length ("single-precision") and another that is 64 bits in length ("double-precision"). In the IEEE standard, one bit is reserved for the sign of the number (0 for positive and 1 for negative), a certain number of bits for the power-of-two exponent, and the rest of the bits for the *mantissa* (the fractional portion of the normalized value). Both formats are shown here:

Figure 4.6: IEEE floating point number formats

A third floating-point IEEE standard called *extended* uses 80 total bits: 1 for the sign, 15 for the exponent, and 64 for the mantissa.

Not only can the IEEE floating-point binary number format express an incredibly wide range of values, but it is also capable of expressing infinity (both positive and negative!) as well as an error code referred to as *Not a Number* (NaN). These representations are important for handling the results of calculations such as division-by-zero and the square-roots or logarithms of negative numbers.

Specific details on IEEE floating-point representation and arithmetic rules extend beyond the scope of this book. Michael L. Overton's excellent paper "Floating Point Representation" is a good primer on the subject.

### 4.2.4 Example of industrial number formats

A human-machine interface (HMI) panel providing graphical view and control of data within a programmable logic controller (PLC) illustrates how different digital data formats are used in industrial systems. HMI panels are nothing more than special-purpose computers equipped with graphic display screens for displaying data to human operators and pushbuttons (sometimes touch-sensitive graphic screens) for receiving input from human operators.

The following illustration shows how an HMI may be used with a PLC to control electric power to a motor:

[5] ANSI/IEEE standard 754-1985

Figure 4.7: Diagram of an industrial control system

When the operator presses a button on the HMI display to tell the motor to start, the HMI panel sends digital data to the PLC instructing the program within the PLC to activate the motor. The PLC, in turn, receives this data from the HMI and determines whether or not[6] to start the motor. If the motor starts, the PLC sends data back to the HMI confirming the motor's "run" status, which the HMI indicates to the operator in graphic form.

The configuration software used to set up communication between an HMI panel and the control device typically requires the human programmer (engineer or technician) to specify which data points inside the control device will be accessed by the HMI panel. This is specified in a list called the *tag name database*. A screen-shot of the tag name database for a particular HMI panel configured to communicate with a PLC using Modbus protocol appears in Figure 4.8.

[6]This motor may be "interlocked" to prevent start-up if certain conditions are not met, thereby automatically prohibiting the operator's instruction to start.

Tag Name Database

Device Name: PLC_01

Fill Down

☑ Display System Tags ☐ Highlight

☑ Display Internal Tags ☐ Highlight

☐ Highlight Unused Tags

| No. | Tag Name | Data Type | PLC Address | Device Name | Attribute |
|---|---|---|---|---|---|
| 1 | START_PUSHBUTTON | Discrete | 10024 | PLC_01 | R |
| 2 | STOP_PUSHBUTTON | Discrete | 10031 | PLC_01 | R |
| 3 | MOTOR_RUN | Discrete | 00005 | PLC_01 | R/W |
| 4 | MOTOR_RUN_TIMER | Unsigned int 32 | 40010 | PLC_01 | R/W |
| 5 | START_COUNTER | Unsigned int 16 | 40242 | PLC_01 | R/W |
| 6 | MOTOR_TEMPERATURE | Floating PT 32 | 30008 | PLC_01 | R |
| 7 | ERROR_MESSAGE | Ascii String | 40560 | PLC_01 | R/W |
| 8 | MOTOR_SPEED | Floating PT 32 | 30017 | PLC_01 | R |

Figure 4.8: Tags in a HMI database pointing to PLC memory registers

This database table shows the "tag name" assigned to each data point (for easy programming reference), the type of digital data format used, the memory address within the control device (the PLC), the network name of the PLC, and whether the data point is read-only (R), write-only (W), or read/write (R/W) from the perspective of the HMI panel.

Note how the first three tag names are all classified as *discrete* data. The word "discrete" here is synonymous with *Boolean*, because it represents single-bit data having only two possible states: on or off, 1 or 0. As the tag names imply, these particular data points are associated with simple on/off states: a pushbutton for starting a motor, another pushbutton for stopping a motor, and a bit representing the motor's "run" status. Once these tag names, data types, and device address values have been established, the programmer may then link the tags to graphic objects on the screen of the HMI panel. Perhaps a square box with the words "Motor start" may serve to activate the `START_PUSHBUTTON` bit and cause the PLC to energize the electric motor. An animated graphic image of a motor may be programmed to show rotation if the `MOTOR_RUN` bit is set, and show no motion if that same bit is cleared.

Four other data types appear in this tag name database, each one representing different forms of information:

- Unsigned 16-bit integer
- Unsigned 32-bit integer
- Floating-point 32-bit
- ASCII string

Two "unsigned integer" data points represent motor run time and a "start counter" for keeping count of how many times the motor has been started. The timer is a 32-bit integer value, while the counter is a 16-bit integer value. Being unsigned integers (no two's complement notation), the numerical range of the timer is 0 to 4,294,967,295 ($2^{32} - 1$) and the numerical range of the counter is 0 to 65,535 ($2^{16} - 1$). The large range afforded by the timer's 32-bit word length allows up to 136.2 years' worth of recorded continuous motor run time, assuming the resolution of this timer is

one second.

Two of these data points use 32-bit floating-point format to represent analog motor data: motor temperature and motor speed. Given the units of measurement common to these types of variables (degrees Celsius or Fahrenheit, revolutions per minute), a resolution of ±1 as given by an integer number would probably be insufficient for the task. Instead, the operator should be able to read *fractions* of a degree in temperature, and *fractions* of an RPM motor speed. This necessity to display numerical values with resolution less than 1 unit dictates the use of the floating-point data type[7].

The last data type represented in this tag name database is an "ASCII string," which means a series of bytes (8-bit groupings) representing text data, such as a message. In this particular case, the tag name itself betrays the purpose of the data: `ERROR_MESSAGE`. The PLC obviously has the ability to generate text messages and send them to the HMI panel to communicate specific errors to the operator.

## 4.3 Digital representation of text

Binary patterns are not just able to represent numerical values. Given a standardized *code*, they may represent other types of data as well such as characters in written languages. The ability to encode non-numerical data in digital form is what allows computers to manipulate and communicate text. In this subsection, some of the ways language characters are digitally encoded are explored.

### 4.3.1 Morse and Baudot codes

In the early days of communication, Morse code was used to represent letters of the alphabet, numerals (0 through 9), and some other characters in the form of "dot" and "dash" signals. Each "dot" or "dash" signal is one *bit* of data, equivalent to "1" and "0". In the International Morse Code, no character requires more than six bits of data, and some (such as the common letters E and T) require only one bit.

A table showing how International Morse code represents all characters of the English alphabet and the numerals 0 through 9 appears here:

[7] It is also possible to "simulate" fractional resolution using an integer number, by having the HMI insert a decimal point in the numerical display. For instance, one could use a 16-bit signed integer having a numerical range of -32,768 to +32,767 to represent motor temperature by programming the HMI to insert a decimal point between the hundreds' and the tens' place. This would give the motor temperature tag a (displayed) numerical range of -327.68 degrees to +327.67 degrees, and a (displayed) resolution of ±0.01 degree. This is basically the concept of a *fixed-point* number, where a fixed decimal point demarcates whole digits (or bits) from fractional digits (or bits).

International Morse Code (English letters and Arabic numerals only)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | •— | J | •——— | S | ••• | 0 | ————— |
| B | —••• | K | —•— | T | — | 1 | •———— |
| C | —•—• | L | •—•• | U | ••— | 2 | ••——— |
| D | —•• | M | —— | V | •••— | 3 | •••—— |
| E | • | N | —• | W | •—— | 4 | ••••— |
| F | ••—• | O | ——— | X | —••— | 5 | ••••• |
| G | ——• | P | •——• | Y | —•—— | 6 | —•••• |
| H | •••• | Q | ——•— | Z | ——•• | 7 | ——••• |
| I | •• | R | •—• | | | 8 | ———•• |
| | | | | | | 9 | ————• |

Figure 4.9: Morse code

The variable bit-length of Morse code, though very efficient[8] in terms of the total number of "dots" and "dashes" required to communicate textual messages, was difficult to automate in the form of teletype machines. In answer to this technological problem, Emile Baudot invented a different code where each and every character was five bits in length. Although this gave only 32 characters, which is not enough to represent the 26-letter English alphabet, plus all ten numerals and punctuation symbols, Baudot successfully addressed this problem by designating two of the characters as "shift" characters: one called "letters" and the other called "figures." The other 30 characters had dual (overloaded) meanings, depending on the last "shift" character transmitted[9].

### 4.3.2 EBCDIC and ASCII

A much more modern attempt at encoding characters useful for text representation was *EBCDIC*, the "Extended Binary Coded Decimal Interchange Code" invented by IBM in 1962 for use with their line of large ("mainframe") computers. In EBCDIC, each character was represented by a one-byte (eight bit) code, giving this code set 256 ($2^8$) unique characters. Not only did this provide enough unique characters to represent all the letters of the English alphabet (lower-case *and* capital letters separately!) and numerals 0 through 9, but it also provided a rich set of *control characters* such as "null," "delete," "carriage return," "linefeed," and others useful for controlling the action of electronic printers and other machines.

A number of EBCDIC codes were unused (or seldom used), though, which made it somewhat

[8] Morse code is an example of a *self-compressing* code, already optimized in terms of minimum bit count. Fixed-field codes such as Baudot and the more modern ASCII tend to waste bandwidth, and may be "compressed" by removing redundant bits.

[9] For example, the Baudot code 11101 meant either "Q" or "1" depending on whether the last shift character was "letters" or "figures," respectively. The code 01010 meant either "R" or "4". The code 00001 meant either "T" or a "5".

inefficient for large data transfers. An attempt to improve this state of affairs was *ASCII*, the "American Standard Code for Information Interchange" first developed in 1963 and then later revised in 1967, both by the American National Standards Institute (ANSI). ASCII is a seven-bit code, one bit shorter per character than EBCDIC, having only 128 unique combinations as opposed to EBCDIC's 256 unique combinations. The compromise made with ASCII versus EBCDIC was a smaller set of control characters.

IBM later created their own "extended" version of ASCII, which was eight bits per character. In this extended code set were included some non-English characters plus special graphic characters, many of which may be placed adjacently on a paper printout or on a computer console display to form larger graphic objects such as lines and boxes.

ASCII is by far the most popular digital code for representing English characters, even today. Nearly every digital transmission of English text in existence employs ASCII as the character encoding[10]. Nearly every text-based computer program's source code is also stored on media using ASCII encoding, where 7-bit codes represent alphanumeric characters comprising the program instructions.

The basic seven-bit ASCII code is shown in this table, with the three most significant bits in different columns and the four least significant bits in different rows. For example, the ASCII representation of the upper-case letter "F" is 1000110, the ASCII representation of the equal sign (=) is 0111101, and the ASCII representation of the lower-case letter "q" is 1110001.

**ASCII code set**

| ↓ LSB / MSB → | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 0000 | NUL | DLE | SP | 0 | @ | P | ‘ | p |
| 0001 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 0010 | STX | DC2 | ” | 2 | B | R | b | r |
| 0011 | ETX | DC3 | # | 3 | C | S | c | s |
| 0100 | EOT | DC4 | $ | 4 | D | T | d | t |
| 0101 | ENQ | NAK | % | 5 | E | U | e | u |
| 0110 | ACK | SYN | & | 6 | F | V | f | v |
| 0111 | BEL | ETB | ’ | 7 | G | W | g | w |
| 1000 | BS | CAN | ( | 8 | H | X | h | x |
| 1001 | HT | EM | ) | 9 | I | Y | i | y |
| 1010 | LF | SUB | * | : | J | Z | j | z |
| 1011 | VT | ESC | + | ; | K | [ | k | { |
| 1100 | FF | FS | , | < | L | \ | l | \| |
| 1101 | CR | GS | - | = | M | ] | m | } |
| 1110 | SO | RS | . | > | N | ˆ | n | ˜ |
| 1111 | SI | US | / | ? | O | _ | o | DEL |

The aforementioned "control characters" occupy the "000" and "001" columns of the table. These characters, while not associated with a printed character on the page, nevertheless play roles the other "visible" characters cannot. The "LF" character ("line feed"), for example, tells the printer to go to the next line on the paper. The "CR" character ("carriage return") tells the printing head

[10]Including the source code for this textbook!

to return to the left-hand side of the paper. The "ESC" character ("escape") does not result in a printed character on paper, nor does it usually command a printing machine to do anything specific, but it is often used within software as a signal to jump up one level in a menu structure, or to switch modes.

### 4.3.3 Unicode

There exist *many* written languages whose characters cannot and are not represented by either EBCDIC or ASCII. In in attempt to remedy this state of affairs, a new standardized code set is being developed called *Unicode*, with sixteen bits per character. This large bit field gives 65,536 possible combinations, which should be enough to represent every unique character in every written language in the entire world. In deference to existing standards, Unicode encapsulates both ASCII and EBCDIC as sub-sets within its defined character set[11].

And no, a table showing all the Unicode characters is not included!

## 4.4 Analog-digital conversion

In order to provide an interface between the internal (digital) world of a computer and the external (analog) world of process measurement and control, there must be some form of *conversion* taking place between these two types of data. Devices that generate digital representations of analog measurements are called *analog-to-digital converters*, or *ADC*s. ADC circuits are found in process transmitters, where some process variable measurement (e.g., pressure, temperature, flow, etc.) must be converted into digital form for a computer to interpret. Devices that generate an analog representation of digital quantities are called *digital-to-analog converters*, or *DAC*s. These are found in the output channels of digital control systems, where the controller must drive some field device (e.g., I/P transducer) with a 4-20 mA analog electronic signal based on a digital calculation (e.g., PID).

### 4.4.1 Converter resolution

Both ADCs and DACs relate analog voltage signals to multi-bit binary (digital) data. Perhaps the most obvious measure of ADC or DAC performance, then, is how many bits of output represent how many volts of analog signal. A simplified diagram of a 12-bit ADC is shown here for illustration:

[11] To illustrate, the first 128 Unicode characters (0000 through 007F hexadecimal) are identical to ASCII's 128 characters (00 through 7F hexadecimal)

Figure 4.10: A simple 12-bit analog-to-digital converter

Since the digital data "width" of this ADC is 12 bits, its digital output ranges from `000000000000` to `111111111111`, constituting a 12-bit binary integer with a range extending from `000` hexadecimal to `FFF` hexadecimal, or 0 decimal to 4095 decimal. This integer number is called the *count* value of the ADC circuit. Although the ADC shown outputs its digital data in *parallel* form (with separate terminals for the 12 individual bits), many modern ADC chips are designed for *serial* data output, where a single terminal generates a sequential series of bits timed to the pulse of a clock signal.

Supposing this 12-bit ADC has an analog input voltage range of 0 to 10 volts, how is any given digital number value related to a voltage value, or vice-versa? The key here is to understand that the 12-bit *resolution* of this ADC means it has $2^{12}$, or 4096 possible count values. The 10 volt DC input range is therefore divided up into $2^{12} - 1$, or 4095, discrete increments:

$$\text{Analog resolution} = \frac{\text{Analog span}}{2^n}$$

Where,

$n$ = Number of binary bits in the output "word"

For the hypothetical 0-10 VDC, 12-bit converter, the analog resolution is 2.4414 millivolts. Thus, for any analog signal between 0 mV and 2.4414 mV, the ADC's output should be zero (binary `000000000000`); for any analog signal between 2.4414 mV and 4.8828 mV, the ADC's output should be one (binary `000000000001`); and so on.

As previously mentioned, the digital value output by an ADC is commonly referred to as a *count*[12]. The word "count" is used in this context as a unit of measurement. For instance, if a 12-bit ADC was subjected to a full-scale input signal of 10 VDC, the expected outcome is a full-scale digital output (binary `111111111111`) of 4095 "counts." Since most ADC circuits are designed to be linear, the mathematical relationship between input voltage and digital output "counts" is a simple proportionality:

[12]The origin of this word has to do with the way many ADC circuits are designed, using binary *counters*. In the *tracking* design of ADC, for instance, an up-down binary counter "tracks" the varying analog input voltage signal. The binary output of this counter is fed to a DAC (digital-to-analog converter) which sends an analog voltage to a comparator circuit, comparing the digital counter's equivalent value to the value of the measured analog input. If one is greater than the other, the up-down counter is instructed to either count up or count down as necessary to equalize the two values. Thus, the up-down counter repeatedly steps up or down as needed to keep pace with the value of that analog voltage, its digital output literally "counting" steps along a fixed scale representing the full analog measurement range of the ADC circuit.

$$\frac{V_{in}}{V_{fullscale}} = \frac{\text{Counts}}{2^n - 1}$$

This formula may be used to generate a partial table of input and output values for the 0-10 VDC, 12-bit ADC:

| $V_{in}$ | Counts (decimal) | Counts (hex) |
|---|---|---|
| 0 V | 0 | `000` |
| 2.46 mV | 1 | `001` |
| 3.85 V | 1576 | `628` |
| 4.59 V | 1879 | `757` |
| 6.11 V | 2502 | `9C6` |
| 9.998 V | 4094 | `FFE` |
| 10 V | 4095 | `FFF` |

In order to calculate a digital count value from a given input voltage, simply divide that voltage value by the full-scale voltage, then multiply by the full-scale count value and round down[13] to the nearest whole number. For any given voltage value input to the ADC, there is (ideally) one corresponding output "count" value. The converse cannot be said, however: for any given output "count" value, there is actually a *range* of possible input voltages (the span of that range being the analog resolution of the ADC, in this case 2.442 mV).

To illustrate, take one of the table entries as an example: an analog input of 6.11 volts should yield a digital output of (precisely) 2502 counts. However, a digital output of 2502 counts could represent any analog input voltage ranging between 6.10989 volts and 6.11233 volts. This uncertainty is inherent to the process of "digitizing" an analog signal: by using a discrete quantity to represent something infinitely variable, some detail is inevitably lost. This uncertainty is referred to as *quantization error*: the (potential) error resulting from "quantizing" (digitizing) an inherently analog quantity into a discrete representation.

Quantization error may be reduced (but never eliminated) by using an ADC with greater resolution. A 14-bit ADC operating over the same 0-10 VDC analog input range would have approximately one-quarter the uncertainty of the 12-bit ADC (0.610 mV instead of 2.442 mV). A 16-bit ADC's uncertainty would only be (approximately) one-sixteenth that of the 12-bit ADC. The number of bits chosen for any particular ADC application is therefore a function of how precise the digitization must be.

Often digital instruments are encountered where the digital "count" scale maps to a live-zero analog range. For example, the Siemens model 353 process controller represents process variable, setpoint, and output ("valve") percentages on a displayed scale of -3.3% to 103.3% with a 12-bit ADC count. For this controller, a digital count of 0 represents an analog signal displayed on the controller's digital readout of -3.3%, and a digital count value of FFF hexadecimal represents a displayed value of 103.3%. The relationship between these two scales may be shown in graphical form, like a number line:

[13] Whether or not the actual ADC will round *down* depends on how it is designed. Some ADCs round down, others "bobble" equally between the two nearest digital values, and others yet "bobble" proportionately between the two nearest values. No matter the calculation of count value is rounded, the rounded value will never be more than 1 count off from the real ADC's value.

Figure 4.11: Scaling of percent analog values into 12-bit hexadecimal

Converting a digital count value to its respective analog percentage (or visa-versa) follows the same procedure used to convert values between any two representative scales where one of the scales has a "live zero:" take the given value and convert to a percentage of span (subtracting any live zero before dividing by the span), then calculate the other value based on that percentage of span (adding any live zero after multiplying by the span).

For example, suppose one of these Siemens controllers displayed a value of 26.7% on its front panel. The digital (count) value that corresponds to this is:

Figure 4.12: Scaling of an analog percentage into hexadecimal

Here, the displayed value of 26.7% is actually 30% away from the scale's starting point of -3.3%. Compared to the scale's span of 106.6%, this equates to a true percentage value of:

$$\frac{26.7 - (-3.3)}{106.6} = \frac{30}{106.6} = 0.2814 = 28.14\%$$

Since a 12-bit binary count goes from 0 to 4095, and the displayed reading of 26.7% corresponds to 28.14% of the ADC's full-scale count (4095), the count value corresponding to the displayed reading of 26.7% will be 1152, or 480 hexadecimal (480h).

Similarly, if the range of this 12-bit ADC is known in actual process engineering units, a conversion may be made between ADC counts and the process value by the same method. Suppose one of these same controllers to display the temperature of a furnace, where the lower- and upper-range values were 900 deg F and 1850 deg F, respectively. It is important to realize that these lower- and upper-range values represent 0% and 100% of scale, and that the ADC is actually configured to cover a range from -3.3% to 103.3%. In other words, the ADC range "spills over" the specified range of 0 to 100% (900 to 1850 $^o$F) by 3.3% on both ends, equating to 31.35 $^o$F above and below

the specified range limits. Thus, the 12-bit ADC's full-count range of 0 to 4095 (000 to FFF hex) actually covers a temperature range of 868.65 $^o$F to 1881.35 $^o$F.

The ADC count can be related to this temperature scale using the same "number line" format as the previous example:

Figure 4.13: Scaling of 12-bit hexadecimal into temperature analog values

Suppose the ADC count inside this Siemens controller for a certain furnace temperature was A59 hexadecimal (A59h), equal to 2649 in decimal form:

Figure 4.14: Scaling of an hexadecimal into temperature

To convert this count value into a temperature, first determine its percentage of span:

$$\frac{2649}{4095} = 0.6469 = 64.69\%$$

Next, calculate the how much of the temperature scale's span this equates to:

$$(0.6469)(1881.35 - 868.65) = 655.10$$

Thus, this temperature is 655.10 degrees hotter than the bottom of the scale (at 868.65 degrees F). Adding the live zero value of 868.65 degrees F, the furnace temperature is 1523.75 degrees F.

### 4.4.2 Converter sampling rate

The next major performance metric for analog signal digitization is how often the analog signal gets converted into digital form. Each time an ADC circuit "samples" its analog input signal, the resulting

digital number is fixed until the next sample. This is analogous to monitoring a continuously moving process by taking a series of still-photographs. Any changes happening to the analog signal between sampling events are not detected by the converter, and therefore are not represented in the digital data coming from the converter.

It stands to reason, then, that the sampling rate of any ADC must be at least as often as significant changes are expected to take place in the analog measurement. According to the *Nyquist-Shannon Sampling Theorem*, the absolute minimum sample frequency necessary to capture an analog waveform is greater than twice the waveform's highest frequency content. If the sample rate is just slightly over twice the highest frequency content, the original waveform can be reconstructed provided an almost infinitely long stream of sampled data is available. As the sampling frequency ratio goes up, the length of the data stream required to recreate the original waveform shrinks. A more realistic approach is to have the ADC sample the waveform *ten times* or more per cycle.

In general electronics work, for example with the design of electronic test equipment such as digital multimeters (DMMs) and digital storage oscilloscopes (DSOs), sampling rates must be rather fast. Modern digital oscilloscopes may have sampling rates in the *billions* of samples per second, to allow for the successful digitization of radio-frequency analog signals.

Industrial process measurements are far more forgiving than measurements commonly performed on an electronic technician's workbench, thankfully. The temperature of a large furnace may be adequately sampled at a rate of only once per minute, if need be. Even "fast" feedback processes such as liquid flow and pressure control may be controlled with reasonable stability by digital systems sampling just a few times per second.

A sampling rate that is too slow (infrequent) may detrimentally affect an instrumentation in more than one way. First, the time between samples is *dead time* to the system: time during which the digital system will be completely unresponsive to any changes in process measurement. Excessive dead time in an alarm system means an unnecessary time delay between the alarm event and the alarm signal. Excessive dead time in a feedback control loop leads to oscillation and instability. Another detrimental effect of low sampling rate is something called *aliasing*: a condition where the digital system "thinks" the frequency of an analog signal is far lower than it actually is.

A dramatic example of aliasing is shown in the following illustration, where a sinusoidal signal (colored blue) is sampled at intervals slightly slower than once per cycle (samples marked by red dots). The result (the red, dashed curve) is what appears to be a much lower-frequency signal as seen by the digital system, which only "sees" the values represented by the red dots:

Figure 4.15: Aliasing of a signal sampled too slowly

The troubling nature of aliasing is that it causes the ADC to report a *completely incorrect*, yet *completely plausible* signal. One simple way to avoid aliasing in an ADC circuit is to place an analog low-pass filter circuit before the ADC's input, preventing any analog signals with frequencies beyond the Nyquist limit to pass through to the ADC. Such a "front-end" circuit is called an *anti-aliasing*

*filter*:

Figure 4.16: Use of an anti-aliasing filter

Aliasing may still occur within digital systems, though, if one portion of a system "samples" the digital output of another portion at too slow of a rate. An example of this might be the rate at which a digital control system (such as a DCS) polls a process variable value collected by a digital sensor network (such as a network of radio-linked process transmitters, or digital fieldbus transmitters). If the DCS polling rate is sufficiently slow compared to the frequency of the signal reported by the digital transmitters, aliasing may result. The best guard against such potential troubles is to synchronize the sampling rates throughout the system, or (alternatively) ensure data sources always output values at a significantly slower rate than any functions reading them. A practical (minimum) sample rate ratio is 10:1 to ensure good signal integrity.

## 4.5 Digital data communication theory

One of the great benefits of digital technology is the ability to *communicate* vast amounts of information over networks. This very textbook was transmitted in digital form over the electronic network called the *Internet*, a feat nearly impossible with any sort of analog electronic technology. The main benefit of digital data communication in industrial control is simple: no longer must a single pair of wires be run to each and every instrument in a facility as is necessary with analog (4-20 mA) signaling. With digital signaling, a single pair of wires or coaxial cable is able to convey a theoretically unlimited number of data points.

This benefit comes at a price, though: in order to communicate multiple variables (data points) over a single channel (wire pair), those signals must be transmitted and received one at a time. This means a digital communications system will necessarily exhibit some degree of *latency, or time delay* in acquiring, transmitting, receiving, and interpreting a signal. Analog systems, by contrast,

are virtually instantaneous[14]. Thus, the contrast between analog and digital communication pitting channel capacity against speed is shown:

| **Analog** | **Digital** |
|---|---|
| Only one signal per channel | Many signals per channel possible |
| Instantaneous | Time-delayed |

With modern electronic technology it is possible to build digital communication systems that are so fast, the time delays are negligible for most industrial processes, which renders the second comparison (instantaneous versus time-delayed) moot. If time is no longer an issue, the advantage that digital communication has over analog in terms of channel usage makes it the superior choice[15].

Another important advantage of digital data communication for industrial processes is increased *noise immunity*. Analog data is *continuous* by nature: a signal of 11.035 milliamps has a different meaning than a signal of 11.036 milliamps, because any measurable increment in signal represents a corresponding increment in the physical variable represented by that signal. A voltage value in a 0-5 volt digital signaling system of 0.03 volts, however, means *the exact same thing* as a voltage value of 0.04 volts: either one is still interpreted as a "0" or "low" state. *Any* amount of electrical noise imposed on an analog signal corrupts that signal to some degree. A digital signal, however, may tolerate a substantial amount of electrical noise with no corruption whatsoever.

Not surprisingly, though, the noise immunity enjoyed by digital signals comes with a price: a sacrifice in *resolution*. Analog signals are able to represent the smallest imaginable changes because they are continuously variable. Digital signals are limited in resolution by the number of bits in each data "word." This is another contrast between analog and digital data representation:

| **Analog** | **Digital** |
|---|---|
| Corrupted by any amount of noise | Immune to certain (limited) amounts of noise |
| Unlimited resolution | Limited resolution |

With modern digital electronic technology, however, the "limited resolution" problem is almost nonexistent. 16-bit converter chipsets are commonly available today for input/output (I/O) modules on digital systems, providing a resolution of $2^{16}$ (65,536) counts, or $\pm$ 0.00153%, which is good enough for the vast majority of industrial measurement and control applications.

This section will focus on *serial* data transmission, as opposed to *parallel*. In order to transmit digital data in parallel form, the number of wires scales directly with the number of bits in each data "word." For example, if a 16-bit ADC chip were to communicate its data to some other digital device using a parallel network, it would require a cable with 16 wires (plus a common "ground" wire) *at minimum*. Since this approach undercuts the "fewer wires" advantage that digital communications theoretically enjoys over analog communication, parallel data transmission is rarely seen in industry

[14]To be fair, there is such a thing as a time-multiplexed analog system for industrial data communication (One such system, used to measure voltages on electrolytic "pots" in the aluminum industry, communicates the voltages across hundreds of individual pots to a central control computer).

[15]There is, of course, the issue of *reliability*. Communicating thousands of process data points over a single cable may very well represent a dramatic cost savings in terms of wire, junction boxes, and electrical conduit. However, it also means all those thousands of data points will be lost if that one cable becomes severed! Even with digital technology, there is still reason sometimes to under-utilize the bandwidth of signal cables.

except for within the internal construction of a digital device (e.g., a parallel data bus inside a personal computer, or inside a PLC or DCS rack).

In serial communications systems, digital data is sent over a wire pair (or fiber optic cable, or radio channel) *one bit at a time*. A 16-bit digital "word" (two *bytes* in length) then will require a succession of 16 bits transmitted one after the other in time. How each bit is encoded an electrical signal, how those bits are arranged in time to group them into meaningful "words," and how multiple devices share access to a common communications channel, is the next subject of exploration: the technical details of serial data communication.

### 4.5.1 Serial communication principles

The task of encoding real-life data as a series of on-and-off electrical signals, and then sending those signals long distances over cables (or optical fibers, or radio waves) requires mutually-agreed *standards* for the encoding, the "packaging" of those bits, the speed at which the bits are sent, methods for multiple devices to use a common channel, and a host of other concerns. This subsection will delineate the major areas of concern where standards must be established before digital devices may communicate between each other serially.

An early form of digital communication was *Morse Code*, used to communicate alpha-numerical information as a series of "dots" and "dashes" over telegraph[16] systems. Each letter in the alphabet, and each numerical digit (0 through 9) was represented in Morse Code by a specific series of "dot" and "dash" symbols, a "dot" being a short pulse and a "dash" being a longer pulse. A similar code system called the *Continental Code* was used for early radio ("radiotelegraph") communications.

As primitive as these codes were, they encapsulated many of the basic principles found in modern digital serial communication systems. First, a system of codes needed to exist to represent the letters and numbers required to spell messages in English, which the general American public conversed in. Next, there needed to be some way to represent these characters individually so that one could distinguish the end of one character and the beginning of the next character.

For example, consider the Continental Code encoding for the word `NOWHERE`. By placing an extra space (a pause in time) between characters, it is easy to represent individual characters in the message:

Figure 4.17: Example of Continental Code

If this space between characters were not present, it would be impossible to determine the message with certainty. By removing the spaces, it is easy to see that multiple non-sensical interpretations are possible for the same string of "dots" and "dashes:"

[16] Readers of this book are not expected to have firsthand knowledge of what a "telegraph" is, but some may have never heard of one until this point. Basically, a telegraph was a primitive electrical communication system stretching between cities using a keyswitch at the transmitting end to transmit on-and-off pulses and a "sounder" to make those pulses audible on the receiving end. Trained human operators worked these systems, one at the transmitting end (encoding English-written messages into a series of pulses) and one at the receiving end (translating those pulses into English letters).

Figure 4.18: Examples of different interpretations of code

For that matter, it is even possible to confuse the meaning of the text string "`NOWHERE`" when the individual characters are properly interpreted. Does the string of characters say "nowhere," or does it say "now here"?

This simple example illustrates the need for *delimiting* in serial data communication. Some means must be employed to distinguish individual groups of bits (generally called *frames* or *packets*) from one another, lest their meanings be lost. In the days when human operators sent and interpreted Morse and Continental code messages, the standard delimiter was an extra time delay (pause) between characters, and between words. This is not much different from the use of whitespace to delineate words, sentences, and paragraphs typed on a page. Sentenceswouldcertainlybeconfusingtoreadifnotforspaces!

In later years, when *teletype* machines were designed to replaced skilled Morse operators, the concept of frame delineation had to be addressed more rigorously. These machines consisted of a typewriter-style keyboard which marked either paper strips or pages with dots corresponding to a 5-bit code called the *Baudot code*. The paper strip or sheets were then read electrically and converted

into a serial stream of on-and-off pulses which were then transmitted along standard telegraph circuit lines. A matching teletype machine at the receiving end would then convert the signal stream into printed characters (a telegram). Not only could unskilled operators use teletype machines, but the data rate far exceeded what the best human Morse operators could achieve[17]. However, these machines required special "start" and "stop" signals to synchronize the communication of each character, not being able to reliably interpret pauses like human operators could.

Interestingly, modern asynchronous[18] serial data communication relies on the same concept of "start" and "stop" bits to synchronize the transmission of data packets. Each new packet of serial data is preceded by some form of "start" signal, then the packet is sent, and followed up by some sort of "stop" signal. The receiving device(s) synchronize to the transmitter when the "start" signal is detected, and non-precision clocks keep the transmitting and receiving devices in step with each other over the short time duration of the data packet. So long as the transmitting and receiving clocks are close enough to the same frequency, and the data packet is short enough in its number of bits, the synchronization will be good enough for each and every bit of the message to be properly interpreted at the receiving end.

### 4.5.2 Physical encoding of bits

Telegraph systems were Boolean in nature: representing "dots" and "dashes" by one electrical state of the telegraph line, and pauses by another. When manually-actuated keyswitches were abandoned in favor of teletype machines, and Morse code abandoned in favor of the *Baudot* (5-bit) code for representing alphanumeric characters, the electrical nature of the telegraph (at least initially[19]) remained the same. The line would either be energized or not, corresponding to *marks* or *spaces* made on the teletype paper.

Many modern digital communication standards represent binary "1" and "0" values in exactly this way: a "1" is represented by a "mark" state and a "0" is represented by a "space" state. "Marks" and "spaces" in turn correspond to different voltage levels between the conductors of the network circuit. For example, the very common TIA-232 serial communications standard (once the most popular way of connecting peripheral devices to personal computers, formerly called RS-232[20]) defines a "mark" (1) state as -3 volts between the data wire and ground, and a "space" (0) state as +3 volts between the data wire and ground. This is referred to as *Non-Return-to-Zero* or `NRZ` encoding:

[17] A test message sent in 1924 between two teletype machines achieved a speed of 1920 characters per minute (32 characters per second), sending the sentence fragments "THE WESTERN ELECTRIC COMPANY", "FRESHEST EGGS AT BOTTOM MARKET PRICES", and "SHE IS HIS SISTER".

[18] "Asynchronous" refers to the transmitting and receiving devices not having to be in perfect synchronization in order for data transfer to occur. Most industrial data communications standards are asynchronous rather than synchronous. In synchronous serial networks, a common "clock" signal maintains transmitting and receiving devices in a constant state of synchronization, so that data packets do not have to be preceded by "start" bits or followed by "stop" bits. Synchronous data communication networks are therefore more efficient (not having to include "extra" bits in the data stream) but also more complex. Most long-distance, heavy traffic digital networks (such as the "backbone" networks used for the Internet) are synchronous for this reason.

[19] Later versions of teletype systems employed audio tones instead of discrete electrical pulses so that many different channels of communication could be funneled along one telegraph line, each channel having its own unique audio tone frequency which could be filtered from other channels' tones.

[20] "RS" refers to recommended standard, the standard is in approved status; "TIA" refers to the Telecommunications Industry Association, the standard's sponsoring organization.

Figure 4.19: Non-Return-to-Zero (NRZ) encoding of bits

An easy way to remember the difference between a "mark" and a "space" in this scheme is to recall the operation of old telegraph printing units, specifically how they created marks and spaces on moving paper strip. When the printing unit was energized (i.e., the transmitting key was pressed, sending current through the solenoid coil of the printer, corresponding to a "1" state), the iron armature in the printer would be pulled *down* to draw a mark on the paper strip. When de-energized (transmitting key released, stopping current in the telegraph line, corresponding to a "0" state), the printer's iron bar would spring-return *up* from the paper to leave an unmarked space.

This is not the only way to represent binary bits, though. An alternative method is to use an oscillating (square-wave) signal, counting *up* and *down* transitions (pulse edges) at specific times to represent 1 and 0 states. This is called *Manchester encoding*, and it is used in the 10 Mbps (10 million bits per second) version of *Ethernet* and in both the *FOUNDATION Fieldbus* "H1" and *Profibus* "PA" instrumentation network standards:

Figure 4.20: Manchester encoding of bits

Note how each binary bit (0 or 1) is represented by the *direction* of the voltage signal's transition. A low-to-high transition represents a "1" state while a high-to-low transition represents a "0" state. Extra "reversal" transitions appear in the pulse stream only to set up the voltage level as needed for the next bit-representing transitions. The representation of bits by transitions rather than by static voltage levels guarantees the receiving device can naturally detect the clock frequency of the transmitted signal[21]. Manchester data is therefore referred to as *self-clocking*.

Interpreting a Manchester waveform is easier than it first appears. The key is identifying which transitions represent "clocked" bits and which transitions represent "reversals" prior to bits. By identifying the widest periods of the waveform, the transitions in these periods must represent real bits because there are no reversals. Another way to say this is the greatest time period found between successive transitions in a Manchester waveform *is* the clock period. Once this clock period is identified, to distinguish clocked bits from reversals simply step along the waveform at that same period width.

Yet another method for encoding binary 1 and 0 states is to use sine waves of different frequencies ("tone bursts"). This is referred to as *Frequency Shift Keying*, or *FSK*, and it is the method of encoding used in the HART "smart" instrument communications standard.

[21] This is most definitely *not* the case with NRZ encoding. To see the difference for yourself, imagine a continuous string of either "0" or "1" bits transmitted in NRZ encoding: it would be nothing but a straight-line DC signal. In Manchester encoding, it is *impossible* to have a straight-line DC signal for an indefinite length of time. Manchester signals *must* oscillate at a minimum frequency equal to the clock speed, thereby guaranteeing all receiving devices the ability to detect that clock speed and thereby synchronize themselves with it.

Figure 4.21: Frequency shift keying (FSK) encoding of bits

In the HART standard – based on the Bell 202 standard used in telephone data exchange – a tone of 2200 Hz represents a "0" bit (space), while a tone of 1200 Hz represents a "1" bit (mark). This standard, along with a series of previous and subsequent modulation standards, was invented as a way to exchange digital data over telephone networks, which were built to communicate audio-frequency AC signals and thus could not reliably communicate the square-wave signals associated with direct digital data. By assigning digital values to different audio frequencies, serial data could be communicated over telephone channels as a series of sine-wave tones.

The same principle of FSK allows HART-compatible instruments to communicate digital data over cables that simultaneously carry DC current (4 to 20 mA) signals representing process and manipulated variables. Since each bit of FSK-encoded data consists of complete AC cycles (one full positive swing for every full negative swing), the time-averaged value of the DC signal carried by the same cable is identical to the value of that signal with no AC tones superimposed.

Other methods exist as well for encoding digital data along network cables, but these three are the most popular in industrial network standards.

### 4.5.3 Communication speed

In order to successfully communicate digital data along a network, there must not only be a standard agreed upon between transmitting and receiving devices for encoding bits (NRZ, Manchester, FSK, etc.), but there must also be a standard in place for the *speed* at which those bits will be sent. This is especially true for NRZ and FSK encoding, where the "clock" speed is not explicitly present in the signal[22].

For example, consider the confusion that could arise interpreting a NRZ signal if the transmitting device sends data at half the speed assumed by the receiving device:

[22] This is one of the advantages of Manchester encoding: it is a "self-clocking" signal.

Figure 4.22: Transmitter/receiver speed mismatch

Thus, one of the essential parameters in a serial data communication system is the *bit rate*, measured in *bits per second* (bps). Some communications standards have fixed bit rates, such as FOUNDATION Fieldbus H1 and Profibus PA, both standardized at exactly 31.25 kbps. Some, such as Ethernet, have a few pre-defined speeds (10 Mbps, 100 Mbps, 1 Gbps) defined by the specific transmitting and receiving hardware used. Others, such as TIA-232 may be arbitrarily set by the user at speeds ranging from 300 bps to over 115 kbps.

An older term sometimes used synonymously with bit rate is *baud rate*, however "bits per second" and "baud" are actually different things. "Baud" refers to the number of voltage (or current) alternations per second of time, whereas "bits per second" refers to the actual number of binary data bits communicated per second of time. Baud is useful when determining whether or not the bandwidth (the maximum frequency capacity) of a communications channel is sufficient for a certain communications purpose. In systems using NRZ encoding, the baud rate is equivalent[23] to the bit rate: for a string of alternating bits (`010101010101`), there will be exactly one voltage transition for each bit. In systems using Manchester encoding, the worst-case[24] baud rate will be exactly *twice* the bit rate, with two transitions (one up, one down) per bit. In some clever encoding schemes, it is possible to encode multiple bits per signal transition, such that the bit rate will actually be greater than the baud rate.

[23]This is likely why "bit rate" and "baud rate" became intermingled in digital networking parlance: the earliest serial data networks requiring speed configuration were NRZ in nature, where "bps" and "baud" are one and the same.

[24]For Manchester encoding, "worst-case" is a sequence of identical bit states, such as `111111111111`, where the signal must make an extra (down) transition in order to be "ready" for each meaningful (up) transition representing the next "1" state.

### 4.5.4 Data frames

As mentioned earlier in this section, serial data is usually communicated *asynchronously* in industrial networks. This means the transmitting and receiving hardware need not be in perfect synchronization to reliably send and receive digital data. In order for this to work, data must be sent in "frames" or "packets" of fixed (maximum) length, each frame preceded by a special "start" signal and concluded with a special "stop" signal. As soon as the transmitting device issues the "start" signal, the receiving device synchronizes to that start time, and runs at the pre-determined clock speed to gather the successive bits of the message until the "stop" signal is received. So long as the internal clock circuits of the transmitting and receiving devices are running at *approximately* the same speed, the devices will be synchronized closely enough to exchange a short message without any bits being lost or corrupted. There is such a thing as a *synchronous* digital network, where all transmitting and receiving devices are locked into a common clock signal so they cannot stray out of step with each other. The obvious advantage of synchronous communication is that no time need be wasted on "start" and "stop" bits, since data transfer may proceed continuously rather than in packets. However, synchronous communication systems tend to be more complex due to the need to keep all devices in perfect synchronization, and thus synchronous systems are typically used for long-distance, high-traffic digital networks such as those use for Internet "backbones" and not for short-distance industrial networks.

Like bit rate, the particular scheme of start and stop bits must also be agreed upon in order for two serial devices to communicate with each other. In some networks, this scheme is fixed and cannot be altered by the user. Ethernet is an example of this, where a sequence of 64 bits (an alternating string of "1" and "0" bits ending with a "1, 1"; this is the "preamble" and "start frame delimiter" or "SFD" bit groups) is used to mark the start of a frame and another group of bits specifies the length of the frame (letting the receiver know ahead of time when the frame will end). A graphic description of the IEEE 802.3 standard for Ethernet data frames is shown here, illustrating the lengths and functions of the bits comprising an Ethernet frame:

Figure 4.23: Ethernet frame makeup per IEEE 802.3

Other serial networks offer choices for the user to select regarding these parameters. One such example is TIA-232 (formerly RS-232), where the user may specify not only the bit rate, but also how many bits will be used to mark the end of the data frame. It is imperative in such systems that *all* transmitting and receiving devices within a given network be configured exactly the same, so that they will all "agree" on how to send and receive data. A screenshot from a UNIX-based serial

communication terminal program (called *minicom*[25]) shows these options:

Figure 4.24: TIA-232 serial communication parameter options

This particular screenshot shows the data rate options (extending from 300 bps all the way up to 230,400 bps!), the number of data bits (from 5 to 8), and the number of stop bits (1 or 2), all configurable by the user. Of course, if this program were being used for communication of data between two personal computers, *both* of those computers would need these parameters set identically in order for the communication to take place. Otherwise, the two computers would not be in agreement on speed, number of data bits, and stop bits; their respective data frames simply would not match.

To give an example of an TIA-232 data frame might look like as a series of voltage states, consider this waveform communicating a string of eight bits (`01011001`), using NRZ encoding. Here, a single "start" marks the beginning of the data frame, while two successive "stop" bits end it. Also note how the bit sequence is transmitted "backwards," with the least-significant bit (LSB) sent first and the most-significant bit (MSB) sent last[26]:

[25] An equivalent program for Microsoft Windows is *Hyperterminal*. A legacy application, available for both Microsoft Windows and UNIX operating systems, is the serial communications program called *kermit*.

[26] This is standard in TIA-232 communications.

Figure 4.25: An example of an TIA-232 data word

Interestingly, the "mark" state (corresponding to a binary bit value of "1") is the default state of the communications channel when no data is being passed. The "start" bit is actually a space (0). This is the standard encoding scheme for TIA-232, TIA-485, and some other NRZ serial communication standards.

One of the options in the "minicom" terminal program screenshot was something called *parity*. This is a simple form of error-checking used in many serial communication standards. The basic principle is quite simple: an extra bit is added at the end of the data frame (just before the "stop" bits) to force the total number of "1" states to be either odd or even. For example, in the data stream just shown (`10011010`), there is an *even* number of "1" bits. If the computer sending this eight-bit data group were configured for "odd" parity, it would append an additional "1" to the end of that frame to make the total number of "1" bits odd rather than even. If the next data group were `11001110` instead (already having an odd number of "1" bits), the transmitting computer would have to attach a "0" parity bit on to the data frame in order to maintain an odd count of "1" bits.

The way this works to check for errors is for the receiving computer to count up all the "1" bits in each data frame (including the parity bit), and check to see that the total number is still odd (if the receiving computer is configured for odd parity just as the transmitting computer, which the two should *always* be in agreement). If any one bit somehow gets corrupted during transmission, the received frame will not have the correct parity, and the receiving computer will "know" something has gone wrong. Parity does not suggest *which* bit got corrupted, but it will indicate if there was a single-bit[27] corruption of data, which is better than no form of error-checking at all.

The following example shows how parity-checking would work to detect a transmission error in a 7-bit data word. Suppose a digital device asynchronously transmits the character "T" using ASCII encoding ("T" = 1010100), with one start bit, one stop bit, and "odd" parity. Since the "start" bit is customarily a 0 state (space), the data transmitted in reverse order (LSB first, MSB last), the parity

[27] It should take only a moment or two or reflection to realize that such a parity check cannot detect an *even* number of corruptions, since flipping the states of any *two* or any *four* or any *six* (or even all eight!) bits will not alter the evenness/oddness of the bit count. So, parity is admittedly an imperfect error-detection scheme. However, it is certainly better than nothing!

bit transmitted after the data's MSB, and the "stop" bit represented by a 1 state (mark), the entire frame will be the following sequence of bits: `0001010101`. Viewed on an oscilloscope display where a negative voltage represents a "mark" and a positive voltage represents a "space," the transmitted data frame will look like this:

Figure 4.26: An example of odd parity on a serial communication link

Note how the parity bit in this particular frame is a 0, because the parity type is set for "odd," and the 7-bit data word already has an odd number of 1 bits.

Now suppose this transmitted frame encounters a significant amount of electrical noise as it travels to the receiving device. If the frame reaches the receiver as shown in the next illustration, the receiver will interpret the message incorrectly:

Figure 4.27: An example of a bit error during serial transmission

One of the bits has been corrupted by noise, such that the frame is received as `0001000101` instead of `0001010101` as it was transmitted. When the receiver goes to count the number of 1 bits in the message (data plus parity bit, disregarding start and stop bits), it will count an even number of 1's instead of an odd number of 1's. Since the receiver is also set for "odd" parity to match the transmitter, it expects an odd number of 1's in the received message. Thus, it "knows" there is a problem somewhere in this transmission, because the received parity is not odd as it should be.

Parity-checking does not tell *which* bit is corrupted, but it does indicate that *something* has gone wrong in the transmission. If the receiving device is programmed to take action on receipt of a non-matching parity, it may reply with a request for the transmitting device to re-send the data as many times as necessary until the parity is correct.

Looking at the "minicom" terminal screenshot in Figure 4.24 to analyze the parity options, it is easy to see there are several options available. The five options for parity in this program include *None*, *Even*, *Odd*, *Mark*, and *Space*. "No" parity is self-explanatory: the transmitting computer does not attach an extra bit for parity at all, and the receiving computer does not bother to check for it. Since the inclusion of a parity bit does add to the bulk of a data frame, it has the unfortunate effect of slowing down communications (more bit "traffic" occupying the channel than would otherwise need to be), thus the option to waive parity altogether for a more compact (faster) data frame. "Even" and "Odd" parity options work as previously described, with the transmitting computer adding a parity bit to each frame to bring the total "1" bit count either to an even number or to an odd number (depending on the user's configuration), and the receiving computer checks for the same. "Mark" and "Space" are really of limited usefulness. In either of these two options, a parity bit is added, but the transmitting computer does not bother to calculate the evenness or oddness of the data bits, rather simply making the parity bit always equal to 1 ("mark") or 0 ("space") as chosen by the user. The receiving computer checks to see that the parity bit is always that value. These two options are of limited usefulness because the parity bit fails to reflect the status of the data being transmitted. The only corruption the receiving computer can detect, therefore, is a corruption of the parity bit itself!

One will often find the communications parameters of a serial network such as this displayed

in "shorthand" notation as seen at the top of the "minicom" terminal display: `38400 8N1`. In this case, the terminal program is configured for a bit rate of 38400 bits per second, with a data field 8 bits long, no parity bit, and 1 stop bit. A computer configured for a bit rate of 9600 bps, with a 7-bit data field, odd parity, and 2 stop bits would be represented as `9600 7O2`.

Parity bits are not the only way to detect error, though. Some communication standards employ more sophisticated means. In the Ethernet (IEEE 802.3) standard, for example, each data frame is concluded with a *frame check sequence*, which is a collection of bits mathematically calculated by the transmitting device based on the content of the data. The algorithm is called a *cyclic redundancy check*, or *CRC*, and is similar to the concept of "checksum" used by computers to check the integrity of data stored in hard disks and other "permanent" media. Like a parity algorithm, the CRC algorithm runs through a mathematical process whereby all the bits in the data field are counted, and a number is generated to reflect the statuses of those bits. The receiving computer takes the received data field and performs the exact same mathematical algorithm, generating its own CRC value. If any of the data's bits become corrupted during transmission, the two CRC values will not match, and the receiving computer will know *something* has gone wrong.

Like parity, the CRC algorithm is not perfect. There exists a chance that just the right combination of errors may occur in transmission causing the CRC values at both ends to match even though the data is not identical, but this is highly unlikely (calculated to be one chance in $10^{14}$). It is certainly better than having no error detection ability at all.

If the communications software in the receiving computer is configured to take action on a detection of error, it may return a "request for re-transmission" to the transmitting computer, so the corrupted message may be re-sent. This is analogous to a human being hearing a garbled transmission in a telephone conversation, and subsequently requesting the other person repeat what they just said.

Another option often found in serial data communications settings is something called *flow control*, not to be confused with the actual control of fluid through a pipe. In the context of digital communications, "flow control" refers to the ability of a receiving device to request a reduction in speed or even a complete cessation of data transmission in case of congestion. An example common to early personal computers was that of a mechanical printer receiving print data from a computer. While the computer may be able to transmit data to be printed at a very rapid pace, the printer is limited by the speed of its mechanism. In order to make the printing process go more smoothly, printers are equipped with *buffer memory* to store portions of the print job received from the transmitting computer that have not had time to print yet. However, these buffers were of finite size, and could become overwhelmed on large print jobs. So, if and when a printer detected its buffer near full capacity, it could issue a command to the computer to freeze data transmission until the printer's buffer has had some time to empty.

Flow control in serial networks may take place in either *hardware* mode or *software* mode. "Hardware" mode refers to the existence of additional connector pins and cable conductors specifically designated for such "halt" signals. "Software" mode refers to data codes communicated over the regular network channel telling the transmitting device to halt and resume. Software flow control is sometimes referred to as XON/XOFF in honor of the names given to these codes[28].

[28] The "XOFF" code told the transmitting device to halt its serial data stream to give the receiving device a chance to "catch up." In data terminal applications, the XOFF command could be issued by a user by pressing the key combination <Ctrl><S>. This would "freeze" the stream of text data sent to the terminal by the host computer. The key combination <Ctrl><Q> sent the "XON" code, enabling the host computer to resume data transmission to the terminal.

The following screen shot shows options for flow control in the "minicom" terminal program:

Figure 4.28: Flow control options for a serial communications link

Here, "hardware" flow control is enabled and "software" flow control is disabled.

### 4.5.5 Channel arbitration

When two or more communication devices exchange data, the directions of their communication may be classified into one of two categories: *simplex* or *duplex*. A "simplex" network is one-way communication only. A sensor outputting digital data to a remotely-located indicator over a digital network would be an example of simplex communication, where the flow of information goes from sensor to indicator, and never the other direction. A public-address (PA) system is an analog example of a simplex communication system, since audio information only goes in one direction (from the person with the microphone to the audience).

"Duplex" communication refers to two-way data exchange. Voice telephony is an analog example of two-way (duplex) communication, where either person at the end of the connection can hear the other person talking. Duplex communication may be further subdivided into *half-duplex* and *full-duplex*, referring to whether or not the two-way communication may be simultaneous. In a "full-duplex" system, both devices may transmit data to each other simultaneously because they have separate channels (separate wires, or optical fibers, or radio frequencies) for their respective transmissions. In a "half-duplex" system, only one device may transmit at any time because the devices must share a common channel. A telephone system is an example of a full-duplex system, although it may be rather difficult for the people to understand each other when they are speaking over one another. A push-to-talk radio system ("walkie-talkie") is an example of a half-duplex system, where each person must take turns talking.

Most industrial data networks are half-duplex, if only for the reason that most networks consist of more than two devices on a network segment. When more than two devices share a network, there are not enough data channels to allow *all* of the devices to simultaneously transmit and listen

to each other. Thus, virtually any network supporting more than two devices will be half-duplex at best, and may even be limited to simplex operation in some cases.

In half-duplex systems, there must be some way for the respective devices to "know" when they are allowed to transmit. If multiple devices sharing one communications channel attempt to transmit simultaneously, their messages will "collide" in such a way that no device on the network will be able to interpret either message. The problem is analogous to two people simultaneously pressing the "talk" buttons on their two-way radio units: neither of the talking people can hear each other, and anyone else on the same channel hears the garbled amalgam of those two peoples' superimposed transmissions. In order to avoid this scenario in a half-duplex network, there must be some strategy to coordinate transmissions so only one device may "talk" at any given time. The problem of deciding "who" gets to "talk" at any given time is generally known as *channel arbitration*. Several strategies for addressing this problem have been developed in the data communications field, a few of which will be described in this subsection.

#### Master-slave

The first method works on the principle of having only one device on the network (the "master") with permission to arbitrarily transmit data. All other devices on the network are "slaves," which may only respond in direct answer to a query from the master. If the network happens to be simplex in nature, slave devices don't even have the ability to transmit data – all they can do is "listen" and receive data from the one master device.

For example, in a half-duplex master-slave network, if one slave device has data that needs to be sent to another slave device, the first slave device must wait until it is prompted ("polled") by the master device before it is allowed to transmit that data to the network. Once the data is transmitted, any and all slave devices may receive that transmission, since they all "listen" to the same communications channel.

An example of an industrial network using master-slave channel arbitration is HART *multidrop*, where multiple HART field instruments are parallel-connected on the same wire pair, and one device (usually a dedicated computer) serves as the master node, polling the field instruments one at a time for their data.

Another example of a master-slave industrial network is a *Modbus* network connecting a programmable logic controller (PLC) to multiple variable-frequency motor drives (VFDs). The master device (the PLC) initiates all communications, with the slave devices (the motor drives) at most replying to the PLC master (and in many cases not replying at all, but merely receiving data from the PLC in simplex mode).

Master-slave arbitration is simple and efficient, but suffers from one glaring weakness: if the master device happens to fail, all communication on the network halts. This means the ability of *any* device on the network to transmit information utterly depends on the proper function of *one* device, representing a high level of dependence on that one (master) device's function.

Some master-slave networks address this problem by pre-assigning special "back-up" status to one or more slave devices. In the event that the master device fails and stops transmitting for a certain amount of time, the back-up device becomes "deputized" to act as the new master, taking over the role of the old master device by ensuring all slave devices are polled on schedule.

**Token-passing**

Another method of arbitrating which device gets to transmit on a channel in a half-duplex network is the *token-passing* method. Here, a special data message called the "token" serves as temporary authorization for each device to transmit. Any device in possession of the token is allowed to act as a master device, transmitting at will. After a certain amount of time, that device must relinquish the token by transmitting the token message on the network, complete with the address of the next device. When that other device receives the token message, it switches into master mode and transmits at will. The strategy is not unlike a group of people situated at a table, where only one of them at a time holds some object universally agreed to grant speaking authority to the holder.

Token-passing ensures only one device is allowed to transmit at any given time, and it also solves the problem inherent to master-slave networks of what happens when the master device fails. If one of the devices on a token-passing network fails, its silence will be detected after the last token-holding device transmits the token message to the failed device. After some pre-arranged period of time, the last token-holding device may re-transmit the token message to the *next* device after the one that failed, re-establishing the pattern of token sharing and ensuring all devices get to "speak" their turn once more.

Examples of token-passing networks include the general-purpose Token Ring network standard (IEEE 802.5) and the defunct Token Bus (IEEE 802.4). Some proprietary industrial networks such as Honeywell's TDC 3000 network (called the *Local Control Network*, or *LCN*) utilized token-passing to arbitrate access to the network.

Token-passing networks require a substantially greater amount of "intelligence" built into each network device than master-slave requires. The benefits, though, are greater reliability and a high level of bandwidth utilization. That being said, token-passing networks may suffer unique disadvantages of their own. For example, there is the question of what to do if such a network becomes severed, so that the one network is now divided into two segments. At the time of the break, only one device will possess the token, which means only one of the segments will possess any token at all. If this state of affairs holds for some time, it will mean the devices lucky enough to be in the segment that still has the token will continue communicating with each other, passing the token to one another over time as if nothing was wrong. The isolated segment, however, lacking any token at all, will remain silent even though all its devices are still in working order and the network cable connecting them together is still functional. In a case like this, the token-passing concept fares no better than a master-slave network. However, what if the designers of the token-passing network decide to program the devices to automatically generate a new token in the event of prolonged network silence, anticipating such a failure? If the network becomes severed and broken into multiple segments, the isolated segments will now generate their own tokens and resume communication between their respective devices, which is certainly better than complete silence as before. The problem now is, what happens if a technician locates the break in the network cable and re-connects it? Now, there will be *multiple* tokens on one network, and confusion will reign!

Another example of a potential token-passing weakness is to consider what would happen to such a network if the device in possession of the token failed before it had an opportunity to relinquish the token to another device. Now, the entire network will be silent, because no device possesses the token! Of course, the network designers could anticipate such a scenario and pre-program the devices to generate a new token after some amount of silence is detected, but then this raises the possibility of the previously-mentioned problem when a network becomes severed and multiple tokens arise in an effort to maintain communication in those isolated network segments, then at some later time

the network is re-connected and now multiple tokens create data collision problems.

### CSMA

A completely different method of channel arbitration is where any and all devices have the ability to initiate communication if the network is silent. This is generally called *CSMA*, or "Carrier Sense Multiple Access." There are no dedicated master and slave devices with CSMA, nor do devices equally share "mastership" on a scheduled cycle as with token-passing. *Any* device on a CSMA network may "talk" whenever the network is free. This is analogous to an informal conversation between multiple people, where anyone is free to speak in a moment of silence.

Of course, such an egalitarian form of channel arbitration invites instances where two or more devices begin communicating simultaneously. This is called a *collision*, and must be addressed in some manner in order for any CSMA network to be practical.

Multiple methods exist to overcome this problem. Perhaps the most popular in terms of number of installed networks is *CSMA/CD* ("Carrier Sense Multiple Access with Collision Detection"), the strategy used in Ethernet. With CSMA/CD, all devices are not only able to sense an idle channel, but are also able to sense when they have "collided" with another transmitting device. In the event of a collision, the colliding devices both cease transmission, and set random time-delays to wait before re-transmission. The individual time delays are randomized to decrease the probability that a re-collision between the same devices will occur after the wait. This strategy is analogous to several peers in one group holding a conversation, where all people involved are equally free to begin speaking, and equally deferential to their peers if ever two or more accidently begin speaking at the same time. Occasional collisions are normal in a CSMA/CD network, and should not be taken as an indication of trouble unless their frequency becomes severe.

A different method of addressing collisions is to pre-assign to each device on the network a priority number, which determines the order of re-transmission following a collision. This is called *CSMA/BA*, or "Carrier Sense Multiple Access with Bitwise Arbitration," and it is analogous to several people of different social levels in one group holding a conversation. All are free to speak when the room is silent, but if two or more people accidently begin speaking at the same time, the person of highest "rank" is allowed to continue while the "lower-rank" person(s) must wait. This is the strategy used in DeviceNet, an industrial network based on CAN technology, one of the more popular data networks used in automotive engine control systems.

Some CSMA networks lack the luxury of collision detection, and must therefore strive to prevent collisions rather than gracefully recover from them. Wireless digital networks are an example where collision detection is not an option, since a wireless (radio) device having a single antenna and a single channel cannot "hear" any other devices' transmissions while it is transmitting, and therefore cannot detect a collision if one were to occur. A way to avoid collisions for such devices is to pre-assign each device on the network with a priority number, which determines how long each device is forced to wait after detecting a "quiet" network before it is allowed to transmit a new message. So long as no two devices on the network have the same "wait" time, there will be no collisions. This strategy is called *CSMA/CA*, or "Carrier Sense Multiple Access with Collision Avoidance," and is the technique used for WLAN networks (the IEEE 802.11 specification). A consequence of collision avoidance, though, is unequal access to the network. Those devices with higher-priority (shorter wait times) will always have an advantage in transmitting their data over devices of lower priority. The degree of disparity in network access grows as more devices occupy the network. CSMA/CA is analogous to a group of shy people talking, each person afraid to speak at the same time as another,

and so each person waits a different amount of time following the conclusion of the last utterance before daring to speak. This sort of ultra-polite behavior may ensure no one accidently interrupts another, but it also means the shiest person will hardly ever get a chance to speak.

A potential problem in any digital network, but particularly networks employing CSMA arbitration, is something known as *jabbering*. If a network device happens to fail in such a way that it ceaselessly transmits a signal on the network, none of the other CSMA devices will ever be allowed to transmit because they continuously detect a "carrier" signal from the jabbering device[29]. Some Ethernet components sport *jabber latch* protection circuits designed to detect jabber and automatically cut off the offending device from the network.

### 4.5.6 The OSI Reference Model

Digital data communication may be described in many ways. For example, a connection formed between two computers to exchange a text document is a multi-layered activity, involving many steps to convert human language into electrical impulses for transmission, then re-convert those electrical impulses into human language again at the receiving end. Not surprisingly, there usually exist many different ways to perform this same task: different types of networks, different encodings, different communications and presentation software, etc.

To illustrate by analogy, think of all the actions and components necessary to transport items using an automobile. In order to move furniture from an apartment to a house, for example, the following would be required:

- An appropriate vehicle
- Addresses or directions for both locations
- A driver's license and knowledge of driving rules
- Fuel for the vehicle
- Knowledge of how to safely stack furniture for transport
- Knowledge of how the furniture is to be placed in the house

These details may seem trivial to mention, as human beings familiar with the common task of moving personal belongings from one location to another, but imagine having to describe every single action and component to someone from a primitive culture ignorant of vehicles, addresses, maps, driver's licenses, fuel, etc. One way to help describe all this complexity would be to assign different people to different layers of detail. For example, an automotive engineer could discuss the details of how engines burn fuel to do mechanical work (propelling the vehicle) while a furniture loader could describe how furniture is to be loaded and taken off the vehicle. A driving instructor could then explain all the procedures of safely driving the vehicle, while a city planner could explain the organization of streets and addresses in relation to a city map. Finally, an interior decorator could wax eloquent on the proper placement of furniture in the house. Each person would be describing a different aspect of the furniture move, each one of those aspects being important to the overall goal of moving furniture from one location to another.

Moreover, for each one of the aspects described by a specialist, there may exist several different alternatives. For example, there are many different models and designs of vehicle one might use

[29] In an example this very type of failure on the job, a copper-to-fiber adapter on a personal computer's Ethernet port jammed the entire network by constantly spewing a meaningless stream of data. Fortunately, indicator lights on all the channels of the communications equipment clearly showed where the offending device was on the network, allowing it to be taken out of service for replacement.

for the job, and there may be different driving rules depending on where the two locations are for the move. Addresses and directions will *certainly* vary from city to city, and even within one city there will be alternative routes between the two locations. Finally, there is virtually no end to arrangements for furniture at the destination house, each one with its own functional and aesthetic merits.

By the same token, the task of transporting digital data may be divided into similar categories. In order to move and process data from one computer to another, the following are required:

- An appropriate cable (or radio link) connecting the two computers
- Standardized electrical signals to represent bit states
- Addresses for each computer on the network
- Algorithms specifying how each computer takes turns "talking" on the common network
- Algorithms specifying how to organize packets of data to be sent and received serially
- Software to format the data on the transmitting end and interpret the data on the receiving end

Each of these aspects is important to the overall goal of creating, moving, and interpreting digital data between two or more computers, and there are many alternative methods (standards) for each aspect. Bit values of 0 and 1 may use NRZ (Non-Return to Zero) encoding, Manchester encoding, FSK modulation, etc.; the signals may be electrical or they may be optical or they may even be radio waves; the options for electrical cables and connector types are many. Bits may be framed differently as they are packaged for transmission, and arbitration between devices on the network handled in a variety of different ways. The addressing scheme used for multiple devices on a network is important as well so messages get routed to their proper destinations.

A scheme originally intended as a formal standard, but now widely regarded as a general model to describe the portions of other standards, helps clarify the complexity of digital communications by dividing communication functions into seven distinct "layers." Developed by the *ISO* (International Organization for Standards)[30] in 1983, the *OSI Reference Model* divides communication functions into the following categories, shown in this table with examples:

[30]The acronym isn't "IOS" because "ISO" is a non-English acronym!

| | |
|---|---|
| Layer 7<br>**Application** | This is where digital data takes on practical meaning in the context of some human or overall system function.<br>*Examples: HTTP, FTP, Telnet, SSH* |
| Layer 6<br>**Presentation** | This is where data gets converted between different formats.<br>*Examples: ASCII, EBCDIC, MPEG, JPG, MP3* |
| Layer 5<br>**Session** | This is where "conversations" between digital devices are opened, closed, and otherwise managed for reliable data flow.<br>*Examples: Sockets, NetBIOS* |
| Layer 4<br>**Transport** | This is where complete data transfer is handled, ensuring all data gets put together and error-checked before use.<br>*Examples: TCP, UDP* |
| Layer 3<br>**Network** | This is where the system determines network-wide addresses, ensuring a means for data to get from one node to another.<br>*Examples: IP, ARP* |
| Layer 2<br>**Data link** | This is where basic data transfer methods and sequences (frames) are defined within the smallest segment(s) of a network.<br>*Examples: CSMA/CD, Token passing, Master/Slave* |
| Layer 1<br>**Physical** | This is where data bits are equated to electrical, optical, or other signals. Other physical details such as cable and connector types are also specified here.<br>*Examples: EIA/TIA-232, 422, 485, Bell 202* |

Figure 4.29: The OSI Reference Model for communications

The vast majority of digital networking standards in existence address mere portions of the 7-layer model. Any one of the various Ethernet standards, for example, applies to layers 1 and 2, but none of the higher-level layers. In other words, Ethernet is a means of encoding digital information in electronic form and packaging that data in a standard format understandable to other Ethernet devices, but it provides no functionality beyond that. Ethernet does *not* specify how data will be routed over large-area networks, how to manage data-exchange sessions between computers (opening connections, initiating data transfer, closing connections), or how to format the data to represent real-world variables and media. Common industrial network standards such as TIA-232 and TIA-485 don't even go that far, being limited mostly to layer 1 concerns (signal voltage levels, wiring, and in some cases types of electrical connectors). For example, TIA-485 does not specify how to address multiple devices connected to a common electrical network – all it does is specify what voltage levels represent "0" and "1" bits.

By contrast, some other industrial networking standards specify nothing about lower-level layers, instead focusing on high-level concerns. Modbus, for example, is concerned only with layer 7, and not with any of the lower-level layers[31]. This means if two or more industrial devices on a network

[31]It should be noted here that some network standards incorporating the name "Modbus" actually do specify

(such as programmable logic controllers, or PLCs) use "Modbus" to communicate with each other, it refers only to the high-level programming codes designed to poll and interpret data within those devices. The actual cable connections, electrical signals, and communication techniques used in that "Modbus" network may vary widely. Anything from TIA-232 to Ethernet to a wireless network such as WLAN may be used to actually communicate the high-level Modbus instructions between PLCs.

The following sections explore some common networking standards used for industrial instrumentation systems. The OSI Reference Model will be mentioned when and where appropriate.

## 4.6 TIA-232, 422, and 485 networks

Some of the simplest types of digital communication networks found in industry are defined by the EIA (Electronic Industry Alliance, now dissolved) and TIA (Telecommunications Industry Alliance) groups, under the numerical labels 232, 422, and 485. This section discusses these three network types.

### 4.6.1 TIA-232

The TIA-232C standard, formerly[32] known as *RS-232*, is a standard defining details found at layer 1 of the OSI Reference Model (voltage signaling, connector types) and some details found at layer 2 of the OSI model (asynchronous transfer, "handshaking" signals between transmitting and receiving devices). In the early days of personal computers, almost every PC had either a 9-pin or a 25-pin connector (and sometimes multiple of each!) dedicated to this form of digital communication. For a while, it was *the* way peripheral devices such as keyboards, printers, modems, and mice were connected to the PC. USB (Universal Serial Bus) has now all but replaced TIA-232 for personal computers, but it still lives on in the world of industrial devices.

TIA-232 networks are point-to-point, intended to connect only two devices[33]. The signaling is *single-ended* (also known as *unbalanced*), which means the respective voltage pulses are referenced to a common "ground" conductor, a single conductor used to transfer data in each direction:

lower-level concerns. *Modbus Plus* is a layer 2 standard, for example.

[32] The designation of "RS-232" has been used for so many years that it still persists in modern writing and manufacturers' documentation, despite the official status of the TIA label. The same is true for TIA-422 and TIA-485, which were formerly known as RS-422 and RS-485, respectively.

[33] "Daisy-chain" networks formed of more than two devices communicating via TIA-232 signals have been built, but they are rarely encountered, especially in industrial control applications.

Figure 4.30: Unbalanced signaling

TIA-232 specifies positive and negative voltages (with respect to the common ground conductor) for its NRZ signaling: any signal more negative than -3 volts detected at the receiver is considered a "mark" (1) and any signal more positive than +3 volts detected at the receiver is considered a "space" (0). TIA-232 transmitters are supposed to generate -5 and +5 volt signals (minimum amplitude) to ensure at least 2 volts of noise margin between transmitter and receiver.

Cable connectors are also specified in the TIA-232 standard, the most common being the DE-9[34] (nine-pin) connector. The "pinout" of a DE-9 connector for any *DTE* (Data Terminal Equipment) device at the end of an TIA-232 cable is shown here:

Figure 4.31: Pinout of a 9-pin connector for DTE

[34]Often (incorrectly) called a "DB-9" connector.

| Pin number | Assignment | Abbreviation |
|---|---|---|
| 1 | Carrier Detect | CD |
| **2** | **Received Data** | **RD** |
| **3** | **Transmitted Data** | **TD** |
| 4 | Data Terminal Ready | DTR |
| **5** | **Signal Ground** | **Gnd** |
| 6 | Data Set Ready | DSR |
| 7 | Request To Send | RTS |
| 8 | Clear To Send | CTS |
| 9 | Ring Indicator | RI |

Those terminals highlighted in **bold** font represent those connections absolutely essential for any TIA-232 link to function. The other terminals carry optional "handshaking" signals specified for the purpose of coordinating data transactions (these are the layer 2 details).

For *DCE* (Data Communications Equipment[35]) devices such as modems, which extend the TIA-232 signal path onward to other devices, the assignments of pins 2 and 3 are swapped: pin 2 is the Transmitted Data (TD) pin while pin 3 is the Received Data (RD) for a DCE device. This allows straight pin-to-pin connections between the DTE and DCE devices, so the transmit pin of the DTE device connects to the receive pin of the DCE, and visa-versa.

Figure 4.32: Connection of two DTE devices using DCE devices

If one desires to directly connect two DTE devices together using TIA-232, a special cable called a *null modem* must be used, which swaps the connections between pins 2 and 3 of each device. A "null modem" connection is necessary for the transmit pin of each DTE device to connect to the receive pin of the other DTE device:

[35] Also known by the unwieldy acronym *DCTE* (Data Circuit Terminating Equipment). Just think of "DTE" devices as being at the very end of the line, whereas "DCE" devices are somewhere in the middle, helping to exchange serial data between DTE devices.

Figure 4.33: Connection of two DTE devices using a null modem cable

The concept of a "null modem" is not unique to TIA-232 circuits[36]. Any communications standard where the devices have separate "transmit" and "receive" channels will require a "null modem" connection with transmit and receive channels swapped to be able to communicate directly without the benefit of interconnecting DCE devices. Four-wire TIA-485 and Ethernet over twisted-pair wiring are two examples of network standards where a "null" style cable is required for two DTE devices to directly connect.

TIA-232 networks may be simple, but they tend to be rather limited both in data bit rate and distance, those two parameters being inversely related. References to the TIA-232 standard repeatedly cite a maximum data rate of 19.2 kbps at 50 feet cable rate. Experimental tests[37] suggest greater rate/distance combinations may be possible in optimum conditions (low cable capacitance, minimum noise, good grounding). Since this communications standard was developed to connect peripheral devices to computers (typically within the physical span of one room), and at modest speeds, neither of these limitations were significant to its intended application.

### 4.6.2 TIA-422 and TIA-485

The next two network standards[38] are less comprehensive than TIA-232, specifying only the electrical characteristics of signaling without any regard for connector types or any layer 2 (handshaking) considerations. Within these domains, the 422 and 485 standards differ significantly from 232, their designs intended to optimize both maximum cable length and maximum data rate.

To begin with, the electrical signaling used for both TIA-422 and TIA-485 is *differential* rather than single-ended (*balanced* rather than unbalanced). This means a dedicated *pair* of wires is used for each communications channel rather than a single wire whose voltage is referenced to a common ground point as is the case with TIA-232:

[36] In fact, the concept is not unique to digital systems at all. Try talking to someone using a telephone handset held upside-down, with the speaker near your mouth and the microphone hear your ear, and it is easy to understand the necessity of having "transmit" and "receive" channels swapped from one end of a network to the other!

[37] One experiment showed the fastest data rate that an TIA-232 network could carry using a "flat" (untwisted, unshielded pair) cable less than ten feet long to be 192 kbps with occasional data corruptions. Park, Mackay, and Wright, in their book *Practical Data Communications for Instrumentation and Control* document cable lengths as long as 20 meters at 115 kbps for TIA-232, and 50 *meters* (over 150 feet!) at 19.2 kbps: over three times better than the TIA-232 standard.

[38] Former labels for TIA-422 and TIA-485 were RS-422 and RS-485, respectively back when they were "Recommended Standards". These older labels persist even today, to the extent that some people will not recognize "TIA-422" or "TIA-485."

Figure 4.34: Balanced signaling

Using dedicated wire pairs instead of single conductors sharing a common ground means that TIA-422 and TIA-485 networks enjoy much greater immunity to induced noise than TIA-232. Any electrical noise induced along the length of network cables tends to be fairly equal on all non-grounded conductors of that cable, but since the receivers in TIA-422 and TIA-485 networks response only to differential voltages (not common-mode voltages), induced noise is ignored.

The advantage differential signaling enjoys over single-ended signaling may be understood by graphical comparison. The first illustration shows how electrical noise imposed on the ungrounded conductor of a simplex communications cable becomes superimposed on the digital data signal, detected at the receiving end. Noise is modeled here as a voltage source in series along the ungrounded conductor, near the receiving end. In reality, it is more likely to be distributed along the bulk of the cable length:

Figure 4.35: Noise in single-ended signaling systems

If the superimposed noise voltage detected at the receiver has sufficient peak-to-peak amplitude to push the signal voltage above or below critical threshold levels, the receiver will interpret this as a change of digital state and cause corruptions in the data stream.

By contrast, any noise superimposed on ungrounded conductors in a differential signaling circuit cancel at the receiver, because the close proximity of those two conductors ensures any induced noise will be the same. Since the receiver responds only to *differential* voltage between its two inputs, this common-mode noise cancels, revealing a "clean" data signal at the end:

Figure 4.36: Common mode noise cancels in differential signaling systems

Both TIA-422 and TIA-485 systems use differential signaling, allowing them to operate over much longer cable lengths at much greater cable speeds than TIA-232 which is single-ended. Other high-speed network standards including Ethernet and USB (Universal Serial Bus) use differential signaling as well.

TIA-422 is a *simplex* (one-way) communications standard, whereas TIA-485 is a *duplex* (two-way) standard. Both support more than two devices on a network segment. With TIA-422, this means one transmitter and multiple receivers. With TIA-485, this may include multiple *transceivers* (devices capable of both transmitting and receiving at different times: half-duplex). Four wires are necessary to connect two such devices when full-duplex (simultaneous two-way communication) is required, and full-duplex is only practical between two devices (as shown in the previous illustration).

TIA-422 and TIA-485 specify positive and negative voltage differences (measured between each dedicated wire pair) for its signaling: any signal more negative than -200 millivolts is a "mark" (1) and any signal more positive than +200 millivolts is a "space" (0). These voltage thresholds are much lower than for TIA-232 ($\pm$ 3 volts) due to the noise-canceling properties of differential signaling. TIA-422 transmitters ("drivers") are supposed to generate -2 and +2 volt signals (minimum amplitude) to ensure at least 1.8 volts of noise margin between transmitter and receiver. TIA-485 drivers are allowed a smaller noise margin, with the minimum signal levels being -1.5 volts and +1.5 volts.

The maximum recommended cable length for both TIA-422 and TIA-485 networks is 1200 meters, which is greater than half a mile[39]. The maximum data rate is inversely dependent on cable length (just as it is for TIA-232), but substantially greater owing to the noise immunity of differential signaling. With the long cable lengths and higher data rates made possible by differential signaling, some applications may require *terminating resistors* to eliminate reflected signals. Experiments conducted by Texas Instruments demonstrate acceptable signal integrity at 200 kbps over a cable 100 feet long with no termination resistors. With a termination resistor at the receiver input (for simplex data transmission) in place on the same 100 foot cable, a data rate of 1 Mbps was achieved.

Due to the lack of standardization for cable connectors in TIA-422 and TIA-485 networks, there

[39] 1200 meters is the figure commonly cited in technical literature. However, Park, Mackay, and Wright, in their book *Practical Data Communications for Instrumentation and Control* document TIA-422 and TIA-485 networks operating with cable lengths up to 5 km (over 16,000 feet!) at data rates of 1200 bps. Undoubtedly, such systems were installed with care, using high-quality cable and good wiring practices to minimize cable capacitance and noise.

are no established pin numbers in certain connectors designated for the differential transmit and receive conductors. A common convention seen in industrial devices, though, are the labels "A" and "B", alternative labeled "-" and "+" or "A-" and "B+" in honor of their idle-state polarities (the "mark" or "1" state). In a 4-wire TIA-485 network, where full-duplex operation is possible, the terminals and connections will look something like this:

Figure 4.37: Full duplex wiring

Note the use of a ground conductor connecting both devices together. Even though the data signaling is differential and therefore does not theoretically require a common ground connection (since common-mode voltage is ignored), a ground connection helps ensure the common-mode voltage does not become excessive, since *real* receiver circuits will not properly function when exposed to certain levels of common-mode voltage.

A popular connection scheme for TIA-485 half-duplex operation is where the Transmitted Data (TD) and Received Data (RD) terminal pairs are combined, so that two-way communication may occur over one pair of wires. With such devices, it is customary to label the terminals simply as "Data" (A- and B+):

Figure 4.38: Half duplex wiring

The possibility of half-duplex operation begs the question of channel arbitration and device addressing, but since the TIA-485 standard does not specify anything outside of layer 1 concerns, these matters are left to other networking standards to fulfill. In other words, TIA-485 is not a complete data communications standard, but merely serves as the layer 1 component of other standards such as Allen-Bradley's *Data Highway* (DH), Opto 22's *Optomux*, and others.

Given the potential for high-speed communication along lengthy runs of cable using TIA-422 or TIA-485, the potential necessity of terminating resistors to prevent signal "reflection" is very real[40]. Networks operating with short cables, and/or slow data rates, may work just fine without termination resistors[41]. However, the effects of reflected signals grows more pronounced as the reflection time (time-of-flight for the signal to travel "round-trip" from one end of the cable to the other and back) approaches a substantial fraction of the bit time.

No network should have more than two termination resistors, one at each (far) end, and care should be taken to limit the lengths of all cable "stubs" or "spurs" branching off of the main "trunk"

[40] A cable that is not terminated in its characteristic impedance will develop standing voltage or current waves on it that may interfere with signaling. For additional information on this phenomenon, consult an electromagnetics textbook

[41] In fact, a great many TIA-485 networks in industry operate "unterminated" with no problems at all.

cable:

Figure 4.39: A half-duplex network that is properly terminated

The proper value for these resistors, of course, is equality with the characteristic impedance of the cable itself. A termination resistor value greater than the cable's surge impedance will still allow positive reflections of limited amplitude, while a termination resistor value less than the cable's surge impedance will still allow negative reflections of limited amplitude.

However, the inclusion of resistive loads to an TIA-422 or TIA-485 network may cause other problems. Many devices use a pair of *biasing resistors* internally to establish the "mark" state necessary for idle conditions, connecting the "A" terminal to the negative supply voltage rail through a resistor and the "B" terminal to the positive supply voltage rail through another resistor. Connecting a terminating resistor between terminals "A" and "B" will alter the voltage levels normally provided by these biasing resistors, consequently causing problems.

The following schematic diagram shows the equivalent circuit of an TIA-485 transceiver device, with and without a terminating resistor connected:

Figure 4.40: Resistances are different when terminating resistors are used

When the driver is in high-impedance (High-Z) mode, the "idle" state of the wire pair will be established by the bias resistors (equal to the supply voltage so long as there is no loading). However, a terminating resistor will act as a DC load to this biasing network, causing a substantial reduction of the "idle" state voltage toward 0 volts. Recall that -200 millivolts was the receiving threshold value for a "mark" state in both TIA-422 and TIA-485 standards (terminal "A" negative and terminal "B" positive). If the presence of a terminating resistor[42] reduces the idle state voltage to less than 200 millivolts absolute, the network's function may be compromised.

Thus, the inclusion of any terminating resistors must be accompanied by an analysis of the devices' bias resistor networks to ensure robust network operation. It is foolhardy to simply attach terminating resistors to an TIA-422 or TIA-485 network without considering their combined effect on biasing.

## 4.7 Ethernet networks

An engineer named Bob Metcalfe conceived the idea of Ethernet in 1973, while working for the Xerox research center in Palo Alto, California. His fundamental invention was the CSMA/CD method of channel arbitration, allowing multiple devices to share a common channel of communication while recovering gracefully from inevitable "collisions." In Metcalfe's vision, all of the "network intelligence" would be built directly into "controller" devices situated between the DTE devices (computers, terminals, printers, etc.) and a completely passive coaxial cable network. Unlike some other networks in operation at the time, Metcalfe's did not rely on additional devices to help coordinate communications between DTE devices. The coaxial cable linking DTE devices together would be completely passive and "dumb," performing no task but the conduction of broadcast signals between all devices. In that sense, it served the same purpose as the "luminiferous ether"

[42] Actually *two* terminating resistors in parallel, since one with be at each end of the cable! The actual DC biasing network will be more complicated as well if more than one device has its own set of internal bias resistors.

once believed to fill empty space: conducting electromagnetic waves between separated points.

The CSMA/CD ("Carrier Sense Multiple Access with Collision Detection") method of bus arbitration works by giving each Ethernet device the ability to sense an idle channel as well as sense if it happens to "collide" with another transmitting device. In the event of a collision, the colliding devices both cease transmission, and set random time-delays to wait before re-transmission. The individual time delays are randomized to decrease the probability that a re-collision between the same devices will occur after the wait. This strategy is analogous to several peers in one group holding a conversation, where all people involved are equally free to begin speaking, and equally deferential to their peers if ever two or more accidently begin speaking at the same time. Occasional collisions are perfectly normal in an Ethernet network, and should not be taken as an indication of trouble unless their frequency becomes severe.

Metcalfe's original network design operated at a data rate of 2.94 Mbps, impressive for its time. By 1980, the three American computer companies DEC (Digital Equipment Corporation), Intel, and Xerox had collaborated to revise the Ethernet design to a speed of 10 Mbps, and released a standard called the *DIX Ethernet* standard (the acronym "DIX" representing the first letter of each company's name). Later, the IEEE Local and Metropolitan Networks Standards Committee codified the DIX Ethernet standard under the numeric label 802.3. At the present time there exist many "supplemental" standards underneath the basic 802.3 definition, a few of them listed here:

- 802.3a-1985 *10BASE2 "thin" Ethernet*
- 802.3d-1987 *FOIRL fiber-optic link*
- 802.3i-1990 *10BASE-T twisted-pair cable Ethernet*
- 802.3u-1995 *100BASE-T "Fast" Ethernet and Auto-Negotiation*
- 802.3x-1997 *Full-Duplex standard*
- 802.3ab-1999 *1000BASE-T "Gigabit" Ethernet over twisted-pair cable*

The IEEE 802.3 standard is limited to layers 1 and 2 of the OSI Reference Model: the "Physical" and "Data link" layers. In the physical layer (1), the various supplements describe all the different ways in which bits are electrically or optically represented, as well as permissible cable and connector types. In the data link layer (2), the IEEE standard describes how devices are addressed (each one with a unique identifier known as a *MAC address*, consisting of a 48-bit binary number usually divided into six bytes, each byte written as a two-character hexadecimal number), and also how data frames are organized for Ethernet transmissions.

### 4.7.1 Repeaters (hubs)

Bob Metcalfe's original design for Ethernet consisted of DTE devices connected to a common coaxial cable through the use of "tee" connectors, like this:

Figure 4.41: The original Ethernet concept

This cabling arrangement suffered several problems. First, it was inconvenient to run through an office building, since each DTE device needed to be coupled rather closely to the main "trunk." Short cable segments (called *stubs*, *spurs*, or *drops*) joining the main trunk line to each DTE device could not be too long, or else they would could cause multiple signal reflections to occur in the main line. Secondly, the signal strength decreased with each "tee" connector: every time the signal branched, it would lose power. Thirdly, the need for termination resistors at the far ends of the "ether" cable opened up the possibility that those terminators might fail, fall off, or be forgotten during installation or maintenance[43].

As Ethernet evolved as a practical networking standard, one of the many improvements added to its design was the concept of a *repeating hub*. A "repeater" is an active device designed to re-broadcast a signal, usually to overcome inevitable power losses incurred as that signal propagates along a cable. Repeaters are common in the telecommunications industry, where telephone, television, and computer signals must travel hundreds or thousands of miles between points of transmission and reception. A "repeating hub" is a repeater with multiple ports for many cables to plug into, where any signal entering on any cable is repeated to *all* ports on the device. Thus, a repeating hub (or simply "hub") allows multiple Ethernet devices to interconnect with no degradation in signal quality:

[43]These very same problems may arise in FOUNDATION Fieldbus networks, for the exact same reason: the cabling is passive (for increased reliability). This makes FOUNDATION Fieldbus instrument systems challenging to properly install for most applications (except in really simple cases where the cable route is straightforward), which is its single greatest weakness (as of 2009). Ethernet's history may likely repeat itself in FOUNDATION Fieldbus at some later date: a system of reliable "hub" devices will probably be introduced so that these problems may be averted, and installations made much simpler.

Figure 4.42: A network using a hub for connections

Not only do hubs improve system performance by boosting signals' voltage levels, but they also eliminate the need for termination resistors in the network. With a hub-based system, each and every cable terminates at either a DTE or DCE device, which is (now) designed with the proper termination resistance built-in to their internal transceiver circuitry. This means each and every Ethernet cable is automatically terminated by the proper impedance simply by plugging it in to the Ethernet port of *any* device. "Stub" or "spur" cables with their length restrictions are also a thing of the past, since no cable ever splits or branches in a hub-based network system.

Hubs are considered "layer 1" devices, because they operate purely on the physical layer of Ethernet: all they do is receive Ethernet signals and re-broadcast those signals in boosted form to all other devices plugged into the hub. As a piece of interconnecting hardware, a hub is considered a DCE (Data Communications Equipment), as opposed to the end-of-cable devices such as computers and printers which are DTEs (Data Terminal Equipment).

Repeating hubs may be connected together to form larger networks. There are practical limits as to how many hubs may be "daisy-chained" together in this manner, just as there are practical limits to how long a twisted-pair cable may be (up to 100 meters). If too many hubs are cascaded, the inevitable time delays caused by the process of repeating those electrical impulses will cause problems in the network. Also, *crossover* cables that interchange the transmit and receive wires may be required to connect hubs to other hubs:

Figure 4.43: A network with cascading hubs

Since hubs are merely "layer 1" devices, mindlessly boosting and re-broadcasting signals received to their ports, their presence does not mitigate collisions between transmitting devices. As far as collisions between those devices is concerned, they might as well be directly connected together on a single piece of coaxial cable. One way to express this concept is to say that all portions of the network are part of the same *collision domain*. In other words, a collision happening in one portion of the network happens in *all* portions of the network.

### 4.7.2 Ethernet cabling

Along with hubs came another form of Ethernet cable and connector: *unshielded, twisted pair* (UTP) wiring and *RJ-45* "flat" connectors. These cables use multiple twisted pairs of wires instead of the coaxial cable specified in Metcalfe's original Ethernet. The purpose of using twisted-wire pairs is to reduce magnetic signal coupling (for more information on this, refer to section A.3.6 beginning on page 1202).

Figure 4.44: RJ-45 wiring for Ethernet

For 10 Mbps Ethernet over UTP cable (called 10BASE-T) and for 100 Mbps Ethernet (called 100BASE-TX), only two[44] out of four available wire pairs are used:

[44] With only half the available wire pairs used in a standard 10 Mbps or 100 Mbps Ethernet cable, this opens the

| Pin number | Assignment | Abbreviation |
|---|---|---|
| 1 | Transmit Data (+) | TD+ |
| 2 | Transmit Data (-) | TD- |
| 3 | Receive Data (+) | RD+ |
| 4 | (not used) | |
| 5 | (not used) | |
| 6 | Receive Data (-) | RD- |
| 7 | (not used) | |
| 8 | (not used) | |

It should be noted that 1000 Mbps ("Gigabit") Ethernet over twisted-wire pairs does in fact use all four pairs in an eight-wire cable: a departure from traditional UTP Ethernet cable wiring.

| Pin number | Assignment | Abbreviation |
|---|---|---|
| 1 | Pair "A" (+) | BI_DA+ |
| 2 | Pair "A" (-) | BI_DA- |
| 3 | Pair "B" (+) | BI_DB+ |
| 4 | Pair "C" (+) | BI_DC+ |
| 5 | Pair "C" (-) | BI_DC- |
| 6 | Pair "B" (-) | BI_DB- |
| 7 | Pair "D" (+) | BI_DD+ |
| 8 | Pair "D" (-) | BI_DD- |

Along with UTP cables and RJ-45 connectors came a significant alteration to the basic electrical scheme of Ethernet. Metcalfe's original design used a simple coaxial cable as the "ether" connecting devices together. Such cables had only two conductors, meaning each device needed to transmit *and* receive data over the same two conductors. With UTP cable's four pairs of conductors, transmission and reception of signals is handled over different wire pairs. This means connections made between Ethernet devices must employ a "swap" between TD and RD wire pairs in order for communication to take place, so that the "receiver" circuitry of one device connects to the "transmitter" circuitry of the other, and visa-versa. This is precisely the same problem experienced inherent to TIA-232 and four-wire TIA-485 networks, where separate wire pairs for "transmit" and "receive" are different.

In a typical Ethernet system, the interconnecting hubs perform this transmit/receive swap. Hubs are considered DCE devices, while computers and other end-of-the-line devices are considered DTE devices. This means the pin assignments of DTE and DCE devices must be different in order to ensure the transmit/receive pin swap necessary for straight-through cables to work. This also means if someone ever wishes to directly connect two Ethernet DTE devices together without the benefit of a hub in between, a special *crossover* cable must be used for the connection, identical in function to the *null modem* cable used to connect two TIA-232 DTE devices together:

possibility of routing *two* Ethernet channels over a single four-pair UTP cable and RJ-45 connector. Although this is non-standard wiring, it may be a useful way to "squeeze" more use out of existing cables in certain applications. In fact, "splitter" devices are sold to allow two RJ-45-tipped cables to be plugged into a single RJ-45 socket such that one four-pair cable will then support two Ethernet pathways.

Figure 4.45: Ethernet connections between DTE devices require crossover cables

Furthermore, the same problem exists when multiple hubs are connected to form larger networks. Since each hub is a DCE device, a straight-through cable connecting two hubs together will pass transmitted signals from one hub directly to the "transmit" pins of the other hub, not the "receive" pins as it needs to. Consequently, a "crossover" cable should be used to connect two Ethernet hubs together in order to avoid this problem:

Figure 4.46: Ethernet connections between DCE devices also require crossover cables

Some early Ethernet hubs provided a different solution to the "crossover" problem, and that was a crossover *switch* built into the hub, allowing a person to manually switch the transmit and receive wire pairs with the push of a button. In this next photograph of a four-port Ethernet hub, the "Normal/Uplink" pushbutton appears on the right-hand side of the front panel, controlling the furthest-right port of the hub. This switch is supposed to be placed in the "Normal" position if the device plugged into that port is a DTE device, and placed in the "Uplink" position if the device is a DCE device (e.g., another hub):

Figure 4.47: An Ethernet hub

Note the LED indicator lights by each port on the hub. One LED indicates whether or not the cable is active (when a powered Ethernet DTE device is plugged into that port of the hub), while the other LED indicates traffic on the cable (by blinking). These LEDs are very helpful for identifying a crossover problem. This hub even has an LED indicating the occurrence of collisions (the "Col" LED just below the main power LED), giving simple visual indication of collision frequency.

Some modern hubs use auto-sensing technology to perform any necessary transmit/receive pin swaps, rendering crossover cables and crossover pushbuttons unnecessary for hub-to-hub connections. 1000BASE-T ("Gigabit" Ethernet) hubs have this as a standard feature.

### 4.7.3 Switching hubs

The next evolutionary step in Ethernet network connections is the introduction of a *switching hub*, or simply *switch*. A "switch" looks exactly like a repeating hub, but it contains intelligence to route transmitted signals only to specific ports, rather than broadcasting every received data frame to all ports. What enables this to happen is the information contained in each Ethernet frame transmitted by DTE devices.

Refer to Figure 4.23. Note that part of the frame includes both a source address and a destination address. These refer to the 48-bit "MAC" addresses uniquely identifying each and every Ethernet device. A switching hub "learns" the identities of all devices plugged into each of its ports by remembering the "source" addresses received through those ports. When a switch receives an Ethernet frame with a destination address it recognizes as residing on one of its ports, it *only* repeats that frame to that specific port, and not to the other ports. This reduces the amount of "traffic" seen at the other ports, and also avoids unnecessary collisions because frames only get sent to their intended destinations. If a switch receives a data frame with an unrecognized destination address, it defaults to basic "hub" behavior by broadcasting that frame to all ports.

The presence of a switching hub in a larger network has the effect of dividing that network into separate collision domains, so that a collision occurring in one domain does not "spill over" into

another domain where it would delay communication between those devices:

Figure 4.48: Ethernet switches create domains in the network

Of course, collisions between these two domains may still occur, for instance if a device in the first domain tries to transmit to a device in the second domain at the exact same time that a device in the second domain attempts to transmit to a device in the first.

With this added intelligence, switching hubs are considered "layer 2" devices, since they operate not just at the physical layer of electrical impulses, but also at the next layer of device addressing. Since switching hubs add benefit beyond repeating hubs without any drawbacks[45], most people elect to use switches whenever possible.

## 4.8 Internet Protocol (IP)

Although the Internet made use of several high-capacity networks (called *backbones*) connecting large data centers in different cities, the real "magic" of the Internet does not reside in any particular cable or link. Instead, the Internet is so widespread and accessible because of a *protocol* allowing for the free exchange of data along and between disparate systems. This "protocol" allows digital data to be packaged in such a way that it could be sent along nearly any kind of communications link (from copper wires to fiber-optic to radio waves) – and indeed along multiple pathways between the same two points – while arriving at the destination intact. Thus, the Internet is akin to a random patchwork of existing communications pathways pressed into coordinated service by the sharing of a common "language." In this section, the protocol at the heart of the Internet, appropriately called *Internet Protocol*, or *IP*, will be investigated.

Physical network standards such as Ethernet only define aspects relevant to lower layers of the OSI Reference Model. While these details are essential for communication to occur, they are not enough on their own to support a wide-spread communications system. For this reason, network

[45] Even the cost difference is slight.

standards such as TIA-485 and Ethernet almost always comprise the lower layer(s) of a more complex communications protocol capable of managing higher-order addresses, message integrity, "sessions" between computers, and a host of other details.

Internet Protocol (IP) manages network addresses and data handling over a much larger physical domain than Ethernet is able to. The basic principle of IP is that large messages are broken down into *packets* transmitted individually and received individually (then reassembled at the receiver to form the original, complete message). An analogy for this process might be an author with a printed paper manuscript for a book, who needs to get her manuscript to a print shop across town. Unfortunately, the mail service in this town cannot handle the bulky manuscript in one piece, so the author divides the manuscript into bundles of 10 pages each and mails each of these bundles to the print shop, with instructions in each envelope on how to re-assemble the bundles into the complete book. The individual bundles may not make it to the print shop on the same day, or even in the correct order, but the instructions contained within each one make it possible for the people at the print shop to reassemble the entire manuscript once all the bundles have arrived.

This strategy for transmitting large digital messages is at the heart of the Internet: data sent from one computer to another over the Internet is first broken down into packets, which are then sent and routed over a variety of pathways to their destination. The receiving computer then reassembles the packets into the original form. This "fragmentation" of data may seem unnecessary, but it actually provides a great deal of flexibility in how data is routed from one point to another.

### 4.8.1 IP addresses

IP is a "layer 3" technology, being concerned with network-wide addresses for routing information between two different locations. IP is not concerned with the details of communication along any particular wire or fiber-optic cable. It is not "aware" of how bits are represented electrically, or what kind of connectors are used to couple cables together. IP is only concerned with "networks" in the broad sense of the word, as abstract collections of computers that *somehow* (it doesn't care exactly how) are connected to each other.

Networking equipment (DCE) designed to pay attention to IP addresses for routing purposes are called, not surprisingly, *routers*. Their purpose is to direct packets to their appropriate destinations in the shortest amount of time.

In order for the Internet Protocol to specify where packets are coming from and where they are going to, each source and destination must bear its own *IP address*. IP version 4 (IPv4) uses 32-bit addresses, usually expressed as four octets (four bytes) written using decimal numbers. For example:

IP address `00000000 00000000 00000000 00000000` is written as `0.0.0.0`
IP address `11111111 11111111 11111111 11111111` is written as `255.255.255.255`
IP address `10101001 11111010 00101101 00000011` is written as `169.250.45.3`

In order for two inter-connected computers to exchange data using Internet Protocol, each one must have a unique IP address:

Figure 4.49: IP communication requires devices to have unique IP addresses

At first, this may seem redundant. Doesn't each and every Ethernet device already have its own unique "MAC address" 48 bits in length to distinguish it from every other Ethernet device in existence? If so, why add *another* set of identifying addresses to the system?

This is true – Ethernet devices are already uniquely addressed – but those MAC addresses serve different purposes than IP addresses. Recall that Ethernet is a standard only at layers 1 and 2, and is not "aware" of any higher-level concerns. Ethernet MAC addresses are useful to switching hubs and other Ethernet DCE devices tasked with management of Ethernet data frames, but those MAC addresses – unique as they may be – have little relevance in the greater picture of IP where messages must be fragmented and reassembled over very large-scale networks. More importantly, the reason IP addresses are needed is to be able to use interconnecting networks other than Ethernet. For example, two computers may be connected to each other with a simple TIA-232 cable (or even using radio transceiver units for a "wireless" connection) instead of Ethernet, but still use Internet Protocol to break up large messages and reassemble them at the receiving end[46]. By having its own dedicated addressing scheme, IP ensures computers may be able to disassemble data into packets, send those packets, receive those packets, then re-assemble the packets into the original data *regardless of the physical interconnection details, channel arbitration methods, or anything else in between.* In a sense, IP is the "glue" that holds disparate networks together, and makes something like a world-wide Internet possible when so many different network types exist to connect digital devices together.

A helpful analogy is to think of Ethernet MAC addresses like Social Security numbers for United States citizens. Each US citizen should have their own unique Social Security number, shared by no one else. This number is used for many purposes, including identification on Federal tax documents, to help route specific information (such as income records and Social Security payments) to the proper people. Despite the uniqueness of these numbers, though, people still need separate mailing addresses in order to receive mail through the postal service and other package distribution agencies. The mailing address serves a different purpose than the Social Security "address" each US citizen possesses. Furthermore, the existence of separate mailing addresses ensures even non-citizens living in the United States (e.g., foreign students, ambassadors, etc.) who have no Social Security numbers still have a way to send and receive mail.

[46]In fact, this is precisely the state of affairs when using a *dial-up* telephone connection to link your personal computer with the Internet. On dial-up, a PC may not use Ethernet at all to make the connection to the telephone provider's network, but rather it might use TIA-232 or USB to a modem (modulator/demodulator) device, which turns those bits into modulated waveforms transmittable over a voice-quality analog telephone line.

### The "ping" utility

Computers enabled to communicate using Internet Protocol (IP) are equipped with a utility program named `ping` useful for detecting the presence of other IP-enabled computers connected to the same network. The classic format of this program is execution by typing the word "ping" at the computer's command-line interface followed by the IP address of the other device. For example, to check for the presence of a computer on the network with an IP address of 133.82.201.5, type this command on the computer's command line and press the "Enter" key:

```
ping 133.82.201.5
```

The `ping` utility works by sending a very short digital message to the specified IP address, requesting a reply from that computer. There are usually multiple attempts. The `ping` command as implemented on the Microsoft Windows (XP) operating system typically makes four attempts before quitting. Some other operating systems' implementation of `ping` continue indefinitely until halted by the user with the "Control-C" keystroke interrupt combination.

When diagnosing problems with IP-enabled network devices, few utilities are as immediately useful as `ping`. Networking professionals commonly use the word "ping" as a verb, as in "I tried to *ping* that computer, but it gave no response." There are many reasons why a computer might fail to respond to a `ping` query, but a successful `ping` attempt proves several things:

- The destination device is powered up and its IP functionality is working
- All network devices (DCE) between your computer and the destination device are communicating
- All cables necessary for the communication of data between your computer and the destination are functional
- Both your computer and the destination device are on the same *subnet* (this topic covered in more detail later)

### IPv4 address ranges

Given the addressing purpose of Internet Protocol (to designate addresses over an extremely large collection of digital communication devices), addresses must be chosen with care. IP version 4 uses a 32-bit field to designate addresses, limiting its address capacity to $2^{32}$ unique addresses. As large as this number is, it is not enough to uniquely identify all Internet-capable devices worldwide. The inventors of IP did not dream their Internet would grow to the proportions it has today. Let this be a lesson to all those involved with computers: the future is usually bigger than can be foreseen! A variety of clever techniques has been developed to deal with this shortage of IP addresses. One of them is to dynamically assign addresses to Internet-connected computers *only when they are turned on*. This is how most personal Internet connections work: when booting up a personal computer to connect to the Internet, the service provider assigns a temporary IP address through a protocol called DHCP (Dynamic Host Configuration Protocol). The provider then forces this temporary IP address to be relinquished on shut down of the computer, so someone else may use it for theirs.

The *Internet Corporation for Assigned Names and Numbers*, or *ICANN*, is the organization responsible[47] for assigning IP addresses to Internet users worldwide (among other tasks). This group has designated certain IP address ranges specific to internal (i.e., *Local Area Network*, or

[47] Prior to ICANN's formation in 1999, the *Internet Assigned Numbers Authority*, or *IANA* was responsible for these functions. This effort was headed by a man named Jon Postel, who died in 1998.

*LAN*) network devices, which shall never be used "publicly" to address devices on the world-wide Internet. These specially-designated "private" LAN address ranges are as follows:

`10.0.0.0` to `10.255.255.255`
`172.16.0.0` to `172.31.255.255`
`192.168.0.0` to `192.168.255.255`

Additionally, all computers have their own special *loopback* IP address, used to send IP message packets to itself for certain purposes (including diagnostics): `127.0.0.1`. This IP address is completely *virtual*, not associated with any network hardware at all[48]. Therefore, the `ping` command executed on any computer should *always* be able to detect address 127.0.0.1, regardless of the status or even existence of actual network hardware (cards or interfaces) on that computer. Failure of the `ping` command to detect the loopback address is a sign that the computer's operating system is not configured to use Internet Protocol.

A computer's loopback address may have uses other than diagnostic. Some computer applications are network-oriented by nature, and rely on IP addresses even if the application is performing some local function rather than a function between computers on an actual network. The *X-windows* graphic-user interface (GUI) system popularly used on UNIX operating systems is an example of this, referencing the loopback address to form a connection between client and server applications running on the same computer.

### 4.8.2 Subnetworks and subnet masks

IP (version 4) addresses are used in conjunction with something called *subnet masks*[49] to divide IP networks into "subnetworks." A "subnetwork" is a range of IP-addressed devices allowed to communicate with each other. A subnet mask may be thought of as sort of "filter" used to identify IP addresses not belonging to the proper range.

The subnet mask works as a bitwise filter, identifying those bits in the IP address defining the subnetwork. For example, if the subnet mask on a computer is set to `255.0.0.0` (binary `11111111 00000000 00000000 00000000`), it means the first 8 bits of the IP address define the subnetwork, and thus the computer is only allowed to communicate with another computer belonging to the same subnetwork (i.e., having the same first octet in its IP address).

A set of examples showing two interconnected computers with differing IP addresses (and in some cases, different masks) illustrates how this works[50]. In the first example, two computers with IP addresses differing in the last two octets are able to communicate because their subnets are the same (`169.254`):

[48] The term "loopback" refers to an old trick used by network technicians to diagnose suspect serial port connections on a computer. Using a short piece of copper wire (or even a paperclip) to "jumper" pins 2 and 3 on an TIA-232 serial port, any serial data transmitted (out of pin 3) would be immediately received (in pin 2), allowing the serial data to "loop back" to the computer where it could be read. This simple test, if passed, would prove the computer's low-level communication software and hardware was working properly and that any networking problems must lie elsewhere.

[49] Also called "netmasks" or simply "masks."

[50] These are real test cases performed between two computers connected on a 10 Mbps Ethernet network. The error messages are those generated by the `ping` utility when communication was attempted between mis-matched computers.

Figure 4.50: IP connections on a subnet

The `ping` command may be used to check to see the IP addresses and subnet masks are correct. A screenshot of `ping` being used on a personal computer running the Microsoft Windows XP operating system is shown here:

Figure 4.51: The `ping` utility

In the next example, two computers with the same mask value are shown, but with different address values in the octets designated by their masks. In other words, these two computers belong to different subnets: one to `167.254` and the other to `169.254`, and as a result they are not allowed to communicate with each other using Internet Protocol. The resulting error messages generated by the `ping` utility are shown in this diagram:

Figure 4.52: The same subnet masks preventing communication

In the last example, two computers having different mask values as well as different IP addresses are shown. The subnet of the left-hand computer is `169.254.10` while the subnet of the right-hand computer is `169.254`:

Figure 4.53: Different subnet masks preventing communication

The computer on the left may only communicate with IP addresses matching in the first three octets (`169.254.10`). Seeing that the destination address for the second computer does not match in its third octet, `ping` returns a "Destination host unreachable" error message when executed from the left-hand computer.

When the computer on the right attempts to communicate with ("ping") the computer on the left, it is allowed to transmit to that computer because its mask only screens for agreement in the first two octets (`169.254`), which happen to match. However, the computer on the left is not allowed to transmit to the computer on the right because of its more restrictive subnet, and so `ping` running on the right-hand computer returns a "Request timed out" error message because it never receives a reply from the left-hand computer to any of its queries.

With just two computers connected by a single cable, the concept of subnetworks and masks seems useless, and indeed it is on this small of a scale. However, "subnetting" is a useful technique for managing high traffic loads on large networked systems using IP addresses, and so it is commonly seen in many local area networks (LANs) such as those found at industry and commercial sites.

Another use of `ping` is to search for unknown IP addresses on a known subnet. This may be done by "pinging" to the *broadcast address* for that subnet: an IP address formed by the known subnet numbers, followed by all binary 1's filling the unknown bit spaces. For example, `ping` can be used to search for devices on the subnet `156.71` (subnet mask `255.255.0.0`) by using the following command:

```
ping 156.71.255.255
```

### 4.8.3 IP version 6

The next version of IP (version 6, or IPv6) uses 128-bit addresses, giving $2^{128}$ address possibilities (in excess of $3.4 \times 10^{38}$), in stark contrast to IPv4's paltry $2^{32}$ address space. To put this enormous quantity into perspective, there are enough IPv6 addresses to designate nearly 57 *billion* of them for each and every gram of the Earth's mass[51]. While IPv4 addresses are typically written as four octets in decimal form (e.g., 169.254.10.5), this notation would be very cumbersome for writing IPv6 addresses. Thus, IPv6 addresses are written as a set of eight hexadecimal numbers (up to four characters per number) separated by colons, such as `4ffd:522:c441:d2:93b2:f5a:8:101f`. The phase-in of IPv6 to replace IPv4 has already started for certain portions of the Internet, but the full transition to IPv6 is expected to take many years. The IPv6 "loopback" virtual address for computers is `0:0:0:0:0:0:0:1`, or more simply[52] written as `::1`.

Note the "shorthand" notation used in the previous IPv6 addresses to eliminate extra characters: some of the 16-bit segments are truncated to less than four hexadecimal characters if the preceding (more-significant) characters are zero. Thus, `:522:` will be seen instead of `:0522:`, and `:d2:` instead of `:00d2:`. The loopback address of `::1` is the ultimate shorthand notation, collapsing all prior segments (which are all zero) into a pair of back-to-back colons.

IP version 6 supports subnetworks just as IPv4 does, but instead of denoting subnet masks in the same colon-delimited fashion as IPv6 addresses, IPv6 subnet masks are simply specified by the number of "1" bits beginning from the first (MSB). The rationale here is that subnet mask bits should be contiguous[53], with no zero bits separating one bits. This being the case, the subnet mask for any practical IP range may be specified as a simple number of 1's from the MSB down to the LSB[54].

It should be noted that an updated version of the `ping` command (called `ping6`) is available to help diagnose IPv6 systems.

### 4.8.4 DNS

The acronym *DNS* actually stands for two related things: *Domain Name System* and *Domain Name Server*. The first meaning of "DNS" refers to the system of exchanging numerical IP addresses with alphanumeric *Uniform Resource Locators* (*URLs*) which are easier for human beings to remember. When a web browser software is used to navigate to a web site on the Internet, the user has the option of entering the URL *name* of that site (e.g., `www.google.com`) or a numerical IP address (e.g., `75.125.53.104`). Special computers connected to the Internet called *Domain Name Servers*, and *Domain Name Resolvers* (*DNRs*) use the *Address Resolution Protocol* (*ARP*) to convert your target web site name to its actual IP address so that a connection may be made between that computer and yours.

[51] According to Douglas Giancoli's *Physics for Scientists and Engineers* textbook, the mass of the Earth is $5.98 \times 10^{24}$ kg, or $5.98 \times 10^{27}$ grams. Dividing $2^{128}$ (the number of unique IPv6 addresses) by the Earth's mass in grams yields the number of available IPv6 address per gram of Earth mass. Furthermore, assuming a grain of sand has a mass of about 1 milligram, and that the Earth is modeled as a very large collection of sand grains (not quite the truth, but good enough for a dramatic illustration!), there will be 57 *million* IPv6 addresses per grain of sand on Earth.

[52] The fully-written loopback address is actually `0000:0000:0000:0000:0000:0000:0000:0001`.

[53] While it is possible to use non-contiguous subnet mask values, the practice is frowned upon by most system administrators.

[54] Indeed, subnet masks for IPv4 can be specified in this manner as well, not just IPv6 subnet masks.

ICANN, the same organization responsible for allotting IP addresses, also maintains databases for all registered domain names.

### 4.8.5 Command-line diagnostic utilities

In addition to `ping`, another utility program useful for troubleshooting network connections from a computer's command line interface is `ipconfig`. When executed, `ipconfig` returns a listing of all available (configured and operating) network interfaces on that computer:

```
C:\Documents and Settings\btc>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 169.254.1.2
        Subnet Mask . . . . . . . . . . . : 255.255.0.0
        Default Gateway . . . . . . . . . :

Ethernet adapter Wireless Network Connection:

        Media State . . . . . . . . . . . : Media disconnected

C:\Documents and Settings\btc>
```

Figure 4.54: Use of the `ipconfig` utility from a Windows command prompt

The equivalent command for UNIX operating systems is `ifconfig`, shown in this screenshot:

```
root@Renegade2:/home# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:13:20:08:ec:e6
          inet addr:192.168.0.64  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::213:20ff:fe08:ece6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:170901 errors:0 dropped:0 overruns:0 frame:0
          TX packets:107550 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:212178154 (212.1 MB)  TX bytes:14005068 (14.0 MB)

eth1      Link encap:Ethernet  HWaddr 00:0e:35:a2:1b:7f
          inet6 addr: fe80::20e:35ff:fea2:1b7f/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:441 errors:0 dropped:0 overruns:0 frame:0
          TX packets:570 errors:0 dropped:6 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:18 Base address:0x2000 Memory:48005000-48005fff

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:516 (516.0 B)  TX bytes:516 (516.0 B)

root@Renegade2:/home#
```

Figure 4.55: Use of the `ipconfig` utility from a UNIX command prompt

Some of the more interesting details contained in the output from `ifconfig` are the IPv6 addresses (in addition to IPv4 addresses), and details on the "loopback" address (IPv4 `127.0.0.1` or IPv6 `::1`).

A utility intended to reveal the DNS name of a computer given its IP address, or visa versa, is `nslookup`. The same command works on Microsoft Windows and UNIX operating systems alike. Here, the UNIX version used to identify four IP addresses of the popular Google search engine web site is shown:

```
root@Renegade2:/home# nslookup www.google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
www.google.com  canonical name = www.l.google.com.
Name:   www.l.google.com
Address: 74.125.53.103
Name:   www.l.google.com
Address: 74.125.53.147
Name:   www.l.google.com
Address: 74.125.53.99
Name:   www.l.google.com
Address: 74.125.53.104
```

Figure 4.56: Use of the `nslookup` utility from a UNIX command prompt

Another utility used to explore network connections is `traceroute` (spelled `tracert` on Microsoft Windows operating systems). This utility sends a test packet to the designated destination address, returning information on all the "hops" the IP packet takes between computers along the network to reach its destination and the amount of time taken to make the trip. Execution of `traceroute` on a UNIX computer and `tracert` on a Microsoft Windows computer are shown here:

```
root@Renegade2:/home# traceroute www.google.com
traceroute to www.google.com (74.125.53.147), 30 hops max, 40 byte packets
 1  home (192.168.0.1)  4.763 ms  4.803 ms  4.784 ms
 2  tukw-dsl-gw22-214.tukw.qwest.net (63.231.10.214)  57.927 ms  59.993 ms  61.916 ms
 3  tukw-agw1.inet.qwest.net (71.217.184.169)  63.924 ms  65.905 ms  67.882 ms
 4  sea-core-01.inet.qwest.net (67.14.1.194)  71.775 ms  71.784 ms  73.609 ms
 5  sea-brdr-01.inet.qwest.net (205.171.26.54)  75.642 ms  77.442 ms  79.421 ms
 6  63.146.26.198 (63.146.26.198)  81.438 ms  67.052 ms  68.856 ms
 7  sl-gw20-sea-0-0-0.sprintlink.net (144.232.6.8)  70.633 ms  56.617 ms  60.219 ms
 8  sl-googl13-199181-0.sprintlink.net (144.224.13.138)  62.133 ms  64.301 ms  66.162
 9  209.85.249.32 (209.85.249.32)  68.140 ms 209.85.249.34 (209.85.249.34)  70.028 ms
10  216.239.46.204 (216.239.46.204)  79.865 ms  81.739 ms  85.587 ms
11  64.233.174.121 (64.233.174.121)  249.193 ms 64.233.174.129 (64.233.174.129)  113.
12  72.14.232.70 (72.14.232.70)  93.458 ms 72.14.232.10 (72.14.232.10)  99.574 ms 72.
13  72.14.232.6 (72.14.232.6)  68.876 ms 72.14.232.2 (72.14.232.2)  70.251 ms pw-in-f
root@Renegade2:/home#
```

Figure 4.57: Use of the `traceroute` utility from a UNIX command prompt

```
U:\>tracert www.google.com

Tracing route to www.l.google.com [209.85.173.99]
over a maximum of 30 hops:

  1  4294964928 ms  4294964927 ms  4294964927 ms  134.39.250.1
  2  4294964927 ms  4294964927 ms  4294964927 ms  bellingham-2691.ctc.edu [192.6
4.1.105]
  3  4294964931 ms  4294964931 ms  4294964930 ms  ge-0-1-0--941.seawescar1.infra
.wa-k20.net [68.179.207.210]
  4  4294964931 ms  4294964931 ms  4294964930 ms  ge-3-0-3--0.seawescore1.infra.
wa-k20.net [68.179.203.26]
  5  4294964931 ms  4294964931 ms  4294964931 ms  ge-2-2-0--311.iccr-sttlwa01-02
.infra.pnw-gigapop.net [209.124.188.182]
  6  4294964931 ms  4294964930 ms  4294964931 ms  pnwgp-cust.tr01-sttlwa01.trans
itrail.net [137.164.131.186]
  7  4294964931 ms  4294964931 ms  4294964931 ms  te4-3--301.tr01-sttlwa01.trans
itrail.net [137.164.131.185]
  8  4294964931 ms  4294964931 ms  4294964931 ms  137.164.130.158
  9  4294964931 ms  4294964931 ms  4294964931 ms  209.85.249.32
 10  4294964935 ms  4294964938 ms  4294964938 ms  216.239.46.208
 11  4294964936 ms  4294964937 ms  4294964937 ms  64.233.174.127
 12  4294964937 ms  4294964936 ms  4294964937 ms  209.85.251.149
 13  4294964943 ms  4294964937 ms  4294964941 ms  209.85.251.145
 14  4294964941 ms  4294964944 ms  4294964937 ms  mh-in-f99.google.com [209.85.1
73.99]

Trace complete.

U:\>_
```

Figure 4.58: Use of the `tracert` utility from a Windows command prompt

## 4.9 Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)

At the next OSI Reference Model layer (layer 4) is a set of protocols specifying virtual "ports" at transmitting and receiving devices through which data is communicated. The purpose of these virtual ports is to manage multiple types of data transactions to and from the same IP address, such as in the case of a personal computer accessing a web page (using HTTP) and sending an email message (using SMTP) at the same time. An analogy to help understand the role of ports is to think of multiple packages delivered to different people at a common address such as a business office. The mailing address for the office is analogous to the IP address of a computer exchanging data over a network: it is how other computers on the network "find" that computer. The persons' names or department numbers written on the different packages are analogous to virtual ports on the computer: "places" where specific messages are directed once they arrive at the common address.

*Transmission Control Protocol* ( *TCP*) and *User Datagram Protocol* (*UDP*) are two methods used to manage "ports" on a DTE device, with TCP being the more complex (and robust) of the two. Both TCP and UDP must rely on IP addressing to specify which devices send and receive data, which is why these protocols are often listed in conjunction with IP (e.g., TCP/IP and UDP/IP). TCP and UDP are both useless on their own: a protocol specifying port locations without an IP address would be as meaningless as a package placed in the general mail system with just a name or a department number but no street address. Conversely, Internet Protocol anticipates the presence of a higher-level protocol such as TCP or UDP by reserving a portion of its "datagram" (packet) bit space for a "protocol" field[55] to specify which high-level protocol generated the data in the IP packet.

[55]Both IPv4 and IPv6 reserve eight bits for this purpose.

TCP is a complex protocol specifying not only which virtual "ports" will be used at the sending and receiving devices, but also how packet transmission will be guaranteed. A *segment* of data sent via TCP will be error-checked to guard against corruption, marked with special bit flags if of an "urgent" priority, and otherwise marked and labeled in such a way that a high degree of communication reliability is assured. A simplified analogy for TCP's actions might be *certified mail* in the United States postal system, where certain extra steps are taken to ensure delivery and receipt of a parcel.

UDP is a much simpler protocol, lacking many of the data-integrity features of TCP. It is quite common to see UDP applied in industrial settings, where communication takes place over much smaller networks than the world-wide Internet. Another reason UDP is more common in industrial applications is that it is easier to implement in the "embedded" computer hardware at the heart of many industrial devices. The TCP algorithm requires greater computational power and memory capacity than the UDP algorithm, and so it is much easier to engineer a single-chip computer (i.e., microcontroller) to implement UDP than it would be to implement TCP.

A noteworthy exception to the use of UDP in industrial applications is FOUNDATION Fieldbus HSE (High-Speed Ethernet), which uses TCP rather than UDP to ensure reliable transmission of data.

Using another utility program on a personal computer called `netstat` (available for both Microsoft Windows and UNIX operating systems) to check active connections[56], the various IP addresses and their respective port numbers (shown by the digits following the colon after the IP address) are shown as a list, organized by TCP connections and UDP connections:

```
C:\Documents and Settings\btc>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2869           0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1025         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:5152         0.0.0.0:0              LISTENING
  TCP    169.254.1.2:23         169.254.1.1:1116       ESTABLISHED
  TCP    169.254.1.2:139        0.0.0.0:0              LISTENING
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:1062           *:*
  UDP    0.0.0.0:4500           *:*
  UDP    0.0.0.0:7725           *:*
  UDP    127.0.0.1:123          *:*
  UDP    127.0.0.1:1063         *:*
  UDP    127.0.0.1:1066         *:*
  UDP    127.0.0.1:1900         *:*
  UDP    169.254.1.2:123        *:*
  UDP    169.254.1.2:137        *:*
  UDP    169.254.1.2:138        *:*
  UDP    169.254.1.2:1900       *:*

C:\Documents and Settings\btc>
```

Figure 4.59: Use of the `netstat` utility from a Windows command prompt

Many different port numbers have been standardized for different applications at OSI Reference Model layers above 4 (above that of TCP or UDP). Port 25, for example, is always used for SMTP (Simple Mail Transfer Protocol) applications. Port 80 is used by HTTP (HyperText

[56] In this particular case, the command structure uses `netstat -an` to specify *all* (`a`) ports with *numerical* (`n`) IP addresses and port numbers shown.

Transport Protocol), a layer-7 protocol used to view Internet "web" pages. Port 107 is used by TELNET applications, a protocol whose purpose it is to establish command-line connections between computers for remote administrative work. Port 22 is used by SSH, a protocol similar to TELNET but with significantly enhanced security. Port 502 is designated for use with Modbus messages communicated over TCP/IP.

## 4.10 The HART digital/analog hybrid standard

A technological advance introduced in the late 1980's was *HART*, an acronym standing for **H**ighway **A**ddressable **R**emote **T**ransmitter. The purpose of the HART standard was to create a way for instruments to digitally communicate with one another over the same two wires used to convey a 4-20 mA analog instrument signal. In other words, HART is a *hybrid* communication standard, with one variable (channel) of information communicated by the analog value of a 4-20 mA DC signal, and another channel for digital communication whereby many other variables could be communicated using pulses of current to represent binary bit values of 0 and 1. Those digital current pulses are *superimposed* upon the analog DC current signal, such that the same two wires carry both analog and digital data simultaneously.

### 4.10.1 Basic concept of HART

Looking at a standard loop-powered (2-wire) process transmitter circuit shown below, there is a transmitter, a DC power supply (voltage source), and usually a 250 ohm resistor to create a 1 to 5 volt signal readable by any voltage-sensing indicator, controller, or recorder:

Figure 4.60: A typical analog 4-20 mA signal loop

The transmitter's primary function in this circuit is to regulate current to a value representative of the measured process variable (e.g., pressure, temperature, flow, etc.) using a range of 4 to 20 mA, while the DC voltage source provides power for the transmitter to operate. Loop-powered instruments are very common in industrial instrumentation because they allow both power and (analog) data to be conveyed on the same pair of wires.

HART communication extends this efficiency one more step: in addition to DC power and analog signal current being carried by a common wire pair, *digital* data may also be carried to and from the loop-powered transmitter in the form of electrical pulses superimposed on the cable.

With the incorporation of digital data on the same wire pair as the DC power and analog signal, a wide range of new possibilities suddenly emerge. Now, the field-mounted transmitter has the ability to communicate self-diagnostic information, status reports, alarms, and even multiple process variables in addition to the original analog signal representing the (main) process variable. Information may even be communicated *to* the transmitter using the same digital protocol, using this digital data channel to set measurement ranges, activate special features (e.g., square-root characterization, damping, etc.), from a remote location.

With HART, the traditional series-connected circuit configuration of transmitter, DC power supply, and resistor remains unchanged. A HART-enabled transmitter, though, is equipped with a built-in digital microcontroller managing its functions, and this miniature computer is able to send and receive digital data as AC signals (current pulses in sending mode, voltage pulses in receiving mode) superimposed on the same two wires carrying the 4 to 20 mA analog signal and DC power. Any other computer device equipped with a HART modem, the necessary configuration software, and an appropriate device description[57] for that particular instrument may communicate with the HART transmitter if connected in parallel with the transmitter's loop power terminals:

Figure 4.61: A typical analog 4-20 mA signal loop with HART signal overlay

This external computer – through the use of HART data transmission – now has the ability to monitor details of the transmitter's operation, configure the transmitter, update its measurement ranges, and perform a host of additional functions.

The connection between the HART modem and the HART-enabled transmitter need not be made directly at the transmitter's terminals. *Any* set of points in the circuit electrically parallel to the transmitter's terminals are permissible as connection points for the HART modem. This flexibility is a great advantage in loop circuits spanning long distances, enabling the technician to

[57] A *Device Description*, or *DD* file, is analogous to a "driver" file used to instruct a personal computer how to communicate with a printer, scanner, or any other complex peripheral device. In this case, the file instructs the HART configuration computer on how it should access parameters inside the field instrument's microcontroller. Without an appropriate DD file loaded on the configuration computer, many of the field instrument's parameters may be inaccessible.

connect their HART configuration equipment at the most physically convenient location:

Figure 4.62: A HART communicator can connect in parallel anywhere in the loop

A convenient alternative to a personal computer and HART modem is a special hand-held device called a *HART communicator*. Two different models of HART communicator are shown in the following photographs, a Rosemount model 268 on the left and an Emerson model 375 on the right:

(a) Rosemount 268 (b) Rosemount 375

Figure 4.63: Examples of HART communicators

HART communicators are battery-powered, portable devices built specifically for configuring HART-enabled field instruments. Like personal computers, they need to be updated with DD files to be able to communicate with the latest models of HART-enabled field instruments.

It should be mentioned that even though HART is a legacy technology, it is still a very popular form of digital field instrument communication in industrial use. However, more modern digital standards such as Profibus and FOUNDATION Fieldbus exist capable of delivering all the benefits of HART technology and more. It seems that HART will remain in wide industrial use for many years to come, but it is really just the beginning of digital field instrument technology and by no means represents the state of the art.

### 4.10.2 HART physical layer

The HART standard was developed with existing installations in mind. The medium for digital communication had to be robust enough to travel over twisted-pair cables of very long length and unknown characteristic impedance. This meant that the data communication rate for the digital data had to be very slow, even by 1980's standards. The HART standard is concerned only with layers 1 (FSK modulation, ± 0.5 mA signaling), 2 (Master-slave arbitration, data frame organization), and 7 (specific commands to read and write device data) of the OSI Reference model. Layers 3 through 6 are irrelevant to the HART standard.

Digital data is encoded in HART using the Bell 202 modem standard: two audio-frequency "tones" (1200 Hz and 2200 Hz) are used to represent the binary states of "1" and "0," respectively,

transmitted at a rate of 1200 bits per second. This is known as *frequency-shift keying*, or *FSK*. The physical representation of these two frequencies is an AC current of 1 mA peak-to-peak superimposed on the 4-20 mA DC signal. Thus, when a HART-compatible device "talks" digitally on a two-wire loop circuit, it produces tone bursts of AC current at 1.2 kHz and 2.2kHz. The receiving HART device "listens" for these AC current frequencies and interprets them as binary bits.

An important consideration in HART current loops is that the total loop resistance (precision resistor values plus wire resistance) must fall within a certain range: 250 ohms to 1100 ohms. Most 4-20 mA loops (containing a single 250 ohm resistor for converting 4-20 mA to 1-5 V) measure in at just over 250 ohms total resistance, and work quite well with HART. Even loops containing two 250 ohm precision resistors meet this requirement. Where technicians often encounter problems is when they set up a loop-powered HART transmitter on the test bench with a lab-style power supply and *no* 250 ohm resistor anywhere in the circuit:

Figure 4.64: A HART loop without resistance of at least 250 ohms won't work

The HART transmitter may be modeled as two parallel current sources: one DC and one AC. The DC current source provides the 4-20 mA regulation necessary to represent the process measurement as an analog current value. The AC current source turns on and off as necessary to "inject" the 1 mA P-P audio-frequency HART signal along the two wires. Inside the transmitter is also a HART modem for interpreting AC voltage tones as HART data packets. Thus, data transmission takes place through the AC current source, and data reception takes place through a voltage-sensitive modem, all inside the transmitter, all "talking" along the same two wires that carry the DC 4-20 mA signal.

For ease of connection in the field, HART devices are designed to be connected in parallel with each other. This eliminates the need to break the loop and interrupt the DC current signal to connect a HART communicator device to communicate with the transmitter. A typical HART communicator may be modeled as an AC voltage source[58] (along with another HART voltage-sensitive modem for receiving HART data). Connected in parallel with the HART transmitter, the complete circuit looks something like this:

[58]The HART standard specifies "master" devices in a HART network transmit AC voltage signals, while "slave" devices transmit AC current signals.

Figure 4.65: A signal model of a HART loop

With all these sources in the same circuit, it is advisable to use the *Superposition Theorem* for analysis. This involves "turning off" all but one source at a time to see what the effect is for each source, then superimposing the results to see what all the sources do when all are working simultaneously.

Only the effects of either AC source need to be considered to see what the problem is in this circuit with no loop resistance. Consider the situation where the transmitter is sending HART data to the communicator. The AC current source inside the transmitter will be active, injecting its 1 mA P-P audio-frequency signal onto the two wires of the circuit. The AC voltage source in the communicator will disconnect itself from the network, allowing the communicator to "listen" to the transmitter's data.

To apply the Superposition Theorem, replace all the other sources with their own equivalent internal resistances (voltage sources become "shorts," and current sources become "opens"). The HART communicator will be modeled as an "open" even though it is technically a voltage source because it must turn itself off (i.e., switch to high-impedance mode) in order for any field device to communicate to it:

Figure 4.66: No HART communications without loop resistance

The HART communicator is "listening" for those audio tone signals sent by the transmitter's AC source, but it "hears" nothing because the DC power supply's equivalent short-circuit prevents any significant AC voltage from developing across the two wires. This is what happens when there is no loop resistance: no HART device is able to receive data sent by any other HART device.

The solution to this dilemma is to install a resistance of at least 250 ohms but not greater than 1100 ohms between the DC power source and all other HART devices, like this:

Figure 4.67: HART communications occurs with proper loop resistance

Loop resistance must be at least 250 ohms to allow the 1 mA P-P AC signal to develop enough voltage to be reliably detected by the HART modem in the listening device. The upper limit (1100 ohms) is not a function of HART communication so much as it is a function of the DC voltage drop, and the need to maintain a minimum DC terminal voltage at the transmitter for its own operation. If there is too much loop resistance, the transmitter will become "starved" of voltage and act erratically. In fact, even 1100 ohms of loop resistance may be too much if the DC power supply voltage is insufficient.

Loop resistance is also necessary for the HART transmitter to receive data signals transmitted by the HART communicator. When the circuit is analyzed when the HART communicator's voltage source is active, this results:

Figure 4.68: HART communications with loop resistance

Without the loop resistance in place, the DC power supply would "short out" the communicator's AC voltage signal just as effectively as it shorted out the transmitter's AC current signal. The presence of a loop resistor in the circuit prevents the DC power supply from "loading" the AC voltage signal by the communicator. This AC voltage is seen in the diagram as being directly in parallel with the transmitter, where its internal HART modem receives the audio tones and processes the data packets.

Manufacturers' instructions generally recommend HART communicator devices be connected directly in parallel with the HART field instrument, as shown in the previous schematic diagrams. However, it is also perfectly valid to connect the communicator device directly in parallel with the loop resistor like this:

Figure 4.69: HART communications in parallel with loop resistance

Connected directly in parallel with the loop resistor, the communicator is able to receive transmissions from the HART transmitter just fine, as the DC power source acts as a dead short to the AC current HART signal and passes it through to the transmitter.

This is nice to know, as it is often easier to achieve an alligator-clip connection across the leads of a resistor than it is to clip in parallel with the loop wires when at a terminal strip or at the controller end of the loop circuit.

HART technology has given a new lease on the venerable 4-20 mA analog instrumentation signal standard. It has allowed new features and capabilities to be added on to existing analog signal loops without having to upgrade wiring or change all instruments in the loop. Some of the features of HART are listed here:

- Diagnostic data may be transmitted by the field device (self-test results, out-of-limit alarms, preventative maintenance alerts, etc.)
- Field instruments may be re-ranged remotely through the use of HART communicators
- Technicians may use HART communicators to force field instruments into different "manual" modes for diagnostic purposes (e.g., forcing a transmitter to output a fixed current so as to check calibration of other loop components, manually stroking a valve equipped with a HART-capable positioner)
- Field instruments may be programmed with identification data (e.g., tag numbers corresponding to plant-wide instrument loop documentation)

### 4.10.3 HART multidrop mode

The HART standard also supports a mode of operation that is totally digital, and capable of supporting multiple HART instruments on the same pair of wires. This is known as *multidrop mode*.

Every HART instrument has an *address* number, which is typically set to a value of zero (0). A network address is a number used to distinguish one device from another on a broadcast network, so messages broadcast across the network may be directed to specific destinations. When a HART instrument operates in digital/analog hybrid mode, where it must have its own dedicated wire pair for communicating the 4-20 mA DC signal between it and an indicator or controller, there is no need for a digital address. An address becomes necessary only when multiple devices are connected to the same network wiring, and there arises a need to digitally distinguish one device from another on the same network.

This is a functionality the designers of HART intended from the beginning, although it is rarely used in industry. Multiple HART instruments may be connected directly in parallel with one another along the same wire pair, and information exchanged between those instruments and a host system, if the HART address numbers are set to non-zero values (between 1 and 15):

Figure 4.70: A multidrop HART communications network

Setting an instrument's HART address to a non-zero value is all that is necessary to engage multidrop mode. The address numbers themselves are irrelevant, as long as they fall within the range of 1 to 15 and are unique to that network.

The major disadvantage of using HART instruments in multidrop mode is its slow speed. Due to HART's slow data rate (1200 bits per second), it may take several seconds to access a particular instrument's data on a multidropped network. For some applications such as temperature measurement, this slow response time may be acceptable. For inherently faster processes such as liquid flow control, it would not be nearly fast enough to provide up-to-date information for the control system to act upon.

### 4.10.4 HART multi-variable transmitters

Some "smart" instruments have the ability to report multiple process variables. A good example of this is Coriolis-effect flowmeters, which by their very nature simultaneously measure the density, flow rate, and temperature of the fluid passing through them. A single pair of wires can only convey one 4-20 mA analog signal, but that same pair of wires may convey multiple digital signals encoded in the HART protocol. Digital signal transmission is required to realize the full capability of such "multi-variable" transmitters.

If the host system receiving the transmitter's signal(s) is HART-ready, it may digitally poll the transmitters for all variables. If, however, the host system does not "talk" using the HART protocol, some other means must be found to "decode" the wealth of digital data coming from the multi-variable transmitter. One such device is Rosemount's model 333 HART "Tri-Loop" demultiplexer shown in the following photograph:

Figure 4.71: HART loop demultiplexer produces multiple analog outputs

This device polls the multi-variable transmitter and converts up to three HART variables into independent 4-20 mA analog output signals, which any suitable analog indicator or controller device may receive.

It should be noted that the same caveat applicable to multidrop HART systems (i.e., slow speed) applies to HART polling of multi-variable transmitters. HART is a relatively slow digital bus standard, and as such it should never be considered for applications demanding quick response. In applications where speed is not a concern, however, it is a very practical solution for acquiring multiple channels of data over a single pair of wires.

## 4.11 Modbus

Developed by the Modicon company (the original manufacturer of the *Programmable Logic Controller*, or *PLC*) in 1979 for use in its industrial control products, *Modbus* is a protocol designed

specifically for exchanging process data between industrial control devices. The Modbus standard does not specify any details of physical networking, and thus may be deployed on any number of physical networks such as TIA-232, TIA-485, Ethernet (the latter via TCP/IP), and a special token-passing network also developed by Modicon called *Modbus Plus*. Modbus itself is primarily concerned with details at layer 7 of the OSI Reference Model, the so-called *Application Layer*.

Modbus, especially when implemented over simple serial networks such as TIA-232 and TIA-485, is a rather primitive protocol. The seemingly arbitrary numerical codes used to issue commands and specify addresses is antiquated by modern standards. For better or for worse, though, a great many digital industrial devices "speak" Modbus, even if they are also capable of communicating via some other network protocol.

### 4.11.1 Modbus data frames

Modbus defines a set of commands for reading (receiving) and writing (transmitting) data between a master device and one or more slave devices connected to the network. Each of these commands is referenced by a numerical code, with addresses of the master and slave devices' internal registers (data sources and data destinations) specified along with the function code in the Modbus message frame.

Two different formats are specified in the Modbus standard: *ASCII* and *RTU*. The difference between these two modes is how addresses, function codes, data, and error-checking bits are represented. In Modbus ASCII mode, all slave device addresses, function codes, and data are represented in the form of ASCII characters (7 bits each), which may be read directly by any terminal program (e.g., `minicom`, `Hyperterminal`, `kermit`, etc.) intercepting the serial data stream. This makes troubleshooting easier: to be able to directly view the Modbus data frames in human-readable form. In Modbus RTU mode, all slave device addresses, function codes, and data are represented directly in hexadecimal form, with four bits per hexadecimal character. Different error-checking techniques are used for ASCII and RTU modes as well. The following diagram compares data frames for the two Modbus communication modes:

Figure 4.72: Modbus data frames

As can be seen from a comparison of the two frames, ASCII frames require nearly twice the

number of bits as RTU frames, making Modbus ASCII slower than Modbus RTU for any given data rate (bits per second).

The contents of the "Data" field vary greatly depending on which function is invoked, and whether or not the frame is issued by the master device or from a slave device. More details on Modbus "Data" field contents will appear in a later subsection.

Since Modbus is strictly a "layer 7" protocol, these message frames are usually embedded within other data frames specified by lower-level protocols. For example, Modbus over TCP/IP encapsulates individual Modbus data frames as TCP/IP packets, which are then (usually) encapsulated again as Ethernet packets to arrive at the destination device. This "multi-layered" approach inherent to Modbus being such a high-level protocol may seem cumbersome, but it offers great flexibility in that Modbus frames may be communicated over nearly any kind of virtual and physical network type.

### 4.11.2 Modbus function codes and addresses

A listing of commonly-used Modbus function codes appears in the following table:

| **Modbus code** (decimal) | **Function** |
|---|---|
| 01 | Read one or more PLC output "coils" (1 bit each) |
| 02 | Read one or more PLC input "contacts" (1 bit each) |
| 03 | Read one or more PLC "holding" registers (16 bits each) |
| 04 | Read one or more PLC analog input registers (16 bits each) |
| 05 | Write (force) a single PLC output "coil" (1 bit) |
| 06 | Write (preset) a single PLC "holding" register (16 bits) |
| 15 | Write (force) multiple PLC output "coils" (1 bit each) |
| 16 | Write (preset) multiple PLC "holding" registers (16 bits each) |

Modbus "984" addressing defines sets of fixed numerical ranges where various types of data may be found in a PLC or other control device. The absolute address ranges (according to the Modbus 984 scheme) are shown in this table:

| **Modbus codes** (decimal) | **Address range** (decimal) | **Purpose** |
|---|---|---|
| 01, 05, 15 | 00001 to 09999 | Discrete outputs ("coils") |
| 02 | 10001 to 19999 | Discrete inputs ("contacts") |
| 04 | 30001 to 39999 | Analog input registers |
| 03, 06, 16 | 40001 to 49999 | "Holding" registers |

This fixed addressing scheme usually does not match conveniently to the addressing within the master or slave devices. Manufacturer's documentation for Modbus-compatible devices normally provide Modbus "mapping" references so technicians and engineers alike may determine which Modbus addresses refer to specific bit or word registers in the device.

Note how all the Modbus address ranges begin at the number one, not zero as is customary for so many digital systems. For example, a PLC with sixteen discrete input channels numbered 0 through

15 by the manufacturer may "map" those inputs ("contacts") to Modbus addresses 10001 through 10016, respectively.

Coil, Contact, and Register addresses are specified within Modbus data frames relative to the starting points of their respective commands. For example, the function to read discrete inputs ("contacts") on a slave device only applies to Modbus absolute addresses 10001 through 19999, and so the actual Modbus command issued to read contact address 10005 will specify the address as 0004 (since 10005 is the *fourth* address up from the start of the contact address block 10001 through 19999), and this *relative* address value is always communicated as a hexadecimal (not decimal) value[59].

### 4.11.3 Modbus function command formats

Every Modbus data frame, whether ASCII or RTU mode, has a field designated for "data." For each Modbus function, the content of this "data" field follows a specific format. It is the purpose of this subsection to document the data formats required for common Modbus functions, both the "Query" message transmitted by the Modbus master device to a slave device, and the corresponding "Response" message transmitted back to the master device by the queried slave device.

Since each Modbus data frame is packaged in multiples of 8 bits (RTU), they are usually represented in text as individual bytes (two hexadecimal characters). For example, a 16-bit "word" of Modbus data such as `1100100101011011` would typically be documented as `C9 5B` with a deliberate space separating the "high" (`C9`) and "low" (`5B`) bytes.

#### Function code 01 – Read Coil(s)

This Modbus function reads the statuses of slave device discrete outputs ("coils") within the slave device, returning those statuses in blocks of eight (even if the "number of coils" specified in the query is not a multiple of eight!). Relevant Modbus addresses for this function range from 00001 to 09999 (decimal) but the starting address is a hexadecimal number representing the $(n-1)^{th}$ register from the beginning of this range (e.g., decimal address 00100 would be specified as hexadecimal `00 63`).

Query message (Function code 01)

| Start | Slave address | Function code | Data | | | | Error check | End |
|---|---|---|---|---|---|---|---|---|
| | | | Starting address | | Number of coils | | | |
| | XX | 01 | Hi | Lo | Hi | Lo | XX | |

*Start* *Stop*

Figure 4.73: Read Coils query message structure

[59] The process of converting a device's I/O addresses to Modbus commands is about as much fun as being flogged with a rubber hose. However, some digital device manufacturers include Modbus address "translation" features into their products, so the programmer isn't burdened with offset calculations and decimal-to-hexadecimal conversions just to move a few blocks of data between devices on a Modbus network.

Figure 4.74: Read Coils response message structure

Note that the second and third bytes representing coil status are shown in grey, because their existence assumes more than one byte worth of coils has been requested in the query.

**Function code 02 – Read Contact(s)**

This Modbus function reads the statuses of slave device discrete inputs ("contacts") within the slave device, returning those statuses in blocks of eight (even if the "number of contacts" specified in the query is not a multiple of eight!). Relevant Modbus addresses for this function range from 10001 to 19999 (decimal), but the starting address is a hexadecimal number representing the $(n-1)^{th}$ register from the beginning of this range (e.g., decimal address 10256 would be specified as hexadecimal `00 FF`).

Figure 4.75: Read Contacts query message structure

Figure 4.76: Read Contacts response message structure

**Function code 03 – Read Holding Register(s)**

This Modbus function reads the statuses of "holding" registers within the slave device, with the size of each register assumed to be two bytes (16 bits). Relevant Modbus addresses for this function range from 40001 to 49999 (decimal), but the starting address is a hexadecimal number representing the $(n-1)^{th}$ register from the beginning of this range (e.g., decimal address 40980 would be specified as hexadecimal `03 D3`).

Figure 4.77: Read Holding Registers query message structure

Figure 4.78: Read Holding Registers response message structure

Note that since the query message specifies the number of registers (each register being two bytes in size), and the response message replies with the number of *bytes*, the response message's "number of bytes" field will have a value twice that of the query message's "number of registers" field. Note also that the maximum number of registers which may be requested in the query message (65536) with "high" and "low" byte values grossly exceeds the number of bytes the response message can report (255) with its single byte value.

**Function code 04 – Read Analog Input Register(s)**

This Modbus function is virtually identical to 03 (Read Holding Registers) except that it reads "input" registers instead: addresses 30001 through 39999 (decimal). As with all the Modbus relative addresses, the starting address specified in both messages is a hexadecimal number representing the $(n-1)^{th}$ register from the beginning of this range (e.g., decimal address 32893 would be specified as hexadecimal `0B 4C`).

Figure 4.79: Read Analog Input Registers query message structure

Figure 4.80: Read Analog Input Registers response message structure

Note that since the query message specifies the number of registers (each register being two bytes in size), and the response message replies with the number of *bytes*, the response message's "number of bytes" field will have a value twice that of the query message's "number of registers" field. Note also that the maximum number of registers which may be requested in the query message (65536) with "high" and "low" byte values grossly exceeds the number of bytes the response message can report (255) with its single byte value.

**Function code 05 – Write (Force) Single Coil**

This Modbus function writes a single bit of data to a discrete output ("coil") within the slave device. Relevant Modbus addresses for this function range from 00001 to 09999 (decimal) but the starting address is a hexadecimal number representing the $(n-1)^{th}$ register from the beginning of this range (e.g., decimal address 07200 would be specified as hexadecimal `1C 1F`).

Figure 4.81: Write Single Coil query and response message structure

The "force data" for a single coil consists of either `00 00` (force coil off) or `FF 00` (force coil on). No other data values will suffice – anything other than `00 00` or `FF 00` will be ignored by the slave device.

A normal response message will be a simple echo (verbatim repeat) of the query message.

**Function code 06 – Write (Preset) Single Holding Register**

This Modbus function writes data to a single "holding" register within the slave device. Relevant Modbus addresses for this function range from 40001 to 49999 (decimal) but the starting address is a hexadecimal number representing the $(n-1)^{th}$ register from the beginning of this range (e.g., decimal address 40034 would be specified as hexadecimal `00 21`).

Figure 4.82: Write Single Holding Register query and response message structure

A normal response message will be a simple echo (verbatim repeat) of the query message.

**Function code 15 – Write (Force) Multiple Coils**

This Modbus function writes multiple bits of data to a set of discrete outputs ("coils") within the slave device. Relevant Modbus addresses for this function range from 00001 to 09999 (decimal) but the starting address is a hexadecimal number representing the $(n-1)^{th}$ register from the beginning of this range (e.g., decimal address 03207 would be specified as hexadecimal `0C 86`).

Figure 4.83: Write Multiple Coils query message structure

Response message (Function code 15)

| Start | Slave address | Function code | Data | | | | Error check | End |
|---|---|---|---|---|---|---|---|---|
| | | | Starting address | | Number of coils | | | |
| | XX | 0F | Hi | Lo | Hi | Lo | XX | |

Start Stop

Figure 4.84: Write Multiple Coils response message structure

Note that the query message specifies both the number of coils (bits) and the number of bytes.

**Function code 16 – Write (Preset) Multiple Holding Register**

This Modbus function writes multiple words of data to a set of "holding" registers within the slave device. Relevant Modbus addresses for this function range from 40001 to 49999 (decimal) but the starting address is a hexadecimal number representing the $(n-1)^{th}$ register from the beginning of this range (e.g., decimal address 47441 would be specified as hexadecimal `1D 10`).

Query message (Function code 16)

| Start | Slave address | Function code | Data | | | | | | | | | Error check | End |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Starting address | | Number of registers | | Number of bytes | Preset data first register | | Preset data second register | | | |
| | XX | 10 | Hi | Lo | Hi | Lo | | Hi | Lo | Hi | Lo | XX | |

Start Stop

Figure 4.85: Write Multiple Holding Registers query message structure

Response message (Function code 16)

| Start | Slave address | Function code | Data | | | | Error check | End |
|---|---|---|---|---|---|---|---|---|
| | | | Starting address | | Number of registers | | | |
| | XX | 10 | Hi | Lo | Hi | Lo | XX | |

Start Stop

Figure 4.86: Write Multiple Holding Registers response message structure

Note that the query message specifies both the number of registers (16-bit words) and the number of bytes, which is redundant (the number of bytes must *always* be twice the number of registers, given that each register is two bytes[60] in size). Note also that the maximum number of registers

[60] Even for devices where the register size is less than two bytes (e.g., Modicon M84 and 484 model controllers have

which may be requested in the query message (65536) with "high" and "low" byte values grossly exceeds the number of bytes the response message can report (255) with its single byte value.

10 bits within each register), data is still addressed as two bytes' worth per register, with the leading bits simply set to zero to act as placeholders.

# References


[1] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1.

[2] 422 and 485 Standards Overview and System Configurations, Application Report SLLA070C, Texas Instruments, Dallas, TX, 2002.

[3] B&B Converters for the Industrial Bus World, Technical Article 13, B&B Electronics Manufacturing Company, Ottawa, IL, 2000.

[4] T. L. Floyd, Digital Fundamentals, 6th ed. Upper Saddle River, NJ: Prentice-Hall, Inc, 1997.

[5] FOUNDATION Fieldbus System Engineering Guidelines, AG 181 Revision 2.0, The Fieldbus Foundation, Austin, TX, 2004.

[6] FOUNDATION Specification System Architecture, FF 581 Revision FS 1.1, The Fieldbus Foundation, Austin, TX, 2000.

[7] Fundamentals of RS-232 Serial Communications, Application Note 83 (AN83), Maxim Integrated Products, Sunnyvale, CA, 2001.

[8] D. C. Giancoli, Physics for Scientists & Engineers, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, Inc, 2000.

[9] F. D. Graham, Audel's New Electric Library, Volume IX. New York, NY: Theo. Audel & Co., 1942.

[10] HART Communications, Technical Information L452 EN, Samson AG, Frankfurt am Main, DE, 1999.

[11] C. Hecker, "Let's get to the (floating) point," Game Developer, vol. 20, pp. 19–24, feb/mar 1996.

[12] R. Horak, Telecommunications and Data Communications Handbook. New York, NY: John Wiley & Sons, 2007.

[13] ——, Webster's New World Telecom Dictionary. Indianapolis, IN: John Wiley & Sons, 2008.

# Chapter 5

# Introduction to Measurement

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: measurand, error, true value, conventional true value, uncertainty, systematic error, random error, range, span, accuracy, precision, hysteresis, resolution, repeatability, linearity, zero (null), sensitivity, drift, thermal stability, dead band
2. Describe the function of the typical subsystems in a measurement system
3. Determine and apply dynamic models for measurement devices
4. Explain how NUSAP notation is used to report a measurement

## 5.1 Measurement

Measurement is the set of operations used to find the value of a parameter. It is necessary to measure parameters for the purposes of trade, health and safety, environmental monitoring, law enforcement, and scientific research, among others. The science of measurement is known as *metrology*, which encompasses both theory and practice. Measurement is often a decision-making tool. For example, the totalized measurement of electrical energy to a house is used by the utility to decide how much to bill the owner. The results of laboratory exams are used by doctors to prescribe courses of treatment. The measurement of a vehicle's speed determines whether or not a police officer will issue a speeding ticket.

An enormous number of parameters, or *measurands*, are measured every day. *Error* is defined as the difference between the measurement result and the true value of a measurand. However, there is no such thing as a true value. In order to determine the true value of a measurand, the measurement process would have to be perfectly ideal. All observations would always result in the same value. Thus, true value is a philosophical concept, and error is defined with respect to a *conventional true value* (truth based on conventional agreement). The conventional true value is usually some agreed-upon value of a defined quantity that can be used as a reference. *Measurement standards* are materials or measurement systems used as conventional true values. In the United States, the National Institute of Standards and Technology (NIST) maintains standards so that measurements in different parts of the country can be compared.

## 5.2 The measurement process

Measurement involves the conveyance of information from the process to a recipient. Measurement typically involves three main functions or subsystems: sensing, conditioning, and reception. *Sensing* uses a transducer to convert the energy of the measurand into another type of energy (usually an electrical signal). When the relationship between the measurand and the signal variable are known and one-to-one, a mapping can be created between the values. *Conditioning* converts the signal obtained from the sensor into a form useful for the receiver, and may involve operations such as amplification or attenuation, filtering, mathematical operations (e.g., linearization or compensation), analog-to-digital conversion, or voltage-to-current conversion (e.g., for use in 4-20 mA signal transmission). *Reception* is the process of receiving the signal from the conditioning stage and using the data for indicating, recording, or controlling a process.

## 5.3 Measurement error

All measurements have error associated with them because there is no such thing as a perfect measurement system. Two types of error are typically defined: systematic error and random error.

### 5.3.1 Systematic error

*Systematic error* is defined as the mean of a number of repeated measurements of the same measurand made under the same conditions minus the true value. It is a repeatable error - that is, it appears in repeated measurements in the same amount. For example, if multiple temperature readings are taken and compared to a temperature reference and all measurements are 3 degF lower

than the reference indicates, a systematic error is present. If the systematic error is known, it is called a *bias* error. Systematic error is determined by taking repeated measurements, determining the arithmentic mean, and subtracting the conventional true value from the mean. Note that since conventional true value is used and only a finite number of measurements can be taken, systematic error can only be estimated. If the cause of systematic error can be identified in a measurement, it can typically be removed by repair or calibration.

### 5.3.2 Random error

*Random error* is defined as the result of a measurement minus the mean of a number of repeated measurements of the same measurand made under the same conditions. It is error that results in a non-repeatable way. It is typically due to uncontrollable or unknown influences. Random errors may occur with a certain probability distribution function, but do not always. Calibration cannot remove random error, it can only quantify it. Random error is typically reported as uncertainty.

## 5.4 Instrument Characteristics

Instrumentation manufacturer's publish information on the characteristics of their products so that engineers can properly select and apply them. The *range* is the region in which a quantity is measured, received, or transmitted. The range is stated using the lower range limit (LRL) and upper range limit (URL), e.g., 20 to 100 psig, and is typically assumed to be defined by a linear characteristic known as the calibration curve. The *span* is the algebraic difference between the LRL and URL. In the example above, the span is 80 psig. *Linearity* is the closeness to which the calibration curve of the device approaches a straight line function. It is usually determined by the maximum deviation of the the calibration curve data from the straight line that minimizes the maximum deviation of the data as determined by a least-squares fit.

*Hysteresis* refers to the property of the characteristic curve where the current value of the output depends on both the history of the output and the direction of the input. In order to determine hysteresis, the maximum value of the separation is determined from measurements going both upscale and downscale (with any dead band subtracted).

*Accuracy* defines the limit of the uncertainty of an instrument when it is used as specified by the manufacturer, and typically includes hysteresis, dead band, and repeatability errors. The term accuracy is misleading, since it really refers to *inaccuracy*. Unfortunately, there is no agreed-upon standard for manufacturers to report accuracy. Accuracy is often expressed in one of several ways as indicated below:

1. In terms of the measurement units, e.g., ±4 psig.
2. As a percent of span, e.g., ±10% of span.
3. As a percent of the reading, e.g., ±10% of reading.

It is important that the accuracy of the device be known for the range of the measurand where the measurements will take place. Table 5.1 compares the uncertainty in a pressure instrument as determined by the different methods described above using the example exaggerated values. Note that the 10% of span accuracy is always a constant value, while the 10% of reading accuracy is small at the smaller readings and large at the larger readings. This means that a device with an accuracy given in percent of reading may be more accurate than a device with an accuracy given in percent of span if the nominal value of the measurand is at the low end of the scale. This is illustrated in

Figure 5.1. The solid line is the pressure reading, the dotted line is the uncertainty bounds in terms of the measurement units, the dashed line is the uncertainty bounds as a percent of span, and the dash-dot line is the uncertainty bounds as a percent of reading.

Table 5.1: Accuracy as determined by various methods

| Pressure (psig) | Measurement Units (psig) | % of Span (psig) | % of Reading (psig) |
|---|---|---|---|
| 20 | ±4 | ±8 | ±2 |
| 30 | ±4 | ±8 | ±3 |
| 40 | ±4 | ±8 | ±4 |
| 50 | ±4 | ±8 | ±5 |
| 60 | ±4 | ±8 | ±6 |
| 70 | ±4 | ±8 | ±7 |
| 80 | ±4 | ±8 | ±8 |
| 90 | ±4 | ±8 | ±9 |
| 100 | ±4 | ±8 | ±10 |

Figure 5.1: Accuracy limites based on measurement units, percentage of span, and percentage of reading

## 5.5 Instrument dynamics

Like all systems, measurement systems have a dynamic response associated with them. Often, this response is governed by a first-order or second-order ordinary differential equation, and typically the sensing element dynamics dominate the response as sensor dynamics are usually slow compared to the dynamics of conditioners or receivers. Additionally, protection elements used with delicate sensors can add dominant dynamics. For example, a bare thermocouple (temperature sensor) has a first-order response. When that thermocouple is covered by a protective sheath or installed in a thermowell, a second-order overdamped response results that is significantly slower than that of the bare sensor because two first-order thermal lags are connected in series. However, since the lag due to the thermowell is usually significantly longer than that of the bare thermocouple element, the response may appear as a dominant first-order lag.

## 5.6 Reporting measurements

Funtowicz and Ravetz proposed a method for reporting measurements known as the NUSAP notation. This method is a way of standardizing the reporting. NUSAP is an abbreviation for the words:

- Numeral - the quantitative value of the measurand
- Unit - the standard unit of the measurand
- Spread - the uncertainty in the value of the measurand
- Assessment - a statement describing the spread characteristics, e.g., a confidence interval
- Pedigree - a statement regarding the history of the particular measurement that bolsters confidence in the value of the measurand

# References


[1] J. L. Bucher, The Metrology Handbook. Milwaukee, WI: ASQ Quality Press, 2004.

[2] P. F. Dunn, Measurement and Data Analysis for Engineering and Science. New York, NY: McGraw-Hill, 2005.

[3] R. S. Figliola and D. E. Beasley, Theory and Design for Mechanical Measurements, 4th ed. New York, NY: Wiley, 2006.

[4] J. P. Holman, Experimental Methods for Engineers, 7th ed. New York, NY: McGraw-Hill, 2001.

[5] A. J. Wheeler and A. R. Ganji, Introduction to Engineering Experimentation, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2010.

# Chapter 6

# Probability and Statistics for Measurement

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: population, sample, probability density function, statistic, probability, mean, dispersion, deviation, population mean, sample mean, population standard deviation, sample standard deviation, standard deviation of the mean, degrees of freedom, confidence level
2. Convert data from one representation to another
3. Identify appropriate representations for data to obtain linear representations
4. Describe and apply the following distribution functions: normal (Gaussian), Student's t, uniform

The world contains a certain amount of indeterminacy. *Probability* allows the prediction of results in many repetitive trials to a certain degree of accuracy, but uncertainty is always present. Any occurence that cannot be predicted perfectly *a priori* has some degree of randomness. Randomness falls within two classes: discrete random variables and continuous random variables. The basic difference is that discrete random variables may only take on discrete values while continuous random variables can assume any possible value between two limits. Only continuous random variables will be discussed here.

## 6.1 Estimates of true values

Assuming that any systematic errors have been compensated, repeated measurements will reveal the randomness in the measurement process. The true value of the measurand must then be estimated from the data at hand. For a given measurand $x$, the best possible estimate of the true value of $x$ will be given by $x = \mu \pm k\sigma$ where $\mu$ is the population mean, $\sigma$ is the population standard deviation, and $k$ is a coverage factor. Unfortunately, the entire population is usually not available, and therefore the estimate of $x$ must be made using sample data. Since different samples of a population result in different calculated sample means, the variation in the sample means must be estimated in order to report the uncertainty in the mean. This is accomplished by calculating the standard deviation of the sample mean, $S_x = S/\sqrt{n}$.Thus, the best possible estimate of the true value of $x$ using sample data will be given by $x = \bar{x} \pm kS_x$ where $\bar{x}$ is the sample mean, $S_x$ is the sample standard deviation of the mean, and $k$ is a coverage factor.

## 6.2 Probabilities and statistics of measurement data

## 6.3 Cumulative distribution function

A cumulative distribution function (CDF) of a continuous random variable describes the probability that the random variable takes on a value equal to or less than an independent variable in a chance experiment trial. Cumulative distribution functions are monotonic (nondecreasing) from 0 at the low end of the independent variable range to 1 at the high end of the independent variable range. Cumulative distribution functions will be referred to using a capital letter as a function of the independent variable, e.g., P(x).

## 6.4 Probability density functions

The probability density function (PDF) of a continuous random variable can be determined by taking the derivative of the cumulative distribution function with respect to the independent variable. Probability density functions will be referred to using a small letter as a function of the independent variable, e.g., p(x).

$$p(x) = \frac{dP(x)}{dx} \tag{6.1}$$

This means that the CDF can be recovered from the PDF by integration:

$$P(x) = \int_{-\infty}^{\infty} p(x)dx \tag{6.2}$$

Since the total probability of all events described by the cumulative distribution function is 1, the probability density function therefore has an area under its curve of 1:

$$\int_{-\infty}^{\infty} p(x)dx = 1 \tag{6.3}$$

### 6.4.1 Moments of a Random Variable

The nth moment of a continuous random variable ($m_n$) is defined as:

$$m_n = \int_{-\infty}^{\infty} x^n p(x)dx \tag{6.4}$$

#### First Moment - Expected Value

The first moment of a probability density function gives its expected value, or mean, $\mu$.

$$\mu = \int_{-\infty}^{\infty} xp(x)dx \tag{6.5}$$

#### Second Moment - Variance

The second central moment (found by replacing $x$ with $x - \mu$) of a probability density function gives its variance, $\sigma^2$.

$$\sigma^2 = \int_{-\infty}^{\infty} (x-\mu)^2 p(x)dx \tag{6.6}$$

The square root of the variance is the standard deviation, $\sigma$.

### 6.4.2 Important PDFs in measurement

#### Gaussian (normal) density function

The Gaussian distrubtion is defined by two parameters: the mean and the standard deviation:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where $\mu$ is the population mean and $\sigma$ is the population standard deviation. The mean describes the central tendency of the Gaussian function and the standard deviation describes the dispersion.

Figure 6.1 is an example of two Gaussian distributions with the same mean of 0 but different standard deviations. The total area under the curve is 1, thus, if $\sigma$ changes, so does the peak value of the function. The function traced out by integrating the Gaussian distribution curve is known as the cumulative distribution function.

Figure 6.1: Gaussian probability density functions with same mean and different standard deviations

A coverage factor $k$ may be used to determine the probability of the true value existing within some value of the mean. The value will be reported $x = \bar{x} \pm kS_x$ $(CI\%)$, where CI is the confidence level (in %) that the true value lies within the bounds of the expression. The following coverage factors are typically associated with Gaussian distributions:

| **Coverage Factor** | **Confidence Level (%)** |
|---|---|
| 1 | 68.3 |
| 1.96 | 95.0 |
| 2 | 95.5 |
| 3 | 99.7 |

These coverage factors for Gaussian distributed measurements may be determined using Microsoft Excel's `NORMSINV`:

$$k = \texttt{NORMSINV}((1 + CI)/2) \tag{6.7}$$

**Example 6.6** Using Microsoft Excel, find the coverage factor for a sample of 100 measurements

of a Gaussian distributed population that will establish a confidence interval of 95%.

**Answer to Example 6.6** Use the command "=`NORMSINV`((1+CI)/2)" with CI = 0.95 to determine the coverage factor is 1.96.

**Example 6.7** One hundred temperature measurements are taken. The set has a mean of 69.3 °C and a standard deviation of 0.15 °C. Determine the reported measurement value for a confidence interval of 95.5%.

**Answer to Example 6.7** Use the Excel command "=`NORMSINV`((1+CI)/2)" to determine the coverage factor, which is 2.00. The sample standard deviation of the mean is $S_x = S/\sqrt{n} = 0.15/\sqrt{100} = 0.015$. Putting these all together:

$$\begin{aligned} x &= \bar{x} \pm kS_x \\ &= 69.3\ ^\circ C \pm 2.00(0.015)\ ^\circ C\ (95.5\%) \\ &= 69.30\ ^\circ C \pm 0.03\ ^\circ C\ (95.5\%) \end{aligned}$$

### Rectangular (uniform) density function

The *rectangular* (also known as *uniform*) probability density function has its basis in the concept that all intervals of the same length on the x-axis have the same probability. The rectangular PDF is defined as:

$$p(x) = \begin{cases} \frac{1}{a_+ - a_-} & \text{for } a_- \leq x \leq a_+ \\ 0 & \text{for } x < a_- \text{ or } x > a_+ \end{cases}$$

**Example 6.8** Prove that the area under the curve of a rectangular PDF equals 1.

**Answer to Example 6.8** The area of a rectangle is simple length times height. The length is $l = a_+ - a_-$, and the height is $h = \frac{1}{a_+ - a_-}$. Therefore:

$$\begin{aligned} A &= lh \\ &= a_+ - a_- \times \frac{1}{a_+ - a_-} \\ &= 1 \end{aligned}$$

Figure 6.2 is an example of a rectangular PDF with a mean of 0 and upper and lower bounds of 2 and -2, respectively.

Figure 6.2: Example of a rectangular PDF

**Mean - Rectangular PDF** The rectangular PDF has a mean of $0.5(a_+ + a_-)$.

**Example 6.9** Prove that the mean of a rectangular PDF is $0.5(a_+ + a_-)$.

**Answer to Example 6.9** The mean is the first moment of the PDF. Therefore:

$$
\begin{aligned}
\mu &= \int_{-\infty}^{\infty} xp(x)dx \\
&= \int_{a_-}^{a_+} x\frac{1}{a_+ - a_-}dx \\
&= \frac{1}{a_+ - a_-}\int_{a_-}^{a_+} xdx \\
&= \frac{1}{a_+ - a_-}\left.\frac{x^2}{2}\right|_{a_-}^{a_+} \\
&= \frac{1}{2(a_+ - a_-)}\left(a_+^2 - a_-^2\right) \\
&= \frac{\left(a_+^2 - a_-^2\right)}{2\left(a_+ - a_-\right)} \\
&= \frac{\left(a_+ - a_-\right)\left(a_+ + a_-\right)}{2\left(a_+ - a_-\right)} \\
&= \frac{\left(a_+ + a_-\right)}{2} \\
&= 0.5\left(a_+ + a_-\right)
\end{aligned}
$$

For Figure 6.2, the mean is 0.

**Standard Deviation - Rectangular PDF** The rectangular PDF has a standard deviation of $a_+/\sqrt{3}$ when it has a mean of 0 (i.e., when the mean has been subtracted).

**Example 6.10** Prove that the standard deviation of a rectangular PDF is $a_+/\sqrt{3}$.

**Answer to Example 6.10** Solve for the variance first, then take the square root of it. Note that $x - \mu$ is simply $x$ since $\mu = 0$, and that $a_- = -a_+$. These facts are used in the derivation. The

variance is the second moment of the PDF. Therefore:

$$
\begin{aligned}
\sigma^2 &= \int_{-\infty}^{\infty} x^2 p(x) dx \\
&= \int_{a_-}^{a_+} x^2 \frac{1}{a_+ - a_-} dx \\
&= \frac{1}{a_+ - a_-} \int_{a_-}^{a_+} x^2 dx \\
&= \frac{1}{a_+ - a_-} \left. \frac{x^3}{3} \right|_{a_-}^{a_+} \\
&= \frac{1}{3(a_+ - a_-)} \left(a_+^3 - a_-^3\right) \\
&= \frac{\left(a_+^3 - a_-^3\right)}{3\left(a_+ - a_-\right)} \\
&= \frac{\left(a_+ - a_-\right)\left(a_+^2 + a_+ a_- + a_-^2\right)}{3\left(a_+ - a_-\right)} \\
&= \frac{\left(a_+ - a_-\right)\left(a_+^2 - a_+ a_+ + a_+^2\right)}{3\left(a_+ - a_-\right)} \\
&= \frac{a_+^2}{3} \\
\Rightarrow \sigma &= \sqrt{\frac{a_+^2}{3}} \\
&= \frac{a_+}{\sqrt{3}}
\end{aligned}
$$

For Figure 6.2, the standard deviation is $2/\sqrt{3}$.

### Student's t density function

When sample sizes are small (n≤30), the Gaussian distribution tends to overestimate the value at the mean and underestimate the value near the tails of the function. The Student's t distribution is therefore used if the data appears to have a Gaussian distribution but the sample size $n$ is less than or equal to 30 data points. Here is a comparison of the Gaussian and Student's t distributions:

Figure 6.3: Comparison of the Gaussian and Student's t distributions

The Student's t curve approaches the Gaussian curve as the sample size grows, and at or above 100 data points the two distributions are essentially identical. The Student's t density function is used to determine the appropriate coverage factor for small sample sizes.

In order to use the Student's t function to determine a coverage factor, two values are required: the degrees of freedom ($\nu$) and the confidence level desired. Given $n$ samples in a data set, these $n$ pieces of information can be used to estimate a parameter. For every estimate made, one degree of freedom is lost in any subsequent calculations that use that estimate. As an example, if the mean is calculated from 15 measurements, there are 14 ($n$-1) degrees of freedom remaining to calculate the sample standard deviation. This is because the sample standard deviation formula uses the previously calculated mean.

The Student's t value determined from a particular combination of $\nu$ and confidence interval is often represented $t_{\nu,CI}$ This value is then used as the coverage factor $k$ instead of the standard values associated with a Gaussian distribution any time the number of samples $n$ is small. In Microsoft Excel, the value of $t_{\nu,CI}$ may be calculated by:

$$t_{\nu,CI} = \texttt{TINV}(1 - CI,\ \nu) \tag{6.8}$$

**Example 6.11** Using Microsoft Excel, find the coverage factor for a sample of 20 measurements of a population that will establish a confidence interval of 95%.

**Answer to Example 6.11** First, find the number of degrees of freedom using $\nu = n$-1, where $n$ is the number of measurements. Therefore, $\nu = 19$. Next, find the value $1 - CI = 1 - 0.95 = 0.05$. Finally, use the command "=`TINV`(0.05, 19)" to determine the coverage factor is 2.093.

**Example 6.12** Ten temperature measurements are taken. The set has a mean of 69.3 °C and a standard deviation of 0.15 °C. Determine the reported measurement value for a confidence interval of 95.5%.

**Answer to Example 6.12** First, find the number of degrees of freedom using $\nu = n$-1, where $n$ is the number of measurements. Therefore, $\nu = 9$. Next, find the value $1 - CI = 1 - 0.95.5 = 0.045$. Use the Excel command "=`TINV`(0.045, 9)" to determine the coverage factor, which is 2.327. The sample standard deviation of the mean is $S_x = S/\sqrt{n} = 0.15/\sqrt{10} = 0.047$. Putting these all together:

$$\begin{aligned} x &= \bar{x} \pm t_{9,95.5} S_x \\ &= 69.3\ ^\circ C \pm 2.327(0.047)\ ^\circ C\ (95.5\%) \\ &= 69.30\ ^\circ C \pm 0.11\ ^\circ C\ (95.5\%) \end{aligned}$$

# References


[1] J. L. Bucher, Ed., The Metrology Handbook. Milwaukee, WI: ASQ Quality Press, 2004.

[2] S. K. Kimothi, Ed., The Uncertainty of Measurements. Milwaukee, WI: ASQ Quality Press, 2002.

[3] P. F. Dunn, Ed., Measurement and Data Analysis for Engineering and Science. New York, NY: McGraw-Hill, 2005.

[4] R. S. Figliola and D. E. Beasley, Eds., Theory and Design for Mechanical Measurements, 4th ed. New York, NY: Wiley, 2006.

[5] J. P. Holman, Ed., Experimental Methods for Engineers, 7th ed. New York, NY: McGraw-Hill, 2001.

[6] A. J. Wheeler and A. R. Ganji, Eds., Introduction to Engineering Experimentation, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[7] C. W. Helstrom, Ed., Probability and Stochastic Processes for Engineers. New York, NY: Macmillan, 1984.

# Chapter 7

# Uncertainty in Measurement

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: relative uncertainty, standard uncertainty, combined standard uncertainty, expanded uncertainty, coverage factor, Type A uncertainty, Type B uncertainty, root sum of squares
2. Evaluate the uncertainty in a measurement in accordance with approved standards
3. Report a measurement using the NUSAP method

## 7.1 Measurement Reporting

Typically, reporting uses either the first three or four categories of the NUSAP method. For example, a measured value $x$ is often reported:

$$x = \bar{x} \pm u_x \; y \; (z\%) \tag{7.1}$$

where $\bar{x}$ is the best estimate of $x$ (number and units), $u_x$ is the estimated expanded uncertainty in the measurement (spread), $y$ is the units, and $z$ is the confidence interval (assessment). However, manufacturers will frequently omit the confidence interval when describing the uncertainty of an instrument on their data sheets. Procedures for dealing with this follow later in this chapter.

### 7.1.1 Relative uncertainty

The relative uncertainty (also known as fractional uncertainty) is the ratio of the estimated uncertainty to the best estimate of the measurement:

$$\delta x = \frac{u_x}{|\bar{x}|} \tag{7.2}$$

### 7.1.2 Significant figures

The least significant figure in any measurement should be of the same order of magnitude, i.e., have the same decimal position, as the uncertainty.

An experimental uncertainty should typically be rounded to one significant figure unless the leading digit in the uncertainty is 2 or smaller, in which case the uncertainty should be rounded to two significant figures.

### 7.1.3 The normal distribution

Many measurands have a normal probability density function and therefore their uncertainty will be associated with their sample standard deviation $S$. For measurands that do not have a normal probability density function, e.g., a uniform density function, an equivalent standard uncertainty can be determined as described later.

#### Central tendency

The mean describes the central tendency of the probability density function. It is calculated by:

Since the mean is based on a sample rather than the population, there is uncertainty associated with its value. The standard deviation of the mean can be calculated by $S/\sqrt{n}$, where $S$ is the sample standard deviation and $n$ is the number of samples in the data.

#### Dispersion

The standard deviation (the square root of the variance) describes the dispersion of the data about the central tendency. A higher standard deviation means that the peak of the distribution is lower and the dispersion is wider.

**Reporting measurements with normal distributions**

The standard deviation of the mean is multiplied by a coverage factor $k$ based on values from the normal distribution where 30 or more samples have been taken. If fewer than 30 samples have been taken, use the Student's t distribution values to determine the coverage factor $k$.

The measurement is then reported:

$$x = \bar{x} + k\frac{S}{\sqrt{n}} \tag{7.3}$$

## 7.2 Classifications of Uncertainty

The current rules classify uncertainty into two types, Type A and Type B. Type A uncertainties are those which are estimated by statistical methods, and Type B uncertainties are those estimated by other means.

### 7.2.1 Type A Uncertainty

The estimation of uncertainty by statistical analysis of a series of measurements is a Type A analysis. The standard uncertainty in the Type A category is simply the standard deviation.

### 7.2.2 Type B Uncertainty

The estimation of uncertainty by other means than statistical analysis is a Type B analysis. These types of analyses include the use of manufacturer's specifications, calibration data, handbook data, etc. Frequently, Type A uncertainties determined by manufacturers or researchers are treated as Type B uncertainties since the originator of the uncertainty data does not provide any insight as to how the uncertainty data was obtained, e.g., the originator does not state that the uncertainty was determined from statistical methods.

There are multiple methods to determine the standard uncertainty of a Type B analysis based on the assumed probability distribution used to determine the uncertainty data.

- Convert a quoted uncertainty that is a stated multiple of an estimated standard deviation to a standard uncertainty by dividing the quoted uncertainty by the multiplier.
- Convert a quoted uncertainty that defines a "confidence interval" having a stated level of confidence, such as 95 or 99 percent, to a standard uncertainty by treating the quoted uncertainty as if a normal distribution had been used to calculate it (unless otherwise indicated) and dividing it by the appropriate factor for such a distribution. These factors are 1.960 and 2.576 for the two levels of confidence given.
- Model the quantity in question by a normal distribution and estimate lower and upper limits $a_-$ and $a_+$ such that the best estimated value of the quantity is $(a_+ + a_-)/2$ (i.e., the center of the limits) and there is 1 chance out of 2 (i.e., a 50 percent probability) that the value of the quantity lies in the interval $a_-$ to $a_+$. Then $u_j \approx 1.48a$, where $a = (a_+ - a_-)/2$ is the half-width of the interval.
- Model the quantity in question by a normal distribution and estimate lower and upper limits $a_-$ and $a_+$ such that the best estimated value of the quantity is $(a_+ + a_-)/2$ and there is about

a 2 out of 3 chance (i.e., a 67 percent probability) that the value of the quantity lies in the interval $a_-$ to $a_+$. Then $u_j \approx a$, where $a = (a_+ - a_-)/2$.

- Estimate lower and upper limits $a_-$ and $a_+$ for the value of the quantity in question such that the probability that the value lies in the interval $a_-$ to $a_+$ is, for all practical purposes, 100 percent. Provided that there is no contradictory information, treat the quantity as if it is equally probable for its value to lie anywhere within the interval $a_-$ to $a_+$; that is, model it by a uniform or rectangular probability distribution. The best estimate of the value of the quantity is then (a+ + a- )/2 with $u_j = a/\sqrt{3}$, where $a = (a_+ - a_-)/2$. If the distribution used to model the quantity is triangular rather than rectangular, then $u_j = a/\sqrt{6}$.

## 7.3 Combining Standard Uncertainties

Uncertainties may only be combined if they all have the same units and all have the same confidence level. If they do not have the same confidence level, their coverage factors will have to be adjusted so that they do, at which point they may be combined. This is why standard uncertainties are used when combining uncertainties; they all have a coverage factor of 1.

### 7.3.1 General Propagation of Uncertainty

The combined standard uncertainty in a measurement result $y$ that is a function of multiple inputs $x_i$, $y = f(x_i)$, is calculated by summation in quadrature (also known as taking the root sum of squares) of the weighted values of the individual standard uncertainties of each input:

$$u_c^2(y) = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i) + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \frac{\partial f}{\partial x_i}\frac{\partial f}{\partial x_j} u(x_i, x_j) \tag{7.4}$$

The first term combines independent uncertainties, the second term combines dependent uncertainties by making use of the covariance between the dependent variables. If all uncertainty terms are independent, the second term vanishes since all covariances are zero. This leaves:

$$u_c^2(y) = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i) \tag{7.5}$$

### 7.3.2 Uncertainty Propagation - Special Cases

#### Propagation in Addition and Subtraction

If two or more independent and random measurements are added or subtracted to obtain a total measurement $q$, the best estimate of the total uncertainty is to add their individual uncertainties in quadrature:

$$u_q = \sqrt{u_x^2 + u_y^2 + \cdots + u_z^2} \tag{7.6}$$

**Propagation in Multiplication and Division**

If two or more independent and random measurements are multiplied or divided to obtain a total measurement $q$, the best estimate of the total relative uncertainty is to add their individual relative uncertainties in quadrature:

$$\frac{u_q}{|q|} = \sqrt{\left(\frac{u_x}{|x|}\right)^2 + \left(\frac{u_y}{|y|}\right)^2 + \cdots + \left(\frac{u_z}{|z|}\right)^2} \tag{7.7}$$

**Propagation in Functions**

If a measurement $x$ is used to calculate $f(x)$, the resultant is:

$$f(x) = f(\bar{x}) \pm \left|\frac{df(x)}{dx}\right| u_x \tag{7.8}$$

## 7.4 Expanded Uncertainty

The combined standard uncertainty is the equivalent of $1\sigma$, with a confidence level of 68%. If a higher confidence level is desired, an expanded uncertainty can be determined by multiplying the combined standard uncertainty by a coverage factor. For example, if the probability distribution is assumed to be normal and the uncertainty in the combined standard uncertainty itself is negligible, typical coverage factors are approximately 2 (1.96) for 95% confidence, 2.6 (2.576) for 99% confidence and 3 for greater than 99.73% confidence.

## 7.5 ISO rules

The ISO "Guide to the Expression of Uncertainty in Measurement" (2008) sets out a fairly step-by-step approach to calculating uncertainty. This document, also known as the ISO GUM, forms the basis for the processes used by NIST in the United States. The procedure below is based on the ISO GUM, but has been modified to allow the NUSAP notation:

1. Write the mathematical model that describes the relationships between the measurand and any inputs
2. Find the estimates of the input quantities from measurement or manufacturer data
3. Determine the standard uncertainty for each estimated input quantity
4. Evaluate whether any correlation exists between input quantities and determine the correlation coefficients for correlated inputs
5. Calculate the best estimate of the measurand based on the input quantities (N and U)
6. Determine the combined standard uncertainty from the uncertainties in each of the input quantities using propagation of uncertainty methods
7. Calculate the expanded uncertainty (S) using a stated coverage factor (A)
8. Craft a statement regarding the history of the measurement device or processs (P)
9. Report the measurement using the NUSAP notation

This procedure can be quite daunting to someone new to evaluating uncertainty, however software (e.g., QMSYS GUM from Qualisyst) is available to assist with the process.

# References


[1] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1.

[2] S. K. Kimothi, Ed., The Uncertainty of Measurements. Milwaukee, WI: Quality Press, 2002.

[3] B. N. Taylor and C. E. Kuyatt, "Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results," 1297, NIST Physics Laboratory, Gaithersburg, MD, Tech. Rep., September 1994.

# Chapter 8

# Instrument Calibration

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: calibration, ranging, zero, span, zero error, span error, linearity error, as-found data, as-left data, primary standard, secondary standard, traceability, static calibration, deviation plot, dynamic calibration
2. Create calibration curves and deviation plots and determine the sensitivity of a sensor given calibration data
3. Identify calibration problems such as zero error, span error, linearity error from as-found calibration curves
4. Explain the difference between calibration and re-ranging

Every instrument has at least one *input* and one *output*. For a pressure sensor, the input would be some fluid pressure and the output would (most likely) be an electronic signal. For a loop indicator, the input would be a 4-20 mA current signal and the output would be a human-readable display. For a variable-speed motor drive, the input would be an electronic signal and the output would be electric power to the motor.

*Calibration* and *ranging* are two tasks associated with establishing an accurate correspondence between any instrument's input signal and its output signal.

## 8.1 Calibration versus ranging

To *calibrate* an instrument means to check and adjust (if necessary) its response so the output accurately corresponds to its input throughout a specified range. In order to do this, one must expose the instrument to an actual input stimulus of precisely known quantity. For a pressure gauge, indicator, or transmitter, this would mean subjecting the pressure instrument to known fluid pressures and comparing the instrument response against those known pressure quantities. One cannot perform a true calibration without comparing an instrument's response to known, physical stimuli.

To *range* an instrument means to set the lower and upper range values so it responds with the desired sensitivity to changes in input. For example, a pressure transmitter set to a range of 0 to 200 psig (0 psig = 4 mA output ; 200 psig = 20 mA output) could be re-ranged to respond on a scale of 0 to 150 psig (0 psig = 4 mA ; 150 psig = 20 mA).

In analog instruments, re-ranging could (usually) only be accomplished by re-calibration, since the same adjustments were used to achieve both purposes. In digital instruments, calibration and ranging are typically separate adjustments (i.e., it is possible to re-range a digital transmitter without having to perform a complete recalibration), so it is important to understand the difference.

## 8.2 Zero and span adjustments (analog transmitters)

The purpose of *calibration* is to ensure the input and output of an instrument correspond to one another predictably throughout the entire range of operation. This expectation may be expressed in the form of a graph, showing how the input and output of an instrument should relate:

Figure 8.1: A calibration curve showing the percentage output versus the percentage input for a linear transmitter

This graph shows how any given percentage of input should correspond to the same percentage of output, all the way from 0% to 100%.

Things become more complicated when the input and output axes are represented by units of measurement other than "percent." Take for instance a pressure *transmitter*, a device designed to sense a fluid pressure and output an electronic signal corresponding to that pressure. Here is a graph for a pressure transmitter with an input range of 0 to 100 pounds per square inch (psig) and an electronic output signal range of 4 to 20 milliamps (mA) electric current:

Figure 8.2: A calibration curve showing for a linear pressure transmitter with a 4-20 mA output

Although the graph is still linear, zero pressure does not equate to zero current. This is called a *live zero*, because the 0% point of measurement (0 psig fluid pressure) corresponds to a non-zero ("live") electronic signal. 0 psig pressure may be the LRV (Lower Range Value) of the transmitter's input, but the LRV of the transmitter's output is 4 mA, not 0 mA.

Any linear, mathematical function may be expressed in "slope-intercept" equation form:

$$y = mx + b$$

Where,

$y$ = Vertical position on graph

$x$ = Horizontal position on graph

$m$ = Slope of line

$b$ = Point of intersection between the line and the vertical ($y$) axis

This instrument's calibration is no different. Let $x$ represent the input pressure in units of psig and $y$ represent the output current in units of milliamps. Thus, an equation for this instrument is as follows:

$$y = 0.16x + 4$$

On the actual instrument (the pressure transmitter), there are two adjustments which let us match the instrument's behavior to the ideal equation. One adjustment is called the *zero* while the other is called the *span*. These two adjustments correspond exactly to the $b$ and $m$ terms of the linear function, respectively: the "zero" adjustment shifts the instrument's function vertically on the graph, while the "span" adjustment changes the slope of the function on the graph. By

adjusting both zero and span, the instrument may be set for any range of measurement within the manufacturer's limits.

The relation of the slope-intercept line equation to an instrument's zero and span adjustments reveals something about how those adjustments are actually achieved in any instrument. A "zero" adjustment is always achieved by *adding* or *subtracting* some quantity, just like the $y$-intercept term $b$ adds or subtracts to the product $mx$. A "span" adjustment is always achieved by *multiplying* or *dividing* some quantity, just like the slope $m$ forms a product with the input variable $x$.

Zero adjustments typically take one or more of the following forms in an instrument:

- Bias force (spring or mass force applied to a mechanism)
- Mechanical offset (adding or subtracting a certain amount of motion)
- Bias voltage (adding or subtracting a certain amount of potential)

Span adjustments typically take one of these forms:

- Fulcrum position for a lever (changing the force or motion multiplication)
- Amplifier gain (multiplying or dividing a voltage signal)
- Spring rate (changing the force per unit distance of stretch)

It should be noted that for most analog instruments, zero and span adjustments are *interactive*. That is, adjusting one has an effect on the other. Specifically, changes made to the span adjustment almost always alter the instrument's zero point[1]. An instrument with interactive zero and span adjustments requires much more effort to accurately calibrate, as one must switch back and forth between the lower- and upper-range points repeatedly to adjust for accuracy.

## 8.3 Damping adjustments

The vast majority of modern process transmitters (both analog and digital) come equipped with a feature known as *damping*. This feature is essentially a low-pass filter function placed in-line with the signal, reducing the amount of process "noise" reported by the transmitter.

Imagine a pressure transmitter sensing water pressure at the outlet of a large pump. The flow of water exiting a pump tends to be extremely turbulent, and any pressure-sensing device connected to the immediate discharge port of a pump will interpret this turbulence as violent fluctuations in pressure. This means the pressure signal output by the transmitter will fluctuate as well, causing any indicator or control system connected to that transmitter to register a very "noisy" water pressure:

[1] It is actually quite rare to find an instrument where a change to the zero adjustment affects the instrument's span.

Figure 8.3: A noisy pressure signal

Such "noise" wreaks havoc with most forms of feedback control, since the control system will interpret these rapid fluctuations as real pressure changes requiring corrective action. Although it is possible to configure some control systems to ignore such noise, the best solution is to correct the problem at the source either by relocating the pressure transmitter's impulse line tap to a place where it does not sense as great an amount of fluid turbulence, or somehow prevent that sensed turbulence from being represented in the transmitter's signal.

Since this noise is of a much greater frequency than the normal cycles of pressure in a process system, it is relatively easy to reduce the amount of noise in the transmitter signal simply by filtering that electronic signal using a low-pass filter circuit.

The simplest low-pass filter circuit is nothing more than a resistor and capacitor:

Figure 8.4: A simple low-pass filter to shunt high frequency noise to ground

Low-frequency voltage signals applied to this circuit emerge at the output terminal relatively unattenuated, because the reactance of the capacitor is quite large at low frequencies. High-frequency signals applied to the same circuit become attenuated by the capacitor, which tends to "short" those

signals to ground with its low reactance to high frequencies. The performance of such a filter circuit is primarily characterized by its *cutoff frequency*, mathematically defined as $f = \frac{1}{2\pi RC}$. The cutoff frequency is the point at which only 70.7% of the input signal appears at the output (a -3 dB attenuation in voltage).

If successfully applied to a process transmitter, such low-pass filtering has the effect of "quieting" an otherwise noisy signal so only the real process pressure changes are seen, while the effect of turbulence (or whatever else was causing the noise) becomes minimal. In the world of process control, the intentional low-pass filtering of process measurement signals is often referred to as *damping* because its effect is to "damp" (turn down) the effects of process noise:

Figure 8.5: A pre- and post-filtered pressure signal

In order for damping to be a useful tool for the engineer in mitigating measurement noise, it must be adjustable. In the case of the RC filter circuit, the degree of damping (cutoff frequency) may be adjusted by changing the value or either $R$ or $C$, with $R$ being the easier component to adjust. In digital transmitters where the damping is performed by a digital algorithm (either a sophisticated digital filtering routine or something as simple as successive averaging of buffered signal values in a first-in-first-out shift register), damping may be adjusted by setting a constant value. In pneumatic transmitters, damping could be implemented by installing viscous elements to the mechanism, or more simply by adding volume to the signal line (e.g. excess tubing length, larger tubing diameter, or even "capacity tanks" connected to the tube for increased volume).

The key question for the engineer then becomes, "how much damping is needed?" Insufficient damping will allow too much noise to reach the control system (causing "noisy" trends, indications, and erratic control), while excessive damping will cause the transmitter to understate the significance of sudden (real) process changes. There is a bad tendency for instrument technicians to apply excessive damping in transmitters. A transmitter with too much damping (i.e., cutoff frequency set too low, or time constant value set too high) causes the trend graph to be very smooth, which

at first appears to be a good thing. After all, the whole point of a control system is to hold the process variable tightly to setpoint, so the appearance of a "flat line" process variable trend is enticing indeed. However, the problem with excessive damping is that the transmitter gives a sluggish response to any sudden changes in the real process variable. A dual-trend graph of a pressure transmitter experiencing a sudden increase in process pressure shows this principle, where the undamped transmitter signal is shown in the upper portion and the over-damped signal in the lower portion (please note the vertical offset between these two trends is shown only for convenience in comparing the two trend shapes):

Figure 8.6: A comparison of an unfiltered signal with an improperly filtered version of the same signal

In summary, excessive damping causes the transmitter to "lie" to the control system by reporting a pressure that changes much slower than it actually does. The degree to which this "lie" adversely affects the control system (and/or the human operator's judgment in manually responding to the change in pressure) depends greatly on the nature of the control system and its importance to the overall plant operation. If any rule may be given as to how much damping to use in any transmitter, it is this: use as *little* as necessary to achieve good control.

When calibrating a transmitter in a shop environment, the damping adjustment should be set to its absolute minimum, so the results of applying stimuli to the transmitter are immediately seen by the technician. Any amount of damping in a transmitter being calibrated serves only to slow down the calibration procedure without benefit.

## 8.4 LRV and URV settings, digital trim (digital transmitters)

The advent of "smart" field instruments containing microprocessors has been a great advance for industrial instrumentation. These devices have built-in diagnostic ability, greater accuracy (due to digital compensation of sensor nonlinearities), and the ability to communicate digitally with host devices for reporting of various parameters.

A simplified block diagram of a "smart" pressure transmitter looks something like this:

Figure 8.7: A smart digital pressure transmitter block diagram

It is important to note all the adjustments within this device, and how this compares to the relative simplicity of an all-analog pressure transmitter:

Figure 8.8: An analog pressure transmitter block diagram

Note how the only calibration adjustments available in the analog transmitter are the "zero" and "span" settings. This is clearly not the case with smart transmitters. Not only can the lower- and upper-range values (LRV and URV) be set in a smart transmitter, but it is also possible to calibrate the analog-to-digital and digital-to-analog converter circuits independently of each other. What this means for the calibration technician is that a full calibration procedure on a smart transmitter potentially requires more work and a greater number of adjustments than an all-analog transmitter[2].

A common mistake made among students and experienced technicians alike is to confuse the range settings (LRV and URV) for actual calibration adjustments. Just because the LRV of a pressure transmitter has been set to 0.00 psig and the URV to 100.00 psig does not necessarily mean it will register accurately at points within that range! The following example will illustrate this fallacy.

Suppose a smart pressure transmitter is ranged for 0 to 100 psig with an analog output range of 4 to 20 mA, but this transmitter's pressure sensor is fatigued from years of use such that an actual applied pressure of 100 psig generates a signal that the analog-to-digital converter interprets

[2] Although those adjustments made on a digital transmitter tend to be easier to perform than repeated zero-and-span adjustments on analog transmitters due to the inevitable "interaction" between analog zero and span adjustments requiring repeated checking and re-adjustment during the calibration period.

as only 96 psig[3]. Assuming everything else in the transmitter is in perfect condition, with perfect calibration, the output signal will still be in error:

Figure 8.9: A smart pressure transmitter that is improperly trimmed

As the saying goes, "a chain is only as strong as its weakest link." The calibration of the most sophisticated pressure transmitter may be corrupted despite perfect calibration of both analog/digital converter circuits, and perfect range settings in the microprocessor. The microprocessor "thinks" the applied pressure is only 96 psig, and it responds accordingly with a 19.36 mA output signal. *The only way anyone would ever know this transmitter was inaccurate at 100 psig is to actually apply a known value of 100 psig fluid pressure to the sensor and note the incorrect response.* The lesson here should be clear: digitally setting a smart instrument's LRV and URV points does *not* constitute a legitimate calibration of the instrument.

For this reason, smart instruments always provide a means to perform what is called a *digital trim* on both the ADC and DAC circuits, to ensure the microprocessor "sees" the correct representation

[3] A 4% calibration error caused by sensor aging is enormous for any modern digital transmitter, and should be understood as an exaggeration presented only for the sake of illustrating how sensor error affects overall calibration in a smart transmitter. A more realistic amount of sensor error due to aging would be expressed in small fractions of a percent.

of the applied stimulus and to ensure the microprocessor's output signal gets accurately converted into a DC current, respectively.

Some technicians incorrectly use the LRV and URV settings in a manner not unlike the zero and span adjustments on an analog transmitter to correct errors such as this. Following this methodology, the URV of the fatigued transmitter would have to be set to 96 psig instead of 100 psig, so an applied pressure of 100 psig would give us the 20 mA output signal desired. In other words, this lets the microprocessor "think" it is only seeing 96 psig, then skew the URV so it outputs the correct signal anyway. Such an approach will work to an extent, but any digital queries to the transmitter (e.g. using a digital-over-analog protocol such as HART) will result in conflicting information, as the current signal represents full scale (100 psig) while the digital register inside the transmitter shows 96 psig. The only comprehensive solution to this problem is to "trim" the analog-to-digital converter so the transmitter's microprocessor "knows" the actual pressure value applied to the sensor.

Once digital trims have been performed on both input and output converters, of course, the technician is free to re-range the microprocessor as many times as desired without re-calibration. This capability is particularly useful when re-ranging is desired for special conditions, such as process start-up and shut-down when certain process variables drift into uncommon regions. An instrument technician may use a hand-held digital "communicator" device to re-set the LRV and URV range values to whatever new values are desired by operations staff without having to re-check calibration by applying known physical stimuli to the instrument. So long as the ADC and DAC trims are both fine, the overall accuracy of the instrument will still be good with the new range. With analog instruments, the only way to switch to a different measurement range was to change the zero and span adjustments, which *necessitated* the re-application of physical stimuli to the device (a full re-calibration). Here and here alone calibration is not necessary for a smart instrument. If overall measurement accuracy must be verified, however, there is no substitute for an actual physical calibration, and this entails both ADC and DAC "trim" procedures for a smart instrument.

Completely digital ("Fieldbus") transmitters are similar to "smart" analog-output transmitters with respect to distinct trim and range adjustments.

## 8.5 An analogy for calibration versus ranging

The concepts of *calibration* (trimming) and *ranging* are often difficult for new students of instrumentation to immediately grasp. A simple analogy useful for understanding these topics is that of setting a digital alarm clock.

Suppose a new digital alarm clock is supposed to alarm at 7:00 a.m. It would be foolish to simply unpack the new clock from its box, power it up, and set the wake-up time to 7:00 a.m. expecting it will alarm at the correct time. Before *trusting* this alarm time of 7:00 a.m., the clock would first have to be set, preferably synchronized to some standard time source. Otherwise, the wake-up setting of 7:00 a.m. will be hopelessly uncertain.

Once the clock is synchronized against a trusted time source, however, the wake-up (alarm) time may be set at will. The clock may then have the wake-up time changed from 7:00 a.m. to 8:00 a.m. without any need to re-synchronize (re-calibrate) the clock. The only reason for re-synchronizing the clock to the time standard is to compensate for inevitable *drift* due to imperfections in the clock circuitry.

Synchronizing the clock to a standard time source is analogous to "calibrating" or "trimming" a smart transmitter: it establishes an accurate correspondence between what the device's

microprocessor *perceives* and what the actual (real-life) values *are*. This step need only be done at the very beginning of the device's service, and every so often as warranted by the device's calibration drift over time[4].

Setting the wake-up (alarm) time on the clock is analogous to setting the LRV and URV parameters of a smart transmitter: it defines the *action(s)* taken by the device at certain measured values. For the alarm clock, it defines the hour and minute of day when the alarm sounds. For a digital transmitter, it defines the measured variable values at which it will output 4 mA and 20 mA (for a 4-20 mA analog output range).

By contrast, an analog transmitter blends the functions of calibration and ranging into one. A useful analogy for this is to imagine using a simple wind-up mechanical timer to alarm at 7:00 a.m. Such a crude timing device does not even register time in hours and minutes like a digital alarm clock: instead, it simply counts down time from its starting point and sounds an alarm when the descending count reaches zero. The current time must be known and the time differerence must be calculated to 7:00 a.m. (e.g., if setting the wind-up alarm occurs at 10:30 PM, this would equate to a timing period of 8.5 hours) in order to set this device for a 7:00 a.m. wake-up alarm.

Every single time this wind-up alarm is set, a time standard must be referenced to know how many hours and minutes of count-down time is needed. If it should alarm a different time in the morning, once again a standard time source must be consulted, the necessary arithmetic performed, and the timer set accordingly. Setting the alarm time on this mechanism necessitates re-calibrating it to the local standard time every single time. Here, there is no distinction between synchronization and alarm setting; no distinction between calibration and ranging.

## 8.6 Calibration procedures

As described earlier in this chapter, *calibration* refers to the adjustment of an instrument so its output accurately corresponds to its input throughout a specified range. This definition specifies the outcome of a calibration process, but not the procedure. It is the purpose of this section to describe procedures for efficiently calibrating different types of instruments.

### 8.6.1 Linear instruments

The simplest calibration procedure for an analog, linear instrument is the so-called *zero-and-span* method. The method is as follows:

1. Apply the lower-range value stimulus to the instrument, wait for it to stabilize
2. Move the "zero" adjustment until the instrument registers accurately at this point
3. Apply the upper-range value stimulus to the instrument, wait for it to stabilize
4. Move the "span" adjustment until the instrument registers accurately at this point
5. Repeat steps 1 through 4 as necessary to achieve good accuracy at both ends of the range

An improvement over this crude procedure is to check the instrument's response at several points between the lower- and upper-range values. A common example of this is the so-called *five-point*

[4] In the case of pressure transmitters, re-trimming may be necessary if the device is ever re-mounted in a different orientation. Changing the physical orientation of a pressure transmitter alters the direction in which gravity tugs on the sensing element, causing it to respond as though a constant bias pressure were applied to it. This bias is often on the order of an inch of water column (or less), and usually consequential only for low-pressure applications such as furnace draft pressure.

*calibration* where the instrument is checked at 0% (LRV), 25%, 50%, 75%, and 100% (URV) of range. A variation on this theme is to check at the five points of 10%, 25%, 50%, 75%, and 90%, while still making zero and span adjustments at 0% and 100%. Regardless of the specific percentage points chosen for checking, the goal is to achieve (at least) the minimum necessary accuracy at all points along the scale, so the instrument's response may be trusted when placed into service.

Yet another improvement over the basic five-point test is to check the instrument's response at five calibration points *decreasing* as well as *increasing*. Such tests are often referred to as *Up-down* calibrations. The purpose of such a test is to determine if the instrument has any significant *hysteresis*: a lack of responsiveness to a change in direction.

Some analog instruments provide a means to adjust linearity. This adjustment should be moved only if absolutely necessary! Quite often, these linearity adjustments are very sensitive, and prone to over-adjustment by zealous fingers. The linearity adjustment of an instrument should be changed only if the required accuracy cannot be achieved across the full range of the instrument. Otherwise, it is advisable to adjust the zero and span controls to "split" the error between the highest and lowest points on the scale, and leave linearity alone.

The procedure for calibrating a "smart" digital transmitter – also known as *trimming* – is a bit different. Unlike the zero and span adjustments of an analog instrument, the "low" and "high" trim functions of a digital instrument are typically non-interactive. This means the low- and high-level stimuli only have to applied *once* during a calibration procedure. Trimming the sensor of a "smart" instrument consists of these four general steps:

1. Apply the lower-range value stimulus to the instrument, wait for it to stabilize
2. Execute the "low" sensor trim function
3. Apply the upper-range value stimulus to the instrument, wait for it to stabilize
4. Execute the "high" sensor trim function

Likewise, trimming the output (Digital-to-Analog Converter, or DAC) of a "smart" instrument consists of these six general steps:

1. Execute the "low" output trim test function
2. Measure the output signal with a precision milliammeter, noting the value after it stabilizes
3. Enter this measured current value when prompted by the instrument
4. Execute the "high" output trim test function
5. Measure the output signal with a precision milliammeter, noting the value after it stabilizes
6. Enter this measured current value when prompted by the instrument

After both the input and output (ADC and DAC) of a smart transmitter have been trimmed (i.e., calibrated against standard references known to be accurate), the lower- and upper-range values may be set. In fact, once the trim procedures are complete, the transmitter may be ranged and ranged again as many times as desired. The only reason for re-trimming a smart transmitter is to ensure accuracy over long periods of time where the sensor and/or the converter circuitry may have drifted out of acceptable limits. This stands in stark contrast to analog transmitter technology, where re-ranging *necessitates* re-calibration.

### 8.6.2 Nonlinear instruments

The calibration of inherently nonlinear instruments is much more challenging than for linear instruments. No longer are two adjustments (zero and span) sufficient, because more than two points are necessary to define a curve.

Examples of nonlinear instruments include expanded-scale electrical meters, square root characterizers, and position-characterized control valves.

Every nonlinear instrument will have its own recommended calibration procedure, so defer to the manufacturer's literature for the specific instrument. However, when calibrating a nonlinear instrument, document all the adjustments made (e.g., how many turns on each calibration screw) just in case the instrument must be returned back to its original condition. It is good to know exactly how to reverse the steps taken and start over if necessary!

### 8.6.3 Discrete instruments

The word "discrete" means *individual* or *distinct*. In engineering, a "discrete" variable or measurement refers to a true-or-false condition. Thus, a discrete sensor is one that is only able to indicate whether the measured variable is above or below a specified setpoint.

Examples of discrete instruments are *process switches* designed to turn on and off at certain values. A pressure switch, for example, used to turn an air compressor on if the air pressure ever falls below 85 psig, is an example of a discrete instrument.

Discrete instruments require periodic calibration just like continuous instruments. Most discrete instruments have but one calibration adjustment: the *set-point* or *trip-point*. Some process switches have two adjustments: the set-point as well as a *deadband* adjustment. The purpose of a deadband adjustment is to provide an adjustable buffer range that must be traversed before the switch changes state. To use the 85 psig low air pressure switch as an example, the set-point would be 85 psig, but if the deadband were 5 psid it would mean the switch would not change state until the pressure rose above 90 psig (85 psig + 5 psid).

When calibrating a discrete instrument, check the accuracy of the setpoint *in the proper direction of stimulus change*. For the air pressure switch example, this would mean checking to see that the switch changes states at 85 psig *falling*, not 85 psig *rising*. If it were not for the existence of deadband, it would not matter which way the applied pressure changed during the calibration test. However, deadband will always be present in a discrete instrument, whether that deadband is adjustable or not.

For example, a pressure switch with a deadband of 5 psid set to trip at 85 psig falling would re-set at 90 psig rising. Conversely, a pressure switch (with the same deadband of 5 psid) set to trip at 85 psig rising would re-set at 80 psig falling. In both cases, the switch "trips" at 85 psig, but the direction of pressure change specified for that trip point defines which side of 85 psig the re-set pressure will be found.

A procedure to efficiently calibrate a discrete instrument without too many trial-and-error attempts is to set the stimulus at the desired value (e.g. 85 psig for the hypothetical low-pressure switch) and then move the set-point adjustment in the *opposite* direction as the intended direction of the stimulus (in this case, *increasing* the set-point value until the switch changes states). The basis for this technique is the realization that most comparison mechanisms cannot tell the difference between a rising process variable and a falling setpoint (or visa-versa). Thus, a falling pressure may be simulated by a rising set-point adjustment. After doing this, perform an actual changing-stimulus test to ensure the instrument responds properly under realistic circumstances, but this "trick" will help achieve good calibration in less time.

## 8.7 Typical calibration errors

Recall that the slope-intercept form of a linear equation describes the response of a linear instrument:

$$y = mx + b$$

Where,
$y$ = Output
$m$ = Span adjustment
$x$ = Input
$b$ = Zero adjustment

A *zero shift* calibration error shifts the function vertically on the graph. This error affects *all* calibration points equally, creating the same percentage of error across the entire range:

Figure 8.10: The calibration curve of a pressure transmitter showing zero shifts

A *span shift* calibration error shifts the slope of the function. This error's effect is unequal at different points throughout the range:

Figure 8.11: The calibration curve of a pressure transmitter showing span shifts

A *linearity* calibration error causes the function to deviate from a straight line. This type of error does not directly relate to a shift in either zero ($b$) or span ($m$) because the slope-intercept equation only describes straight lines. If an instrument does not provide a linearity adjustment, "split the error" between high and low extremes, so the maximum absolute error at any point in the range is minimized:

Figure 8.12: The calibration curve of a pressure transmitter showing linearity errors

A *hysteresis* calibration error occurs when the instrument responds differently to an increasing input compared to a decreasing input. The only way to detect this type of error is to do an *up-down* calibration test, checking for instrument response at the same calibration points going down as going up:

Figure 8.13: The calibration curve of a pressure transmitter showing hysteresis error

Hysteresis errors are almost always caused by mechanical friction on some moving element (and/or a loose coupling between mechanical elements) such as bourdon tubes, bellows, diaphragms, pivots, levers, or gear sets. Flexible metal strips called *flexures* – which are designed to serve as frictionless pivot points in mechanical instruments – may also cause hysteresis errors if cracked or bent.

In practice, most calibration errors are some combination of zero, span, linearity, and hysteresis problems.

### 8.7.1 As-found and as-left documentation

An important principle in calibration practice is to document every instrument's calibration as it was found *and* as it was left after adjustments were made. The purpose for documenting both conditions is to make data available for calculating instrument *drift* over time. If only one of these conditions is documented during each calibration event, it will be difficult to determine how well an instrument is holding its calibration over long periods of time. Excessive drift is often an indicator of impending failure, which is vital for any program of predictive maintenance or quality control.

Typically, the format for documenting both As-Found and As-Left data is a simple table showing the points of calibration, the ideal instrument responses, the actual instrument responses, and the calculated error at each point. The following table is an example for a pressure transmitter with a range of 0 to 200 psig over a five-point scale:

| **Percent of range** | **Input** pressure | **Output current** (ideal) | **Output current** (measured) | **Error** (percent of span) |
|---|---|---|---|---|
| 0% | 0 psig | 4.00 mA | | |
| 25% | 50 psig | 8.00 mA | | |
| 50% | 100 psig | 12.00 mA | | |
| 75% | 150 psig | 16.00 mA | | |
| 100% | 200 psig | 20.00 mA | | |

### 8.7.2 Up-tests and Down-tests

It is not uncommon for calibration tables to show multiple calibration points going *up* as well as going *down*, for the purpose of documenting hysteresis and deadband errors. Note the following example, showing a transmitter with a maximum hysteresis of 0.313 % (the offending data points are shown in bold-faced type):

| **Percent of range** | **Input** pressure | **Output current** (ideal) | **Output current** (measured) | **Error** (percent of span) |
|---|---|---|---|---|
| 0% | 0 psig | 4.00 mA | 3.99 mA | -0.0625 % |
| 25% ↑ | 50 psig | 8.00 mA | **7.98 mA** | **-0.125 %** |
| 50% ↑ | 100 psig | 12.00 mA | 11.99 mA | -0.0625 % |
| 75% ↑ | 150 psig | 16.00 mA | 15.99 mA | -0.0625 % |
| 100% ↑ | 200 psig | 20.00 mA | 20.00 mA | 0 % |
| 75% ↓ | 150 psig | 16.00 mA | 16.01 mA | +0.0625 % |
| 50% ↓ | 100 psig | 12.00 mA | 12.02 mA | +0.125 % |
| 25% ↓ | 50 psig | 8.00 mA | **8.03 mA** | **+0.188 %** |
| 0% ↓ | 0 psig | 4.00 mA | 4.01 mA | +0.0625 % |

In the course of performing such a directional calibration test, it is important not to overshoot any of the test points. If this occurs in setting up one of the input conditions for the instrument, simply "back up" the test stimulus and re-approach the test point from the same direction as before. Unless each test point's value is approached from the proper direction, the data cannot be used to determine hysteresis/deadband error.

## 8.8 NIST traceability

As defined previously, *calibration* means the comparison and adjustment (if necessary) of an instrument's response to a stimulus of precisely known quantity, to ensure operational accuracy. In order to perform a calibration, the physical quantity used to stimulate the instrument must be accurate. For example, to calibrate a pressure gauge to read accurately at an applied pressure of 200 psig, the pressure used to stimulate the gauge must actually be 200 psig. If it is not 200 psig, then it makes no sense to adjust the pressure gauge to register 200 psig when in fact it is sensing something different.

Ultimately, this is a philosophical question of *epistemology*: *what is true?* There are no easy answers here, but teams of scientists and engineers known as *metrologists* devote their professional lives to the study of calibration standards to ensure access to the best approximation of "truth" for calibration purposes. *Metrology* is the science of measurement, and the central repository of expertise on this science within the United States of America is the *National Institute of Standards and Technology*, or the *NIST* (formerly known as the *National Bureau of Standards*, or *NBS*).

Experts at the NIST work to ensure there is a means of tracing measurement accuracy back to *intrinsic standards*, which are quantities inherently fixed (as far as anyone knows). The vibrational frequency of an isolated cesium atom when stimulated by radio energy, for example, is an intrinsic standard used for the measurement of time (forming the basis of the so-called *atomic clock*). So far as anyone knows, this frequency is fixed in nature and cannot vary: each and every isolated cesium atom has the exact same resonant frequency. The distance traveled in a vacuum by 1,650,763.73 wavelengths of light emitted by an excited krypton-86 ($^{86}$Kr) atom is the intrinsic standard for one meter of length. Again, so far as anyone knows, this distance is fixed in nature and cannot vary. This means any suitably equipped laboratory in the world should be able to build their own intrinsic standards to reproduce the *exact* same quantities based on the same (universal) physical constants. Intrinsic standards therefore serve as absolute references against which certain instruments may be calibrated.

The machinery necessary to replicate intrinsic standards for practical use are quite expensive and usually delicate. This means the average metrologist (let alone the average industrial instrument technician) simply will never have access to one. In order for these intrinsic standards to be useful within the industrial world, they have to be used to calibrate other instruments, which are then used to calibrate other instruments, and so on until finally the instrument is calibrated for field service in a process. So long as this "chain" of instruments is calibrated against each other regularly enough to ensure good accuracy at the end-point, field instruments may be calibrated with confidence. The documented confidence is known as *NIST traceability*: that the accuracy of the field instrument calibrated is ultimately ensured by a trail of documentation leading to intrinsic standards maintained by the NIST. This "paper trail" proves to anyone interested that the accuracy of a calibrated field instrument is of the highest pedigree.

## 8.9 Instrument turndown

An important performance parameter for transmitter instruments is something often referred to as *turndown* or *rangedown*. "Turndown" is defined as the ratio of maximum allowable span to the minimum allowable span for a particular instrument.

Suppose a pressure transmitter has a maximum calibration range of 0 to 300 pounds per square inch (psig), and a turndown of 20:1. This means that a technician may adjust the span anywhere between 300 psig and 15 psig. This is important to know in order to select the proper transmitter for any given measurement application. The odds of finding a transmitter with just the perfect factory-calibrated range for the measurement application may be quite small, meaning its range will have to be adjusted to fit the need. The turndown ratio tells how far the instrument's range will be able to be practically adjusted.

For example, suppose the operations personnel request a pressure transmitter installed on a process vessel with a measurement range of 50 psig to 90 psig. The warehouse contains a pressure transmitter with a (maximum) range of zero to 1000 psig and a turndown ratio of 20:1. Dividing the

maximum span of 1000 psig by 20 determines a minimum span of 50 psig. The span requested by operations for this pressure transmitter is 40 psig (90 psig − 50 psig), which means the transmitter found in the warehouse will *not* be able to "turn down" that far. At best, it could be ranged for 50 psig to 100 psig, or perhaps for 40 psig to 90 psig, but not the 50 psig to 90 psig requested by operations. At this point, return to the operations personnel to ask if a 50 psig span would be acceptable – if not, order a different pressure transmitter with a smaller span (or with a greater turndown ratio[5]).

## 8.10 Practical calibration standards

Within the context of a calibration shop environment, where accurate calibrations are important yet intrinsic standards are not readily accessible, a workable degree of accuracy must be maintained in the calibration equipment used to calibrate field instruments. This typically requires sending out the calibration equipment to a calibration lab to have it calibrated periodically per the manufacturer's instructions.

It is important that the degree of uncertainty in the accuracy of a test instrument is *significantly less* than the degree of uncertainty in the instruments to be calibrated. Otherwise, calibration becomes an exercise in futility. This ratio of uncertainties is called the *Test Uncertainty Ratio*, or *TUR*. A good rule-of-thumb is to maintain a TUR of at least 4:1 (ideally 10:1 or better), the test equipment being many times more accurate (less uncertain) than the field instruments calibrated with it.

Trying to calibrate a field instrument to a tighter tolerance than what the calibrating equipment possesses will result in confusion and wasted time. In one case, an instrument technician attempted to calibrate a pneumatic pressure transmitter to a tolerance of +/- 0.5% of span using a test gauge that was only good for +/- 1% of the same span. This poor technician kept going back and forth, adjusting zero and span over and over again, trying to stay within the stated specification of 0.5%. After giving up, he tested the test gauges by comparing three of them, one against the other. When it was realized no two test gauges would agree with each other to within the tolerance he was trying to achieve in calibrating the transmitter, the problem became clear.

The lesson to be learned here is to always ensure the equipment used to calibrate industrial instruments is reliably accurate (enough). No piece of test equipment will ever be *perfectly* accurate, but perfection is not what is needed. The goal is to be *accurate enough* that the final calibration will be reliable within specified boundaries.

The next few subsections describe various standards used in instrument shops to calibrate industrial instruments.

### 8.10.1 Electrical standards

Electrical calibration equipment – used to calibrate instruments measuring voltage, current, and resistance – must be periodically calibrated against higher-tier standards maintained by outside laboratories. In years past, instrument shops would often maintain their own *standard cell* batteries

[5] Modern "smart" electronic pressure transmitters typically boast turndown ratios exceeding 100:1, with some having turndown ratios of 200:1 or more! Large turndown ratios are good because they allow users of instrumentation to maintain a smaller quantity of new transmitters in stock, since transmitters with large turndown ratios are more versatile (i.e., applicable to a wider variety of spans) than transmitters with small turndown ratios.

(often called *Weston* cells) as a primary voltage reference. These special-purpose batteries produced 1.0183 volts DC at room temperature with low uncertainty and drift, but were sensitive to vibration and non-trivial to actually use. Now, electronic voltage references have all but displaced standard cells in calibration shops and laboratories, but these references must be checked and adjusted for drift in order to maintain their NIST traceability.

One enormous benefit of electronic calibration references is that they are able to generate accurate currents and resistances in addition to voltage (and not just voltage at one fixed value, either!). Modern electronic references are digitally-controlled as well, which lends themselves well to automated testing in assembly-line environments, and/or programmed multi-point calibrations with automatic documentation of as-found and as-left calibration data.

If a shop cannot afford one of these versatile references for benchtop calibration use, an acceptable alternative in some cases is to purchase a high-accuracy multimeter and equip the calibration bench with adjustable voltage, current, and resistance sources. These sources will be simultaneously connected to the high-accuracy multimeter and the instrument under test, and adjusted until the high-accuracy meter registers the desired value. The measurement shown by the instrument under test is then compared against the reference meter and adjusted until matching (to within the required tolerance). The following illustration shows how a high-accuracy voltmeter could be used to calibrate a handheld voltmeter in this fashion:

Figure 8.14: The calibration setup for a voltmeter

It should be noted that the variable voltage source shown in this test arrangement need not be sophisticated. It simply needs to be *variable* (to allow precise adjustment until the high-accuracy

voltmeter registers the desired voltage value) and *stable* (so the adjustment will not drift appreciably over time).

### 8.10.2 Temperature standards

The most common technologies for industrial temperature measurement are electronic in nature: RTDs and thermocouples. As such, the standards used to calibrate such devices are the same standards used to calibrate electrical instruments such as digital multimeters (DMMs). For RTDs, this means a precision resistance standard such as a *decade box* used to precisely set known quantities of electrical resistance. For thermocouples, this means a *precision potentiometer* used to generate precise quantities of low DC voltage (in the millivolt range, with microvolt resolution). Modern, electronic calibrators are also available now for RTD and thermocouple instrument calibration, able to generate accurate quantities of electrical resistance and DC millivoltage for the simulation of RTD and thermocouple elements, respectively.

However, there are some temperature-measuring instruments that are not electrical in nature. This category includes bimetallic thermometers, filled-bulb temperature systems, and optical pyrometers. In order to calibrate these types of instruments, the calibration temperatures must be accurately created in the instrument shop. In other words, the instrument to be calibrated must be subjected to an actual temperature of accurately known value.

Even with RTDs and thermocouples – where the sensor signal may be easily simulated using electronic test equipment – there is merit in using an actual source of precise temperature to calibrate the temperature transmitter or indicator. Simulating the voltage produced by a thermocouple at a precise temperature, for example, is fine for calibrating the instrument normally receiving the millivoltage signal from the thermocouple, but this calibration test does nothing to validate the accuracy of the thermocouple element itself! The *best* type of calibration for any temperature-measuring instrument, from the perspective of overall integrity, is to actually subject the sensing element to a precisely known temperature. Special calibration equipment designed to produce accurate temperature samples on demand is needed for this.

A time-honored standard for low-temperature industrial calibrations is pure water, specifically the freezing and boiling points of water. Pure water at sea level (full atmospheric pressure) freezes at 32 degrees Fahrenheit (0 degrees Celsius) and boils at 212 degrees Fahrenheit (100 degrees Celsius). In fact, the Celsius temperature scale is *defined* by these two points of phase change for water at sea level.

To use water as a temperature calibration standard, simply prepare a vessel for one of two conditions: thermal equilibrium at freezing or thermal equilibrium at boiling. "Thermal equilibrium" in this context simply means equal temperature throughout the mixed-phase sample. In the case of freezing, this means a well-mixed sample of solid ice and liquid water. In the case of boiling, this means a pot of water at a steady boil (vaporous steam and liquid water in direct contact). Ample contact between the two phases (either solid and liquid; or liquid and vapor) is needed to eliminate hot or cold spots. When the entire water sample is homogeneous in temperature and changing phase (either freezing or boiling), the sample will have only one degree of thermodynamic freedom: its temperature is an exclusive function of atmospheric pressure. Since atmospheric pressure is relatively stable and well-known, this fixes the temperature at a constant value. For ultra-precise temperature calibrations in laboratories, the *triple point* of water is used as the reference. When water is brought to its triple point, the sample will have *zero degrees* of thermodynamic freedom, which means both its temperature and its pressure will become locked at stable values: pressure at

0.006 atmospheres, and temperature at 0.01 degrees Celsius.

The major limitation of water as a temperature calibration standard is it only provides two points of calibration: 0 $^o$C and 100 $^o$C, with the latter[6] being strongly pressure-dependent. If other reference temperatures are required for a calibration, some substance other than water must be used.

A variety of substances with known phase-change points have been standardized as fixed points on the International Practical Temperature Scale (ITS-90). The following list is a sample of some of these substances and their respective phase states and temperatures[7]:

- Neon (triple point) = -248.6 $^o$C
- Oxygen (triple point) = -218.8 $^o$C
- Mercury (triple point) = -38.83 $^o$C
- Tin (freezing point) = 231.93 $^o$C
- Zinc (freezing point) = 419.53 $^o$C
- Aluminum (freezing point) = 660.32 $^o$C
- Copper (freezing point) = 1084.62 $^o$C

Substances at the triple point must be in thermal equilibrium with solid, liquid, and vaporous phases co-existing. Substances at the freezing point must be a two-phase mixture of solid and liquid (i.e., a liquid *in the process of freezing*, neither a completely liquid nor a completely solid sample). The physical principle at work in all of these examples is that of *latent heat*: the thermal energy exchange required to change the phase of a substance. So long as the minimum heat exchange requirement for complete phase change is not met, a substance in the midst of phase transition will exhibit a fixed temperature, and therefore behave as a temperature *standard*. Small amounts of heat gain or loss to such a sample will merely change the proportion of one phase to another (e.g., how much solid versus how much liquid), but the temperature will remain locked at a constant value until the sample becomes a single phase.

One major disadvantage of using phase changes to produce accurate temperatures in the shop is the limited availability of temperatures. To create some other temperature for calibration purposes, either a suitable material with a phase change happening at that exact same temperature is needed (good luck finding one) or a finely adjustable temperature source and use an accurate thermometer are needed to compare against the instrument under test. The latter scenario is analogous to the use of a high-accuracy voltmeter and an adjustable voltage source to calibrate a voltage instrument: comparing one instrument (trusted to be accurate) against another (under test).

Laboratory-grade thermometers are relatively easy to secure. Variable temperature sources suitable for calibration use include *oil bath* and *sand bath* calibrators. These devices are exactly what they sound like: small pots filled with either oil or sand, containing an electric heating element and a temperature control system using a laboratory-grade (NIST-traceable) thermal sensor. In the case of sand baths, a small amount of compressed air is introduced at the bottom of the vessel to "fluidize" the sand so the grains move around much like the molecules of a liquid, helping the system reach thermal equilibrium. To use a bath-type calibrator, place the temperature instrument to be

[6] Pressure does have some influence on the freezing point of most substances as well, but not nearly to the degree it has on the boiling point. For a comparison between the pressure-dependence of freezing versus boiling points, consult a phase diagram for the substance in question, and observe the slopes of the solid-liquid phase line and liquid-vapor phase line. A nearly-vertical solid-liquid phase line shows a weak pressure dependence, while the liquid-vapor phase lines are typically much closer to horizontal.

[7] For each of these examples, the assumptions of a 100% pure sample and an airless testing environment are made. Impurities in the initial sample and/or resulting from chemical reactions with air at elevated temperatures, may introduce serious errors.

calibrated such the sensing element dips into the bath, then wait for the bath to reach the desired temperature.

An oil bath temperature calibrator is shown in the following photograph, with sockets to accept seven temperature probes into the heated oil reservoir:

Figure 8.15: An oil bath temperature calibrator

*Dry-block* temperature calibrators also exist for creating accurate calibration temperatures in the instrument shop environment. Instead of a fluid (or fluidized powder) bath as the thermal medium, these devices use metal blocks with blind (dead-end) holes drilled for the insertion of temperature-sensing instruments.

An inexpensive dry-block temperature calibrator intended for bench-top service is shown in this photograph:

Figure 8.16: An dry-block temperature calibrator

Optical temperature instruments require a different sort of calibration tool: one that emits radiation equivalent to that of the process object at certain specified temperatures. This type of calibration tool is called a *blackbody calibrator*, having a target area where the optical instrument may be aimed. Like oil and sand bath calibrators, a blackbody calibrator relies on an internal temperature sensing element as a reference, to control the optical emissions of the blackbody target at any specified temperature within a practical range.

### 8.10.3 Pressure standards

In order to accurately calibrate a pressure instrument in a shop environment, fluid pressures of known magnitude must be created against which the instrument being calibrated is compared. As with other types of physical calibrations, the methods used can be separated into two broad categories: use devices that inherently *produce* known pressures or use devices that accurately measure pressures created by some (other) adjustable source.

A *deadweight tester* (sometimes referred to as a *dead-test* calibrator) is an example in the former category. These devices *create* accurately known pressures by means of precise masses and pistons of precise area:

Figure 8.17: Diagram of a deadweight pressure calibration

After connecting the gauge (or other pressure instrument) to be calibrated, the technician adjusts the secondary piston to cause the primary piston to lift off its resting position and be suspended by oil pressure alone. So long as the mass placed on the primary piston is precisely known, Earth's gravitational field is constant, and the piston is perfectly vertical, the fluid pressure applied to the instrument under test *must* be equal to the value described by the following equation:

$$P = \frac{F}{A}$$

Where,

$P$ = Fluid pressure

$F$ = Force exerted by the action of gravity on the mass ($F_{weight} = mg$)

$A$ = Area of piston

The primary piston area, of course, is precisely set at the time of the deadweight tester's manufacture and does not change appreciably throughout the life of the device.

A very simple deadweight tester unit appears in the next photograph, mounted to a yellow wooden base:

Figure 8.18: A deadweight pressure calibrator

When sufficient pressure has been accumulated inside the tester to overcome the weight on the piston, the piston rises off its rest and "floats" on the pressurized oil, as shown in this close-up photograph:

Figure 8.19: Close up of a deadweight pressure calibrator piston

A common operating practice for any deadweight tester is to gently spin the mass during testing so the primary piston continually rotates within its cylinder. Any motion will prevent static friction from taking hold, helping to ensure the only force on the primary piston is the force of the fluid within the deadweight tester.

Most modern deadweight testers include extra features such as hand pumps and bleed valves in addition to secondary pistons, to facilitate both rapid and precise operation. The next photograph shows a newer deadweight tester, with these extra features:

Figure 8.20: A newer model deadweight pressure calibrator

There is also such a thing as a *pneumatic* deadweight tester. In these devices, a constant flow of gas such as compressed air or bottled nitrogen vents through a bleed port operated by the primary piston. The piston moves as necessary to maintain just enough gas pressure inside the unit to suspend the mass(es) against gravity. This gas pressure passes on to the instrument under test, just as liquid pressure in a hydraulic deadweight tester passes to the test instrument for comparison:

Figure 8.21: A diagram of a pneumatic deadweight pressure calibrator

In fact, the construction and operation of a pneumatic deadweight tester is quite similar to a

self-balancing (force-balance) pneumatic instrument mechanism with a baffle/nozzle assembly. A moving element opens or closes a variable restriction downstream of a fixed restriction to generate a varying pressure. In this case, that pressure directly operates the bleed vent to self-regulate gas pressure at whatever value is necessary to suspend the mass against gravity.

Deadweight testers (both hydraulic and pneumatic) lend themselves well to relatively high pressures, owing to the practical limitations of mass and piston area. A deadweight tester can be used to calibrate a 100 psig pressure gauge used for measuring pressure of a water main, for example, but a deadweight tester can't be used to calibrate a 0 to 1 "W.C. (zero to one inch water column) pressure gauge used to measure draft pressure in a furnace flue.

For low-pressure calibrations, the simple *manometer* is a much more practical standard. Manometers, of course, do not generate pressure on their own. In order to use a manometer to calibrate a pressure instrument, both devices must be connected to a source of variable fluid pressure, typically instrument air through a precision pressure regulator or a low-pressure test pump:

Figure 8.22: A manometer used as a calibration standard

The difference in liquid column heights ($h$) within the manometer shows the pressure applied to the gauge. As with the deadweight tester, the accuracy of this pressure measurement is bound by just a few physical constants, none of which are liable to spurious change. So long as the manometer's liquid density is precisely known, Earth's gravitational field is constant, and the manometer tubes are perfectly vertical, the fluid pressure indicated by the manometer *must* be equal to the value described by the following equation (two different forms given):

$$P = \rho g h \qquad \text{(or)} \qquad P = \gamma h$$

Where,

$P$ = Fluid pressure

$\rho$ = Mass density of fluid

$\gamma$ = Weight density of fluid
$g$ = Acceleration of gravity
$h$ = Height difference between manometer liquid columns

Of course, with pressure-measuring test instruments of suitable accuracy (preferably NIST-traceable), the same sort of calibration jig may be used for virtually any desired range of pressures:

Figure 8.23: A diagram of a pressure calibration setup using an electronic gauge

When the electronic test gauge is designed for very low pressures (inches of water column), they are sometimes referred to as *electronic manometers*.

Instrument calibrations performed in the field (i.e., in locations near or at the intended point of use rather than in a professionally-equipped shop) are almost always done this way: a pressure-generating source is connected to both the instrument under test and a trusted calibration gauge ("test gauge"), and the two indications are compared at several points along the calibrated range. Test equipment suitable for field pressure calibrations include *slack-tube manometers* made from flexible plastic tubing hung from any available anchor point near eye level, and *test gauges* typically of the helical bourdon tube variety. Portable electronic test gauges are also available for field use, many with built-in hand pumps for generating precise air pressures.

A noteworthy example of a pneumatic pressure calibrator for field use was a device manufactured by the Wallace & Tiernan corporation, affectionately called a *Wally box* by at least one generation of instrument technicians. A "Wally box" consisted of a large dial pressure gauge (several inches in diameter) with a multi-turn needle and a very fine scale, connected to a network of valves and regulators which were used to set different air pressures from any common compressed air source. The entire mechanism was housed in an impact-resistance case for ruggedness. One of the many nice features of this calibration instrument was a selector valve allowing the technician to switch between two different pressures output by independent pressure regulators. Once the two pressure regulator values were set to the instrument's lower- and upper-range values (LRV and URV), it was possible to switch back and forth between those two pressures at will, making the task of adjusting an analog instrument with interactive zero and span adjustments much easier than it would have been to precisely adjust a single pressure regulator again and again.

### 8.10.4 Flow standards

Most forms of continuous flow measurement are inferential; that is, flow is measured indirectly by measuring some other variable (such as pressure, voltage, or frequency) directly that has a known functional relationship to flow. With this in mind, reasonable calibration accuracy may usually be achieved simply by calibrating the primary sensor and replacing the flow element (if inspection proves necessary). In the case of an orifice plate used to measure fluid flow rate, this would mean calibrating the differential pressure transmitter to measure pressure accurately and replacing the orifice plate if it shows signs of wear.

In some cases, though, direct validation of flow measurement accuracy is needed. Most techniques of flow rate validation take the form of measuring accumulated fluid volume over time. This may prove to be complicated, especially if the fluids in question are hazardous in any way, and/or the flow rates are large, and/or the fluid is a gas or vapor.

For simple validation of liquid flow rates, the flow may be diverted from its normal path in the process and into a container where either accumulated volume or accumulated weight may be measured over time. If the rate of flow into this container is constant, the accumulated volume (or weight) should increase linearly over time. The actual flow rate may then be calculated by dividing the change in volume ($\Delta V$) by the time interval over which the change in volume was measured ($\Delta t$). The resulting quotient is the average flow rate between those two points in time, which is an approximation of instantaneous flow rate:

$$\frac{\Delta V}{\Delta t} = \text{ Average flow}$$

$$\frac{\Delta V}{\Delta t} \approx \frac{dV}{dt} = \text{ Instantaneous flow}$$

If a suitable vessel exists in the process with level-measuring capability (e.g., a liquid storage vessel equipped with a level transmitter), the same mathematical technique may be applied: use that vessel as an accumulator for the flow in question, tracking the accumulated (or lost) volume over time and then calculating $\frac{\Delta V}{\Delta t}$. The accuracy of this technique rests on some additional factors, though:

- The accuracy of the level transmitter (as a *volume* measuring instrument!)
- The ability to ensure only *one* flow path in or out of that vessel

The first condition listed here places significant limitations on the flow calibration accuracy that can be achieved with this method. In essence, the level instrument is used as the "test gauge" for the flow instrument, so it needs to possess high accuracy in order to achieve even reasonable accuracy for the flowmeter being calibrated.

A more sophisticated approach for direct flow validation is the use of a device called a *flow prover*. A "flow prover" is a precision piston-and-cylinder mechanism used to precisely measure a quantity of liquid over time. Process flow is diverted through the prover, moving the piston over time. Sensors on the prover mechanism detect when the piston has reached certain positions, and time measurements taken at those different positions enable the calculation of average flow ($\frac{\Delta V}{\Delta t}$).

### 8.10.5 Analytical standards

An *analyzer* measures intrinsic properties of a substance sample such as its density, chemical content, or purity. Whereas the other types of instruments discussed in this chapter measure quantities

incidental to the composition of a substance (pressure, level, temperature, and flow rate), an analyzer measures something related to the *nature* of substance being processed.

As previously defined, to *calibrate* an instrument means to check and adjust (if necessary) its response so the output accurately corresponds to its input throughout a specified range. In order to do this, one must expose the instrument to an actual input stimulus of precisely known quantity. This is no different for an analytical instrument. In order to calibrate an analyzer, it must be exposed to known quantities of substances with the desired range of properties (density, chemical composition, etc.).

A classic example of this is the calibration of a pH analyzer. pH is the measurement of hydrogen ion activity in an aqueous solution. The standard range of measurement is 0 pH to 14 pH, the number representing a negative power of 10 approximately describing the hydrogen ion molarity of the solution (how many moles of active hydrogen ions per liter of solution)[8].

The pH of a solution is typically measured with a pair of special electrodes immersed in the solution, which generate a voltage proportional to the pH of the solution. In order to calibrate a pH instrument, it must be compared against a sample of liquid solution with a known pH value. For pH instrumentation, such calibration solutions are called *buffers*, because they are specially formulated to maintain stable pH values even in the face of (slight levels of) contamination.

pH buffers may be purchased in liquid form or in powder form. Liquid buffer solutions may be used directly out of the bottle, while powdered buffers must be dissolved in appropriate quantities of de-ionized water to generate a solution ready for calibration use. Pre-mixed liquid buffers are convenient to use, but have a fairly limited shelf life. Powdered buffer capsules are generally superior for long-term storage, and also enjoy the advantage of occupying less storage space in their dry state than a liquid buffer solution.

The following photograph shows a few 7.00 pH (+/- 0.02 pH) buffer capsules ready to be mixed with water to form a usable buffer solution:

[8] For example, a solution with a pH value of 4.7 has a concentration of $10^{-4.7}$ moles of active hydrogen ions per liter

Figure 8.24: pH calibration standard buffer capsules

After preparing the buffer solution in a cup, the pH probe is inserted into the buffer solution and given time to stabilize. One stabilized, the pH instrument may be adjusted to register the proper pH value. Buffer solutions should not be exposed to ambient air for any longer than necessary (especially alkaline buffers such as 10.0 pH) due to contamination[9]. Pre-mixed liquid buffer storage containers should be capped immediately after pouring into working cups. Used buffer solution should be discarded rather than re-used at a later date.

Analyzers designed to measure the concentration of certain gases in air must be calibrated in a similar manner. Oxygen analyzers, for example, used to measure the concentration of free oxygen in the exhaust gases of furnaces, engines, and other combustion processes must be calibrated against known standards of oxygen concentration. An oxygen analyzer designed to measure oxygen concentration over a range of ambient (20.9% oxygen) to 0% oxygen may be calibrated with ambient air as one of the standard values[10], and a sample of pure nitrogen gas (containing 0% oxygen) as the other standard value. An oxygen analyzer intended for the measurement of oxygen concentrations in excess of ambient air would require a different standard, most likely a sample of 100% pure oxygen, as a calibration reference.

An analyzer designed to measure the concentration of hydrogen sulfide ($H_2S$), a toxic gas produced by anaerobic bacterial decomposition of organic matter, will require a sample of gas with a precisely known concentration of hydrogen sulfide mixed in it as a calibration reference. A typical

[9]Carbon dioxide gas in ambient air will cause carbonic acid to form in an aqueous solution. This has an especially rapid effect on high-pH (alkaline) buffers.

[10]It is assumed that the concentration of oxygen in ambient air is a stable enough quantity to serve as a calibration standard for most industrial applications. It is certainly an *accessible* standard!

reference gas concentration might be 25 or 50 parts per million (ppm). Gas mixtures with such precise concentration values as this may be purchased from chemical laboratories for the purpose of calibrating concentration analyzers, and are often referred to as *span gases* because they are used to set the span of analyzer instruments.

Analytical instruments are generally subject to greater drifting over time than instruments that measure incidental quantities such as pressure, level, temperature, or flow rate. It is not uncommon for instrument technicians to be tasked with *daily* calibration checks of certain instruments responsible for monitoring atmospheric or water emissions at industrial facilities. For this reason, it is often practical to equip such critical analyzers with *self-calibration* systems. A self-calibration system is a system of solenoid (electrically controlled on-off) valves and reference gas bottles set up in such a way that a computer is able to switch the analyzer off-line and subject it to standard reference gases on a regular schedule to check calibration. Many analyzers are programmed to automatically calibrate themselves against these reference gases, thus eliminating tedious work for the instrument technician.

A typical self-calibration system for a gas analyzer might look like this:

Figure 8.25: Diagram of a calibration setup for a gas analyzer

The gas analyzer is equipped with its own auto-calibration controls and programming, allowing it to periodically shut off the process sample and switch to known reference gases for "zero" and "span"

calibration checks. If these checks indicate excessive drift or any other questionable results, the analyzer has the ability to flag a maintenance alarm to alert an instrument technician to a potential problem that may require servicing. This sort of self-calibration and self-diagnostic capability saves the instrument technician from having to spend substantial time running manual calibration checks, yet alerts the technician if anything is in need of actual repair. Barring any component failures within this system, the only maintenance this system will need is periodic replacement of the calibration gas bottles.

# References


[1] "Calibration - philosophy in practice," Everett, WA, 1994.

[2] J. L. Bucher, The Quality Calibration Handbook. Milwaukee, WI: ASQ Quality Press, 2007.

[3] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1.

# Chapter 9

# Discrete Event Detection

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: discrete measurement, normal switch status, hand switch, proximity switch, limit switch
2. Identify the normal state for various types of switches
3. Describe the difference between a limit switch and a proximity switch
4. Classify a switch by its number of poles, throws, and breaks
5. Classify a switch by its method of actuation
6. Differentiate between the following types of contacts: a, b, c

Figure 9.1: A normally-open momentary pushbutton switch

The word "discrete" means *individual* or *distinct*. In engineering, a "discrete" variable or measurement refers to a true-or-false condition. Thus, a discrete sensor is one that is only able to indicate whether the measured variable is above or below a specified setpoint.

Discrete sensors typically take the form of *switches*, built to "trip" when the measured quantity either exceeds or falls below a specified value. These devices are less sophisticated than so-called *continuous* sensors capable of reporting an analog value, but they are quite useful in industry.

Many different types of discrete sensors exist, detecting variables such as position, fluid pressure, material level, temperature, and fluid flow rate. These are discussed in their respective chapters later in the text. The output of a discrete sensor is typically electrical in nature, whether it be an active voltage signal or just resistive continuity between two terminals on the device.

## 9.1 Status of a switch

Perhaps the most confusing aspect of discrete sensors is the definition of a sensor's *normal* status. Electrical switch contacts are typically classified as either *normally-open* or *normally-closed*, referring to the open or closed status of the contacts under "normal" conditions. But what exactly defines "normal" for a switch? The answer is not complex, but it is often misunderstood.

The "normal" status for a switch is the status its electrical contacts are in *under a condition of no physical stimulation*. Another way to think of the "normal" status is to think in terms of the switch being *at rest*. For a momentary-contact pushbutton switch, this would be the status of the switch contact when it is *not* being pressed. The "normal" status of any switch is the way it must be drawn in an electrical schematic per IEEE Standard 315. For instance, Figure 9.1 shows a normally-open pushbutton switch controlling a lamp on a 120 volt AC circuit (the "hot" and "neutral" poles of the AC power source labeled L1 and L2, respectively).

This switch is a normally-open (NO) switch because it is drawn in an open position. The lamp will energize only if someone presses the switch, holding its normally-open contacts in the "closed" position. Normally-open switch contacts are sometimes referred to in the electrical industry as *form-A* contacts.

If a normally-closed pushbutton switch was used instead, the behavior would be exactly opposite. The lamp would energize if the switch was left alone, but it would turn off if anyone pressed the switch. Normally-closed switch contacts are sometimes referred to in the electrical industry as *form-B* contacts . An example of this type of switch is shown in Figure 9.2.

This seems rather simple. What could possibly be confusing about the "normal" status of a switch? The confusion becomes evident, though, when considering the case of a different kind of discrete sensor such as a flow switch.

Figure 9.2: A normally-closed momentary pushbutton switch

Figure 9.3: A normally-closed flow switch that opens when flow is above its trip setpoint

A flow switch is built to detect fluid flow through a pipe. In a schematic diagram, the switch symbol appears to be a toggle switch with a "flag" hanging below. The schematic diagram, of course, only shows the circuitry and not the pipe where the switch is physically mounted. An example schematic is shown in Figure 9.3.

This particular flow switch is used to trigger an alarm light if coolant flow through the pipe ever falls to a dangerously low level, and the contacts are *normally-closed* as evidenced by the closed status in the diagram. Here is where things get confusing: even though this switch is designated as "normally-closed," it will spend most of its lifetime being held in the open status by the presence of adequate coolant flow through the pipe. Only when the flow through the pipe slows down enough will this switch return to its "normal" status (remember, the resting condition means *no stimulation*) and conduct electrical power to the lamp. In other words, the "normal" status of this switch (closed) is actually an *abnormal* status for the process it is sensing (low flow)!

Students often wonder why process switch contacts are labeled according to this convention of "no stimulation" instead of according to the typical status of the process in which the switch is used. The answer to this question is that the manufacturer of the sensor has no idea whatsoever as to the intended use. The manufacturer of the switch does not know and does not care if the intended use of their flow switch is as a low-flow alarm or as a high-flow alarm. In other words, the manufacturer cannot predict what the typical status of a process will be, and so the definition of "normal" status for the switch must be founded on some common criterion unrelated to the particular application. That common criterion is the resting status: when the sensor is exposed to the *least* (or no) amount of stimulation from the process it senses.

Here is a listing of "normal" definitions for various discrete sensor types:

- **Hand switch**: no one pressing the switch
- **Limit switch**: target not contacting the switch
- **Proximity switch**: target far away
- **Pressure switch**: low pressure (or even a vacuum)

Figure 9.4: A momentary hand switch in the normal position

- **Level switch**: low level (empty)
- **Temperature switch**: low temperature (cold)
- **Flow switch**: low flow rate (fluid stopped)

These are the conditions represented by the switch statuses shown in a schematic diagram. These may very well *not* be the statuses of the switches when they are exposed to *typical* operating conditions in the process.

## 9.2 Actuation Method

### 9.2.1 Hand switches

A *hand switch* is exactly what the name implies: an electrical switch actuated by a person's hand motion. These may take the form of toggle, pushbutton, rotary, pull-chain, etc. A common form of industrial pushbutton switch is shown in Figure 9.4.

The threaded neck inserts through a hole cut into a metal or plastic control panel, with a matching nut to hold it in place. Thus, the button faces the human operator(s) while the energized switch terminals and contacts reside on the inside of the panel.

When pressed, the downward motion of the actuator breaks the electrical bridge between the two NC contacts, forming a new bridge between the NO contacts as shown in Figure 9.5.

The schematic diagram symbol for this type of switch looks much like the real thing, with the normally-closed contact set on top and the normally-open contact set below:

Connecting two of the terminals together makes this form of switch electrically identical to a *Form C* contact set:

This switch contact arrangement is referred to as a *form-C* contact set since it incorporates both a form-A contact (normally-open) as well as a form-B contact (normally-closed) [3].

Most industrial hand switches are available in modular form, where sets of switch contact blocks may be "stacked" together to be actuated by the same pushbutton or rotary knob:

Figure 9.5: A momentary hand switch in the activated position

Figure 9.6: Schematic symbol of a double-pole, double throw momentary pushbutton switch

Figure 9.7: Schematic symbol of a Form C momentary pushbutton switch

### 9.2.2 Limit switches

A *limit switch* detects the physical motion of an object by direct contact with that object. An example of a limit switch is the switch detecting the open position of an automobile door, automatically energizing the cabin light when the door opens.

Recall from section 9.1 that the "normal" status of a switch is the resting condition of *no stimulation*. A limit switch will be in its "normal" status when it is not in contact with anything (i.e., nothing touching the switch actuator mechanism).

Limit switches find many uses in industry, particular in robotic control and CNC (Computer Numerical Control) machine tool systems. In many motion-control systems, the moving elements have "home" positions where the computer assigns a position value of zero. For example, the axis controls on a CNC machine tool such as a lathe or mill all return to their "home" positions upon start-up, so the computer can know with confidence the starting locations of each piece. These home positions are detected by means of limit switches. The computer commands each servo motor to travel fully in one direction until a limit switch on each axis trips. The position counter for each axis resets to zero as soon as the respective limit switch detects that the home position has been reached.

A typical limit switch design uses a roller-tipped lever to make contact with the moving part. Screw terminals on the switch body provide connection points with the NC and NO contacts inside

the switch. Most limit switches of this design share a "common" terminal between the NC and NO contacts like this:

This switch contact arrangement is sometimes referred to as a *form-C* contact set, since it incorporates both a form-A contact (normally-open) as well as a form-B contact (normally-closed).

A close-up view of several limit switches (used on a drum sequencer) shows the arrangement of connection terminals for form-C contacts. Each limit switch has its own "NO" (normally-open), "NC" (normally-closed), and "C" (common) screw terminal for wires to attach:

A limit switch assembly attached to the stem of a rotary valve – used to detect the fully-closed and fully-open positions of the valve – is shown in the following photograph:

### 9.2.3 Proximity switches

A *proximity switch* is one detecting the proximity (closeness) of some object. By definition, these switches are *non-contact sensors*, using magnetic, electric, or optical means to sense the proximity of objects.

Recall from section 9.1 that the "normal" status of a switch is the resting condition of *no stimulation*. A proximity switch will be in its "normal" status when it is distant from any actuating object.

Being non-contact in nature, proximity switches are often used instead of direct-contact limit switches for the same purpose of detecting the position of a machine part, with the advantage of never wearing out over time due to repeated physical contact. However, the greater complexity (and cost) of a proximity switch over a mechanical limit switch relegates their use to applications where lack of physical contact yields tangible benefits.

Most proximity switches are *active* in design. That is, they incorporate a powered electronic circuit to sense the proximity of an object. *Inductive* proximity switches sense the presence of metallic objects through the use of a high-frequency magnetic field. *Capacitive* proximity switches sense the presence of non-metallic objects through the use of a high-frequency electric field. *Optical* proximity switches detect the interruption of a light beam by an object. *Ultrasonic* proximity switches sense the presence of dense matter by the reflection of sound waves.

The schematic diagram symbol for a proximity switch with mechanical contacts is the same as for a mechanical limit switch, except the switch symbol is enclosed by a diamond shape, indicating a powered (active) device:

Many proximity switches, though, do not provide "dry contact[1]" outputs. Instead, their output elements are transistors configured either to *source* current or *sink* current. The terms "sourcing" and "sinking" are best understood by visualizing electric current in the direction of *conventional flow* rather than *electron flow*.

The following schematic diagrams contrast the two modes of switch operation, using red arrows to show the direction of current (conventional flow notation). In both examples, the load being driven by each proximity switch is a light-emitting diode (LED):

[1]This curious label is used to describe mechanical switch contacts, as opposed to solid-state (transistor) switching elements. Any device with one or more "dry contact" outputs is therefore able to switch virtually any kind of signal, from DC to AC to even high-frequency pulses. The major disadvantages of dry contact outputs are (relatively) slow response time, mechanical wear, and contact bounce.

An electronic switch designed to sink current through its signal wire is alternatively referred to as an *NPN* switch due to the type of transistor used in its output. Conversely, an electronic switch designed to source current through its signal wire may be referred to as a *PNP* switch. The key to understanding these labels is to recognize the emitter terminal of the output transistor is always the one connected to the power supply rail. For a sinking switch, this means the emitter must connect to the negative rail, necessitating[2] an NPN transistor to do the switching. For a sourcing switch, this means the emitter must connect to the positive rail, in which case only a PNP transistor will suffice.

Yet another convention for describing sourcing and sinking transistor switches is to refer to them as *high-side* switches and *low-side* switches, respectively. A sourcing transistor (PNP) has

[2]To be honest, one *could* use an NPN transistor to source current or a PNP to sink, but it would require the transistor be used in the *common-collector* configuration which does not allow for saturation. The engineers designing these proximity switches strive for complete saturation of the transistor, in order to achieve minimum "on" resistance, and that requires a common-emitter configuration.

its emitter terminal attached directly to the "high" rail (+) of the DC power supply. A sinking transistor (NPN), by contrast, has its emitter terminal attached directly to the "low" rail (-) of the DC power supply.

These photographs show two different styles of electronic proximity switch:

Many industrial proximity switches have built-in LED indicator lamps to help technicians diagnose circuit problems by directly indicating switch status. With just a glance, one may tell whether or not the proximity switch is detecting the presence of an object.

The next photograph shows a proximity switch detecting the passing of teeth on a chain sprocket, generating a slow square-wave electrical signal as the sprocket rotates. Such a switch may be used as a rotational speed sensor (sprocket speed proportional to signal frequency) or as a broken chain sensor (when sensing the rotation of the driven sprocket instead of the drive sprocket):

Like other types of process switches, proximity switches come in both "normally open" (NO) and "normally closed" (NC) varieties. This distinction has nothing whatsoever to do with sourcing

versus sinking (PNP vs. NPN), but rather what the status of the proximity switch will be when no objects are near. Thus, it is possible to find normally-open proximity switches that are sinking (NPN) as well as normally-open proximity switches that are sourcing (PNP), and normally-closed proximity switches in either sinking or sourcing designs as well.

These switch characteristics are commonly fixed, and must be specified when ordering the device. Likewise, the detection range of a proximity switch is usually a fixed parameter rather than being adjustable.

# References


[1] Installation and Service Manual - 1500 Induction Control Relays, 511 1500.M4R 02/05.Z145 10M, AMETEK Automation and Process Technologies, Clawson, MI, 2005.

[2] Installation and Service Manual - 5200 Solid State Relays, 432 5200.M4R 05/07.Z152, AMETEK Automation and Process Technologies, Clawson, MI, 2007.

[3] IEEE Graphic Symbols for Electrical and Electronics Diagrams (Including Reference Designation Letters), Institute of Electrical and Electronics Engineeers Std. 315, 1975.

# Chapter 10

# Pressure Measurement

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Discuss the basic functions of pressure detectors.
2. Draw P&ID symbols used for pressure and differential pressure detection equipment, including sensing elements, transmitters, and protective diaphragms.
3. Explain how the following types of pressure detectors produce an output representative of the measured pressure: Bourdon tube, bellows, diaphragm, manometer
4. Describe the environmental concerns that can affect the accuracy and reliability of pressure detection instrumentation.
5. Describe the function of the following electrical pressure elements: piezoresistive, differential capacitance, resonant
6. Explain how a force-balance pressure transmitter works
7. Discuss the failure modes of the various types of pressure indication instruments.
8. Discuss and sketch methods of pressure sensor installation, including accessories that may be required, and describe advantages and disadvantages of various installations.

In many ways, pressure is the primary variable for a wide range of process measurements. Many types of industrial measurements are actually inferred from pressure, such as:

- Flow (measuring the pressure dropped across a restriction)
- Liquid level (measuring the pressure created by a vertical liquid column)
- Liquid density (measuring the pressure difference across a fixed-height liquid column)
- Weight (hydraulic load cell)

Even temperature may be inferred from pressure measurement, as in the case of a fluid-filled chamber where fluid pressure and fluid temperature are directly related. As such, pressure is a very important quantity to measure, and measure accurately. This section describes different technologies for the measurement of pressure.

## 10.1 Manometers

A very simple device used to measure pressure is the *manometer*: a fluid-filled tube where an applied gas pressure causes the fluid height to shift proportionately. This is why pressure is often measured in units of liquid height (e.g. inches of water, inches of mercury). As can be seen, a manometer is fundamentally an instrument of *differential* pressure measurement, indicating the difference between two pressures by a shift in liquid column height:

Figure 10.1: A U-tube manometer

Of course, it is entirely acceptable to simply vent one tube of a manometer and use it as a *gauge* pressure instrument, comparing the applied pressure at one tube against atmospheric pressure in the other.

Liquid column height in a manometer should always be interpreted at the centerline of the liquid column, regardless of the shape of the liquid's meniscus (the curved air/liquid interface):

Figure 10.2: Manometer meniscus readings

Manometers come in a variety of forms, the most common being the *U-tube*, *well* (sometimes called a *cistern*), *raised well*, and *inclined*:

Figure 10.3: Comparison of manometer types

Figure 10.4: Additional manometer types

U-tube manometers are very inexpensive, and are generally made from clear plastic (see the left-hand photo). Cistern-style manometers are the norm for calibration bench work, and are typically constructed from metal cisterns and glass tubes (see the right-hand photo):

Inclined manometers are used to measure very low pressures, owing to their exceptional sensitivity (note the fractional scale for inches of water column in the following photograph, extending from 0 to 1.5 inches on the scale, reading left to right):

Figure 10.5: An inclined manometer

Note that venting one side of a manometer is standard practice when using is as a *gauge pressure* indicator (responding to pressure in excess of atmospheric). Both pressure ports will be used if the manometer is applied to the measurement of differential pressure, just as in the case of the U-tube manometer first shown in this section. Absolute pressure may also be measured by a manometer, if one of the pressure ports connects to a sealed vacuum chamber. This is how a *mercury barometer* is constructed for the measurement of absolute ambient air pressure: by sealing off one side of a manometer and removing all the air in that side, such that the applied (atmospheric) pressure is always compared against a vacuum.

Manometers incorporating a "well" have the advantage of single-point reading: one need only compare the height of *one* liquid column, not the difference in height between *two* liquid columns. The cross-sectional area of the liquid column in the well is so much greater than that within the transparent manometer tube that the change in height within the well is usually negligible. In cases where the difference is significant, the spacing between divisions on the manometer scale may be skewed to compensate[1].

Inclined manometers enjoy the advantage of increased sensitivity. Since manometers fundamentally operate on the principle of pressure balanced by liquid height, and this liquid height is always measured parallel to the line of gravitational pull (perfectly vertical), inclining the manometer tube means that liquid must travel further along the tube to generate the same change in (purely) vertical height than it would in a vertical manometer tube. Thus, an inclined manometer tube causes

[1] If this concept is difficult to understand, imagine a simple U-tube manometer where one of the tubes is opaque, and therefore one of the two liquid columns cannot be seen. In order to be able to measure pressure just by looking at one liquid column height, a custom scale must be made where every inch of height registered as *two* inches of water column pressure, because for each inch of height change in the liquid column that is visible, the invisible liquid column also changes by an inch. A scale custom-made for a well-type manometer is just the same concept, only without such dramatic skewing of scales.

an amplification in liquid motion for a given amount of pressure change, allowing measurements of greater resolution.

## 10.2 Mechanical pressure elements

Mechanical pressure-sensing elements include the *bellows*, the *diaphragm*, and the *bourdon tube*. Each of these devices converts a fluid pressure into a force. If unrestrained, the natural elastic properties of the element will produce a motion proportional to the applied pressure.

Figure 10.6: Pressure sensing elements

Bellows resemble an accordion constructed from metal instead of fabric. Increasing pressure inside a bellows unit causes it to elongate. A photograph of a bellows is shown here:

Figure 10.7: A bellows used for sensing pressure

A diaphragm is nothing more than a thin disk of material which bows outward under the influence of a fluid pressure. Many diaphragms are constructed from metal, which gives them spring-like qualities. Some diaphragms are intentionally constructed out of materials with little strength, such that there is negligible spring effect. These are called *slack diaphragms*, and they are used in conjunction with external mechanisms that produce the necessary restraining force to prevent damage from applied pressure.

The following photograph shows the mechanism of a small pressure gauge using a brass diaphragm as the sensing element:

Figure 10.8: A diaphragm used for sensing pressure

As pressure is applied to the rear of the diaphragm, it distends upward (away from the table on which it rests as shown in the photograph), causing a small shaft to twist in response. This twisting motion is transferred to a lever which pulls on a tiny link chain wrapped around the pointer shaft, causing it to rotate and move the pointer needle around the gauge scale. Both the needle and scale on this gauge mechanism have been removed for easier viewing of diaphragm and mechanism.

Bourdon tubes are made of spring-like metal alloys bent into a circular shape. Under the influence of internal pressure, a bourdon tube "tries" to straighten out into its original shape before being bent at the time of manufacture.

Most pressure gauges use a Bourdon tube as their pressure-sensing element. Most pressure transmitters use a diaphragm as their pressure-sensing element. Bourdon tubes may be made in *spiral* or *helical* forms for greater motion (and therefore greater gauge resolution).

A typical C-shaped Bourdon tube pressure gauge mechanism is shown in the following illustration:

Figure 10.9: A diagram of a Bourdon tube used for sensing pressure

A photograph of a C-tube pressure gauge mechanism (taken from the rear of the gauge, behind the pointer and scale) reveals its mechanical workings:

Figure 10.10: A Bourdon tube used for sensing pressure

The dark, C-shaped tube is the bourdon tube sensing element, while the shiny metal parts are the linkage, lever, and gear assembly.

The next photograph shows a *spiral* Bourdon tube, designed to produce a wider range of motion than a C-tube Bourdon:

Figure 10.11: A spiral Bourdon tube used for sensing pressure

It should be noted that bellows, diaphragms, and Bourdon tubes alike may all be used to measure differential and/or absolute pressure in addition to gauge pressure. All that is needed for these other functionalities is to subject the *other* side of each pressure-sensing element to either another applied pressure (in the case of differential measurement) or to a vacuum chamber (in the case of absolute pressure measurement).

This next set of illustrations shows how bellows, diaphragms, and Bourdon tubes may be used as differential pressure-sensing elements:

Figure 10.12: Differential pressure sensing elements

The challenge in doing this, of course, is how to extract the mechanical motion of the pressure-sensing element to an external mechanism (such as a pointer) while maintaining a good pressure seal. In gauge pressure mechanisms, this is no problem because one side of the pressure-sensing element must be exposed to atmospheric pressure anyway, and so that side is always available for mechanical connection.

A differential pressure gauge is shown in the next photograph. The two pressure ports are clearly evident on either side of the gauge:

Figure 10.13: A differential pressure gauge

## 10.3 Electrical pressure elements

Several different technologies exist for the conversion of fluid pressure into an electrical signal response. These technologies form the basis of electronic *pressure transmitters*: devices designed to measure fluid pressure and transmit that information via electrical signals such as the 4-20 mA analog standard, or in digital form such as HART or FOUNDATION Fieldbus.

A brief survey of electronic pressure transmitters in contemporary use reveals a diverse representation of electrical pressure-sensing elements:

| Manufacturer | Model | Pressure sensor technology |
|---|---|---|
| ABB/Bailey | PTSD | Differential reluctance |
| ABB/Bailey | PTSP | Piezoresistive (strain gauge) |
| Foxboro | IDP10 | Piezoresistive (strain gauge) |
| Honeywell | ST3000 | Piezoresistive (strain gauge) |
| Rosemount | 1151 | Differential capacitance |
| Rosemount | 3051 | Differential capacitance |
| Rosemount | 3095 | Differential capacitance |
| Yokogawa | EJX series | Mechanical resonance |

### 10.3.1 Piezoresistive (strain gauge) sensors

*Piezoresistive* means "pressure-sensitive resistance," or a resistance that changes value with applied pressure. The *strain gauge* is a classic example of a piezoresistive element:

Figure 10.14: Diagram of a strain gage

As the test specimen is stretched or compressed by the application of force, the conductors of

the strain gauge are similarly deformed. Electrical resistance of any conductor is proportional to the ratio of length over cross-sectional area ($R \propto \frac{l}{A}$), which means that tensile deformation (stretching) will increase electrical resistance by simultaneously increasing length and decreasing cross-sectional area while compressive deformation (squishing) will decrease electrical resistance by simultaneously decreasing length and increasing cross-sectional area.

Attaching a strain gauge to a diaphragm results in a device that changes resistance with applied pressure. Pressure forces the diaphragm to deform, which in turn causes the strain gauge to change resistance. By measuring this change in resistance, the amount of pressure applied to the diaphragm can be inferred.

The classic strain gauge system represented in the previous illustration is made of metal (both the test specimen and the strain gauge itself). Within its elastic limits, many metals exhibit good spring characteristics. Metals, however, are subject to *fatigue* over repeated cycles of strain (tension and compression), and they will begin to "flow" if strained beyond their elastic limit. This is a common source of error in metallic piezoresistive pressure instruments: if overpressured, they tend to lose accuracy due to damage of the spring and strain gauge elements.[2]

Modern manufacturing techniques have made possible the construction of strain gauges made of silicon instead of metal. Silicon exhibits very linear spring characteristics over its narrow range of motion, and a high resistance to fatigue. When a silicon strain gauge is over-stressed, it fails completely rather than "flows" as is the case with metal strain gauges. This is generally considered a better result, as it clearly indicates the need for sensor replacement (whereas a metallic strain sensor may give the false impression of continued function after an over-stress event).

Thus, most modern piezoresistive-based pressure instruments use silicon strain gauge elements to sense deformation of a diaphragm due to applied fluid pressure. A simplified illustration of a diaphragm / strain gauge pressure sensor is shown here:

[2]For a simple demonstration of metal fatigue and metal "flow," simply take a metal paper clip and repeatedly bend it back and forth until the metal wire weakens. Gentle force applied to the paper clip will cause it to deform in such a way that it returns to its original shape when the force is removed. Greater force, however, will exceed the paper clip's elastic limit, causing permanent deformation and also altering the spring characteristics of the clip.

Figure 10.15: Diagram of a strain gage used with a diaphragm to sense pressure

In some designs, a single silicon wafer serves as both the diaphragm and the strain gauge so as to fully exploit the excellent mechanical properties of silicon (high linearity and low fatigue). However, silicon is not chemically compatible with many process fluids, and so pressure must be transferred to the silicon diaphragm/sensor via a non-reactive *fill fluid* (commonly a silicone-based or fluorocarbon-based liquid). A metal *isolating diaphragm* transfers process fluid pressure to the fill fluid. Another simplified illustration shows how this works:

Figure 10.16: Diagram of a strain gage used with an isolating diaphragm to sense pressure

The isolating diaphragm is designed to be much more flexible (less rigid) than the silicon diaphragm, because its purpose is to seamlessly transfer fluid pressure from the process fluid to the fill fluid, not to act as a spring element. In this way, the silicon sensor experiences the same pressure that it would if it were directly exposed to the process fluid, without having to contact the process fluid.

An example of a pressure instrument utilizing a silicon strain gauge element is the Foxboro model IDP10 differential pressure transmitter, shown in the following photograph:

Figure 10.17: An industrial pressure transmitter

### 10.3.2 Differential capacitance sensors

Another common electrical pressure sensor design works on the principle of *differential capacitance*. In this design, the sensing element is a taut metal diaphragm located equidistant between two stationary metal surfaces, forming a complementary pair of capacitances. An electrically insulating fill fluid (usually a liquid silicone compound) transfers motion from the isolating diaphragms to the sensing diaphragm, and also doubles as an effective dielectric for the two capacitors:

Figure 10.18: Diagram of a differential capacitance pressure sensor

Any difference of pressure across the cell will cause the diaphragm to flex in the direction of least pressure. The sensing diaphragm is a precision-manufactured spring element, meaning that its displacement is a predictable function of applied force. The applied force in this case can only be a function of differential pressure acting against the surface area of the diaphragm in accordance with the standard force-pressure-area equation $F = PA$. In this case, there are two forces caused by two fluid pressures working against each other, so the force-pressure-area equation may be re-written to describe *resultant* force as a function of differential pressure $(P_1 - P_2)$ and diaphragm area: $F = (P_1 - P_2)A$. Since diaphragm area is constant, and force is predictably related to diaphragm displacement, all that is needed now in order to infer differential pressure is to accurately measure displacement of the diaphragm.

The diaphragm's secondary function as one plate of two capacitors provides a convenient method for measuring displacement. Since capacitance between conductors is inversely proportional to the distance separating them, capacitance on the low-pressure side will increase while capacitance on the high-pressure side will decrease:

Figure 10.19: Diagram of a capacitance pressure sensor exposed to a pressure differential

A capacitance detector circuit connected to this cell uses a high-frequency AC excitation signal to measure the different in capacitance between the two halves, translating that into a DC signal which ultimately becomes the signal output by the instrument representing pressure.

These pressure sensors are highly accurate, stable, and rugged. The solid frame bounds the motion of the two isolating diaphragms such that the sensing diaphragm cannot move past its elastic limit. This gives the differential capacitance excellent resistance to overpressure damage.

A classic example of a pressure instrument based on the differential capacitance sensor is the Rosemount model 1151 differential pressure transmitter, shown in assembled form in the following photograph:

Figure 10.20: An industrial differential capacitance pressure transmitter

By removing four bolts from the transmitter, two flanges are able to removed from the pressure capsule, exposing the isolating diaphragms to plain view:

Figure 10.21: An industrial pressure transmitter with exposed isolating diaphragms

A close-up photograph shows the construction of one of the isolating diaphragms, which unlike the sensing diaphragm is designed to be very flexible. The concentric corrugations in the metal of the diaphragm allow it to easily flex with applied pressure, transmitting process fluid pressure through the silicone fill fluid to the taut sensing diaphragm inside the differential capacitance cell:

Figure 10.22: A close-up of an isolating diaphragm

The differential capacitance sensor inherently measures *differences* in pressure applied between its two sides. In keeping with this functionality, this pressure instrument has two threaded ports into which fluid pressure may be applied. A later section in this chapter will elaborate on the utility of differential pressure transmitters (section 10.5 beginning on page 315).

All the electronic circuitry necessary for converting the sensor's differential capacitance into an electronic signal representing pressure is housed in the blue-colored structure above the capsule and flanges.

A more modern realization of the differential capacitance pressure-sensing principle is the Rosemount model 3051 differential pressure transmitter:

Figure 10.23: A more modern industrial differential pressure transmitter

As is the case for all differential pressure devices, this instrument has two ports through which fluid pressure may be applied to the sensor. The sensor, in turn, responds only to the *difference* in pressure between the ports.

The differential capacitance sensor construction is more complex in this particular pressure instrument, with the plane of the sensing diaphragm lying perpendicular to the plane of the two isolating diaphragms. This "coplanar" design is far more compact than the older style of sensor, and it isolates the sensing diaphragm from flange bolt stress – one of the main sources of error in the previous design[3].

[3]Not only did applied torque of the four capsule bolts affect measurement accuracy in the older 1151 model design, but changes in temperature resulting in changing bolt tension also had a detrimental impact on accuracy. Most modern differential pressure transmitter designs strive to isolate the sensing diaphragm assembly from flange bolt stress for these reasons.

### 10.3.3 Resonant element sensors

As any guitarist, violinist, or other stringed-instrument musician knows, the natural frequency of a tensed string increases with tension. This, in fact, is how stringed instruments are tuned: the tension on each string is precisely adjusted to achieve the desired resonant frequency.

Mathematically, the resonant frequency of a string may be described by the following formula:

$$f = \frac{1}{2L}\sqrt{\frac{F_T}{\mu}}$$

Where,

$f$ = Fundamental resonant frequency of string (Hertz)
$L$ = String length (meters)
$F_T$ = String tension (newtons)
$\mu$ = Unit mass of string (kilograms per meter)

It stands to reason, then, that a string may serve as a force sensor. All that is needed to complete the sensor is an oscillator circuit to keep the string vibrating at its resonant frequency, and that frequency becomes an indication of tension (force). If the force stems from pressure applied to some sensing element such as a bellows or diaphragm, the string's resonant frequency will indicate fluid pressure. A proof-of-concept device based on this principle might look like this:

Figure 10.24: Diagram of a mechanical resonant pressure sensor

The Foxboro company pioneered this concept in an early *resonant wire* design of pressure transmitter. Later, the Yokogawa corporation of Japan applied the concept to a pair of micro-machined[4] silicon resonator structures, which became the basis for their successful line of "DPharp" pressure transmitters.

A photograph of a Yokogawa model EJA110 pressure transmitter with this technology is seen here:

[4]This is an example of a micro-electro-mechanical system, or *MEMS*.

Figure 10.25: An industrial pressure transmitter with a mechanical resonant pressure sensor

Process pressure enters through ports in two flanges, presses against a pair of isolating diaphragms, transferring motion to the sensing diaphragm where the resonant elements change frequency with diaphragm strain. Electronic circuits within the upper housing measure the two resonant elements' frequencies and generate an output signal proportional to their frequency difference. This, of course, is a representation of applied differential pressure.

Even when disassembled, the transmitter does not look much different from the more common differential capacitance sensor design.

Figure 10.26: An industrial pressure transmitter with a flanges removed

The important design differences are hidden from view, inside the sensing capsule. Functionally, though, this transmitter is much the same as its differential-capacitance cousin.

An interesting advantage of the resonant element pressure sensor is that the sensor signal is very easy to digitize. The vibration of each resonant element is sensed by the electronics package as an AC frequency. Any frequency signal may be easily "counted" over a given span of time and converted to a binary digital representation. Quartz crystal electronic oscillators are extremely precise, providing the stable frequency reference necessary for comparison in any frequency-based instrument.

In the Yokogawa "DPharp" design, the two resonant elements oscillate at a nominal frequency of 90 kHz. As the sensing diaphragm deforms with applied differential pressure, one resonator experiences tension while the other experiences compression, causing the frequency of the former to shift up and the latter to shift down (as much as +/- 20 kHz). The signal conditioning electronics inside the transmitter measures this difference in resonator frequency to infer applied pressure.

### 10.3.4 Mechanical adaptations

Most modern electronic pressure sensors convert very small diaphragm motions into electrical signals through the use of sensitive motion-sensing techniques (strain gauge sensors, differential capacitance cells, etc.). Diaphragms made from elastic materials behave as springs, but circular diaphragms exhibit very nonlinear behavior when significantly stretched unlike classic spring designs such as coil and leaf springs which exhibit linear behavior over a wide range of motion. Therefore, in order to yield a linear response to pressure, a diaphragm-based pressure sensor must be designed in such a way that the diaphragm stretches very little over the normal range of operation. Limiting the displacement of a diaphragm necessitates highly sensitive motion-detection techniques such as

strain gauge sensors, differential capacitance cells, and mechanical resonance sensors to convert that diaphragm's very slight motion into an electronic signal.

An alternative approach to electronic pressure measurement is to use mechanical pressure-sensing elements with more linear pressure-displacement characteristics – such as Bourdon tubes and spring-loaded bellows – and then detect the large-scale motion of the pressure element using a less-sophisticated electrical motion-sensing device such as a potentiometer, LVDT, or Hall Effect sensor. In other words, the sort of mechanism commonly found in a direct-reading pressure gauge may be attached to a potentiometer (or similar device) so that an electrical signal from the pressure measurement be derived.

The following photographs show front and rear views of an electronic pressure transmitter using a large C-shaped Bourdon tube as the sensing element (seen in the left-hand photograph):

This alternative approach is undeniably simpler and less expensive to manufacture than the more sophisticated approaches used with diaphragm-based pressure instruments, but is prone to greater inaccuracies. Even Bourdon tubes and bellows are not perfectly linear spring elements, and the substantial motions involved with using such pressure elements introduces the possibility of hysteresis errors (where the instrument does not respond accurately during reversals of pressure, where the mechanism changes direction of motion) due to mechanism friction, and deadband errors due to backlash (looseness) in mechanical connections.

This sort of pressure instrument design is often encountered in direct-reading gauges equipped with electronic transmitting capability. An instrument manufacturer will take a proven product line of pressure gauge and add a motion-sensing device to it that generates an electric signal proportional to mechanical movement inside the gauge, resulting in an inexpensive pressure transmitter that happens to double as a direct-reading pressure gauge.

## 10.4 Force-balance pressure transmitters

An important legacy technology for all kinds of continuous measurement is the *self-balancing system*. A "self-balance" system continuously balances an adjustable quantity against a sensed quantity, the adjustable quantity becoming an indication of the sensed quantity once balance is achieved. A common manual-balance system is the type of scale used in laboratories to measure mass:

Figure 10.27: Diagram of a force balance

Here, the unknown mass is the sensed quantity, and the known masses are the adjustable quantity. A human lab technician applies as many masses to the left-hand side of the scale as needed to achieve balance, then counts up the sum total of those masses to determine the quantity of the unknown mass.

Such a system is perfectly linear, which is why these balance scales are popularly used for scientific work. The scale mechanism itself is the very model of simplicity, and the only thing the pointer needs to accurately sense is a condition of balance (equality between masses).

If the task of balancing is given to an automatic mechanism, the adjustable quantity will continuously change and adapt as needed to balance the sensed quantity, thereby becoming a representation of that sensed quantity. In the case of pressure instruments, pressure is easily converted into force by acting on the surface area of a sensing element such as a diaphragm or a bellows. A balancing force may be generated to exactly cancel the process pressure's force, making a *force-balance* pressure instrument. Like the laboratory balance scale, an industrial instrument built on the principle of balancing a sensed quantity with an adjustable quantity will be inherently linear, which is a tremendous advantage for measurement purposes.

Here, a diagram of a force-balance pneumatic pressure transmitter[5] is shown, balancing a sensed differential pressure with an adjustable air pressure which becomes a pneumatic output signal:

[5] Based on the design of Foxboro's popular model 13A pneumatic "DP cell" differential pressure transmitter.

Figure 10.28: Diagram of a pneumatic pressure transmitter that uses a force balance sensor

Differential pressure is sensed by a liquid-filled diaphragm "capsule," which transmits force to a "force bar." If the force bar moves out of position due to this applied force, a highly sensitive "baffle" and "nozzle" mechanism senses it and causes a pneumatic amplifier (called a "relay") to send a different amount of air pressure to a bellows unit. The bellows presses against the "range bar" which pivots to counter-act the initial motion of the force bar. When the system returns to equilibrium, the air pressure inside the bellows will be a direct, linear representation of the process fluid pressure applied to the diaphragm capsule.

With minor modifications to the design of this pressure transmitter[6], it may be converted from pneumatic to electronic force-balancing:

[6] Very loosely based on the design of Foxboro's popular E13 electronic "DP cell" differential pressure transmitter.

Figure 10.29: Diagram of an electronic pressure transmitter that uses a force balance sensor

Differential pressure is sensed by the same type of liquid-filled diaphragm capsule, which transmits force to the force bar. If the force bar moves out of position due to this applied force, a highly sensitive electromagnetic sensor detects it and causes an electronic amplifier to send a different amount of electric current to a force coil. The force coil presses against the range bar which pivots to counteract the initial motion of the force bar. When the system returns to equilibrium, the milliampere current through the force coil will be a direct, linear representation of the process fluid pressure applied to the diaphragm capsule.

A distinct advantage of force-balance pressure instruments (besides their inherent linearity) is the constraining of sensing element motion. Unlike a modern diaphragm-based pressure transmitter which relies on the spring characteristics of the diaphragm to convert pressure into force and then into motion (displacement) which is sensed and converted into an electronic signal, a force-balance transmitter works best when the diaphragm is slack and has no spring characteristics at all. Balance with the force of the process fluid pressure is achieved by the application of either an adjustable

air pressure or an adjustable electric current, not by the natural tensing of a spring element. This makes a force-balance instrument far less susceptible to errors due to metal fatigue or any other degradation of spring characteristics.

Unfortunately, force-balance instruments have significant disadvantages as well. Force-balance mechanisms tend to be bulky[7], and they translate external vibration into inertial force which adds "noise" to the output signal. Also, the amount of electrical power necessary to provide adequate balancing force in an electronic force-balance transmitter is such that it is nearly impossible to limit below the level necessary to ensure intrinsic safety (protection against the accidental ignition of explosive atmospheres by limiting the amount of energy the instrument could possibly discharge into a spark).

## 10.5 Differential pressure transmitters

One of the most common, and most useful, pressure measuring instruments in industry is the *differential pressure transmitter*. This device senses the difference in pressure between two ports and outputs a signal representing that pressure in relation to a calibrated range. Differential pressure transmitters may be based on any of the previously discussed pressure-sensing technologies, so this section discusses practical application rather than internal workings.

Differential pressure transmitters look something like this:

Figure 10.30: Diagram of differential pressure transmitters

Two models of electronic differential pressure transmitter are shown here, the Rosemount model 1151 (left) and model 3051 (right):

[7] One instrument technician has referred to the Foxboro E13 differential pressure transmitter as "pig iron" after having to hoist it by hand to the top of a distillation tower.

Two more models of electronic differential pressure transmitter are shown in the next photograph, the Yokogawa EJA110 (left) and the Foxboro IDP10 (right):

Regardless of make or model, every differential pressure ("DP", "d/p", or $\Delta$P)[8] transmitter has *two* pressure ports to sense different process fluid pressures. One of these ports is labeled "high" and the other is labeled "low". This labeling does not necessarily mean that the "high" port must always be at a greater pressure than the "low" port. What these labels represent is the effect that a pressure at that point will have on the output signal.

The concept of differential pressure instrument port labeling is very similar to the "inverting" and "noninverting" labels applied to operational amplifier input terminals:

[8] As far as is able to determined, the labels "D/P" and "DP cell" were originally trademarks of the Foxboro Company. Those particular transmitter models became so popular that the term "DP cell" came to be applied to nearly *all* makes and models of differential pressure transmitter, much like the trademark "Vise-Grip" is often used to describe *any* self-locking pliers, or "Band-Aid" is often used to describe *any* form of self-adhesive bandage.

Figure 10.31: An operational amplifier

The "+" and "-" symbols do not imply polarity of the input voltage(s). It is not as though the "+" input must be more positive than the "-" input. These symbols merely represent the different effects on the output signal that each input has. An increasing voltage applied to the "+" input drives the op-amp's output positive, while an increasing voltage applied to the "-" input drives the op-amp's output negative.

In a similar manner, an increasing pressure applied to the "high" port of a DP transmitter will drive the output signal to a greater level (up), while an increasing pressure applied to the "low" port of a DP transmitter will drive the output signal to a lesser level (down)[9]:

Figure 10.32: Diagram of an electronic differential pressure transmitter

We can use metal or plastic tubes (or pipes) to connect one or more ports of a pressure transmitter to points in a process. These tubes are commonly called *impulse lines*, or *gauge lines*, or *sensing lines*[10]. This is equivalent to the test wires used to connect a voltmeter to points in a circuit for measuring voltage. Typically, these tubes are connected to the transmitter and to the process by means of *compression fittings* which allow for relatively easy disconnection and reconnection of tubes.

The combination of two differential pressure ports makes the DP transmitter very versatile as a pressure-measuring device. We may use the DP transmitter to measure an actual difference of pressure across a fluid device such as a filter. Here, the amount of differential pressure across the filter represents how clogged the filter is:

[9] One transmitter manufacturer (ABB/Bailey) actually does use the "+" and "-" labels to denote high- and low-pressure ports rather than the more customary "H" and "L" labels found on other manufacturers' differential pressure products.

[10] Also called impulse *tubes*, gauge *tubes*, or sensing *tubes*.

Figure 10.33: Monitoring filter status using a differential pressure transmitter

Note how the high side of the DP transmitter connects to the upstream side of the filter, and the low side of the transmitter to the downstream side of the filter. This way, increased filter clogging will result in an increased transmitter output. Since the transmitter's internal pressure-sensing diaphragm only responds to *differences* in pressure between the "high" and "low" ports, the pressure in the filter and pipe relative to the atmosphere is completely irrelevant to the transmitter's output signal. The filter could be operating at a pressure of 10 psig or 10,000 psig: the only thing the DP transmitter measures is the pressure *drop* across the filter. If the upstream side is at 10 psig and the downstream side is at 9 psig, the differential pressure will be 1 psid ("D" for *differential*). If the upstream pressure is 10,000 psig and the downstream pressure is 9,999 psig, the DP transmitter will still see a differential pressure of just 1 psid. Likewise, the technician calibrating the DP transmitter on the workbench could use a precise air pressure of just 1 psig (applied to the "high" port, with the "low" port vented to atmosphere) to simulate either of these real-world conditions. The DP transmitter simply cannot tell the difference between these three scenarios, nor should it be able to tell the difference if its purpose is to exclusively measure differential pressure.

In the world of electronics, the ability of a differential voltage sensor (such as an operational amplifier) to sense small differences in voltage while ignoring large potentials measured with reference to ground is referred to by the phrase *common-mode rejection*. An ideal operational amplifier completely ignores the amount of voltage common to both input terminals, responding only to the *difference* in voltage *between* those terminals. This is precisely what a well-designed differential

pressure instrument does, except with fluid pressure instead of electrical voltage. A differential pressure instrument all but ignores gauge pressure common to both ports, while responding only to *differences* in pressure *between* those two ports.

A vivid example of this may be inferred from the nameplate of a Foxboro model 13A differential pressure transmitter, shown in this photograph:

Figure 10.34: A differential pressure transmitter nameplate

This nameplate tells us that the transmitter has a calibrated differential pressure range of 50" $H_2O$ (50 inches water column, which is only about 1.8 psig). However, the nameplate also tells us that the transmitter has a *maximum working pressure* (MWP) of 1500 psig. "Working pressure" refers to the amount of gauge pressure common to each port, not the differential pressure between ports. Taking these figures at face value means this transmitter will register zero (no differential pressure) even if the gauge pressure applied equally to both ports is a full 1500 psig! In other words, this differential pressure transmitter will *reject* up to 1500 psig of gauge pressure, and respond only to small differences in pressure between the ports (1.8 psid being enough to stimulate the transmitter to full scale output).

## 10.6 Pressure switches

A *pressure switch* is one detecting the presence of fluid pressure. Pressure switches often use diaphragms or bellows as the pressure-sensing element, the motion of which actuates one or more switch contacts.

Recall from section 9.1 that the "normal" status of a switch is the resting condition of *no*

*stimulation*. A pressure switch will be in its "normal" status when it senses minimum pressure (e.g., an applied pressure, or in some cases a vacuum condition)[11]. For a pressure switch, "normal" status is any fluid pressure *below* the trip threshold of the switch.

Figure 10.35: Schematic representations of pressure switches

The following photograph shows two pressure switches sensing the same fluid pressure as an electronic pressure transmitter (the device on the far left):

Figure 10.36: A pressure transmitter sharing impulse piping with two pressure switches

A legacy design of pressure switch uses a Bourdon tube as the pressure-sensing element, and a glass bulb partially filled with mercury as the electrical switching element. When applied pressure causes the Bourdon tube to flex sufficiently, the glass bulb tilts far enough to cause the mercury to fall against a pair of electrodes, thus completing an electrical circuit. A great many pressure switches of this design were sold under the brand name of "Mercoid," with a few appearing in this photograph of a steam boiler (the round-shaped units with glass covers allowing inspection of the Bourdon tube and mercury tilt switch):

[11] If the trip setting of a pressure switch is below atmospheric pressure, then it will be "actuated" at atmospheric pressure and in its "normal" status only when the pressure falls below that trip point (i.e., a vacuum).

Figure 10.37: Pressure switches installed on a boiler

A close-up photograph of one of these pressure switches appears here. The Bourdon tube is grey in color, and almost as wide in diameter as the circular switch housing. The mercury tilt switch bottles have yellow-colored plastic caps covering up their external electrical contacts:

Figure 10.38: Closeup of a pressure switch installed on a boiler

The next set of photographs show a mercury tilt switch removed from the pressure switch mechanism, so the switch appears in two different states (contact open on the left, and closed on the right):

Advantages of mercury tilt switches include immunity to switch contact degradation from harmful atmospheres (oil mist, dirt, dust, corrosion) as well as safety in explosive atmospheres (since a spark contained within a hermetically sealed glass bulb cannot touch off an explosion in the surrounding

atmosphere). Disadvantages include the possibility of intermittent electrical contact resulting from mechanical vibration, as well as sensitivity to mounting angle (e.g., do *not* use this kind of switch aboard a moving vehicle!), not to mention that mercury is a hazardous material.

A pressure switch manufactured by the Danfoss corporation appears in the next photograph. This particular model of pressure switch has windows on the front cover allowing a technician to see the pressure limit setting inside:

Figure 10.39: Closeup of a pressure switch

This switch balances the force generated by a pressure-sensing element against a mechanical spring. Tension on the spring may be adjusted by a technician, which means the trip point of this switch is adjustable.

One of the settings on this switch is the *dead-band* or *differential* pressure setting, seen in the lower window. This setting determines the amount of pressure change required to re-set the switch to its normal state after it has tripped. For example, a high-pressure switch with a trip point of 67 psig (changes state at 67 psig, increasing) that re-sets back to its normal state at a pressure of 63 psig decreasing has a "dead-band" or "differential" pressure setting of 4 psid (67 psig − 63 psig = 4 psid).

The "differential" pressure setting of a gauge pressure switch is not to be confused with a true *differential pressure* switch. The next photograph shows a pressure switch truly actuated by *differential* pressure (the difference in fluid pressure sensed between two ports):

Figure 10.40: A differential pressure switch

The electrical switch element is located underneath the blue cover, while the diaphragm pressure element is located within the grey metal housing. The net force exerted on the diaphragm by the two fluid pressures varies in magnitude and direction with the magnitude of those pressures. If the two fluid pressures are precisely equal, the diaphragm experiences no net force (zero differential pressure).

Like the Danfoss gauge pressure switch seen previously, this differential pressure switch has a "trip" or "limit" setting as well as a "dead-band" or "differential" setting. It is important to recognize and clearly distinguish the two meanings of *differential pressure* in the context of this device. It senses differences in pressure between two input ports ("differential pressure" – the difference between two different fluid pressure connections), but being a switch, it also exhibits some dead band in its action ("differential pressure" – a change in pressure required to re-set the switch's state).

## 10.7 Pressure sensor installation and accessories

Multiple accessories exist for pressure-sensing devices to function optimally in challenging process environments. Sometimes, special accessories must be used to protect the pressure instrument against hazards of certain process fluids. One such hazard is pressure *pulsation*, for example at the discharge of a piston-type (positive-displacement) high-pressure pump. Pulsating pressure can quickly damage mechanical sensors such as Bourdon tubes, either by wear of the mechanism transferring pressure element motion to an indicating needle, and/or fatigue of the metal element itself.

Pressure sensing elements are typically connected to the sides of horizontal pipes carrying liquid (to avoid noncondensing gas bubbles that may be trapped at the top of the pipe and "sludge" that may exist along the bottom of the pipe), and to the tops of horizontal pipes carrying gases (to avoid condensation that may exist along the bottom of the pipe). Impulse lines that are liquid-filled are generally sloped downward slightly (at a minimum of one inch per foot) to ensure they stay filled and vent out any noncondensible gas bubbles; impulse lines that are gas-filled are generally sloped upward slightly (at the same minimum slope) to allow condensate to drain out of them.

### 10.7.1 Valve manifolds

An important accessory to the differential pressure transmitter is the *three-valve manifold*. This device incorporates three manual valves to isolate and equalize pressure from the process to the transmitter, for maintenance and calibration purposes.

The following illustration shows the three valves comprising a three-valve manifold (within the dotted-line box), as well as a fourth valve called a "bleed" valve used to vent trapped fluid pressure to atmosphere:

Figure 10.41: Diagram of a three-valve manifold and a bleed valve used with a DP sensor

While this illustration shows the three valves as separate devices, connected together and to the transmitter by tubing, three-valve manifolds are more commonly manufactured as monolithic devices: the three valves cast together into one block of metal, attaching to the pressure transmitter

by way of a flanged face with O-ring seals. Bleed valves are most commonly found as separate devices threaded into one or more of the ports on the transmitter's diaphragm chambers.

The following photograph shows a three-valve manifold bolted to a Honeywell model ST3000 differential pressure transmitter. A bleed valve fitting may be seen inserted into the upper port on the nearest diaphragm capsule flange:

Figure 10.42: An installation with a three-valve manifold

In normal operation, the two block valves are left open to allow process fluid pressure to reach the transmitter. The equalizing valve is left tightly shut so no fluid can pass between the "high" and "low" pressure sides. To isolate the transmitter from the process for maintenance, one must first close the block valves, then open the equalizing valve to ensure the transmitter "sees" no differential pressure. The "bleed" valve is opened at the very last step to relieve pent-up fluid pressure within the manifold and transmitter chambers:

Figure 10.43: Operation of a three-valve manifold

A variation on this theme is the *five-valve manifold*, shown in this illustration:

Figure 10.44: Diagram of a five-valve manifold

Manifold valve positions for normal operation and maintenance are as follows:

Figure 10.45: Operation of a five-valve manifold

It is critically important that the equalizing valve(s) never be open while both block valves are open! Doing so will allow process fluid to flow through the equalizing valve(s) from the high-pressure side of the process to the low-pressure side of the process. If the impulse tubes connecting the manifold to the process are intentionally filled with a *fill fluid* (such as glycerin, to displace process water from entering the impulse tubes; or water in a steam system), this fill fluid will be lost. Also, if the process fluid is dangerously hot or radioactive, a combination of open equalizing and block valves will let that dangerous fluid reach the transmitter and manifold, possibly causing damage or creating a personal hazard. If this mistake is made on a differential pressure transmitter connected to a steam system, hot steam might flow through the manifold and overheat the equalizing valve so that it seizes open and cannot be shut again!

Pressure transmitter valve manifolds also come in single block-and-bleed configurations, for gauge pressure applications. Here, the "low" pressure port of the transmitter is vented to atmosphere, with only the "high" pressure port connected to the impulse line:

Figure 10.46: Diagram of a block-and-bleed manifold

The following photograph shows a bank of eight pressure transmitters, seven out of the eight being equipped with a single block-and-bleed manifold. The eighth transmitter (bottom row, second-from left) sports a 5-valve manifold:

Figure 10.47: Photo of installed pressure and differential pressure transmitters

### 10.7.2 Bleed fittings

Before removing a pressure transmitter from live service, the technician must "bleed" stored fluid pressure to atmosphere in order to achieve a *zero energy state* prior to disconnecting the transmitter from the impulse lines. Some valve manifolds provide a bleed valve for doing just this, but many do not[12]. An inexpensive and common accessory for pressure-sensing instruments (especially transmitters) is the *bleed valve fitting*, installed on the instrument as a discrete device. The most common bleed fitting is equipped with 1/4 inch male NPT pipe threads, for installation into one of the 1/4 inch NPT threaded pipe holes typically provided on pressure transmitter flanges. The bleed is operated with a small wrench, loosening a ball-tipped plug off its seat to allow process fluid to escape through a small vent hole in the side of the fitting. The following photographs show close-up views of a bleed fitting both assembled (left) and with the plug fully extracted from the fitting (right). The bleed hole may be clearly seen in both photographs:

When installed directly on the flanges of a pressure instrument, these bleed valves may be used to bleed unwanted fluids from the pressure chambers, for example bleeding air bubbles from an instrument intended to sense water pressure, or bleeding condensed water out of an instrument intended to sense compressed air pressure.

The following photographs show bleed fittings installed two different ways on the side of a pressure transmitter flange, one way to bleed gas out of a liquid process (located on top) and the other way to bleed liquid out of a gas process (located on bottom):

[12]The standard 3-valve manifold, for instance, does not provide a bleed valve – only block and equalizing valves.

### 10.7.3 Pressure pulsation damping

A simple way to mitigate the effects of pulsation on a pressure gauge is to fill the inside of the gauge with a viscous liquid such as glycerin or oil. The inherent friction of this fill liquid has a "shock-absorber" quality which damps the gauge mechanism's oscillatory motion and helps protect against damage from pulsations or from external vibration. This method is ineffectual for high-amplitude pulsations, though.

An oil-filled pressure gauge may be seen in the following photograph. Note the air bubble near the top of the gauge face, which is the only visual indication of an oil filling:

Figure 10.48: An oil-filled pressure gauge

A more sophisticated method for damping pulsations seen by a pressure instrument is called a *snubber*, and it consists of a fluid restriction placed between with the pressure sensor and the process. The simplest example of a snubber is a simple *needle valve* (an adjustable valve designed for low flow rates) placed in a mid-open position, restricting fluid flow in and out of a pressure gauge:

Figure 10.49: Diagram of a needle valve used as a snubber with a pressure gauge

At first, the placement of a throttling valve between the process and a pressure-measuring

instrument seems rather strange, because there should not be any continuous flow in or out of the gauge for such a valve to throttle! However, a *pulsing* pressure causes a small amount of *alternating* flow in and out of the pressure instrument, owing to the expansion and contraction of the mechanical pressure-sensing element (bellows, diaphragm, or Bourdon tube). The needle valve provides a restriction for this flow which, when combined with the fluid capacitance of the pressure instrument, combine to form a low-pass filter of sorts. By impeding the flow of fluid in and out of the pressure instrument, that instrument is prevented from "seeing" the high and low peaks of the pulsating pressure. Instead, the instrument registers a much steadier pressure over time. An electrical analogy for a pressure snubber is an RC low-pass filter circuit "damping" voltage pulsations from reaching a voltmeter:

Figure 10.50: Electrical analogy of a mechanical snubber

One potential problem with the needle valve solution is that the small orifice inside the valve may plug up over time with debris or deposits from dirty process fluid. This, of course, would be bad because if that valve were to ever completely plug, the pressure instrument would stop responding to any changes in process pressure at all, or perhaps just become too slow in responding to major changes.

A solution to this problem is to fill the pressure sensor mechanism with a clean liquid (called a *fill fluid*), then transfer pressure from the process fluid to the fill fluid (and then to the pressure-sensing element) using a slack diaphragm or some other membrane:

Figure 10.51: Use of an isolating diaphragm and fill fluid with a snubber

In order for the fill fluid and isolating diaphragm to work effectively, there cannot be any gas bubbles in the fill fluid – it must be a "solid" hydraulic system from the diaphragm to the sensing element. The presence of gas bubbles means that the fill fluid is compressible, which means the isolating diaphragm may have to move more than necessary to transfer pressure to the instrument's sensing element. This will introduce pressure measurement errors if the isolating diaphragm begins to tense from excessive motion (and thereby oppose some process fluid pressure from fully transferring to the fill fluid), or hit a "stop" point where it cannot move any further (thereby preventing any further transfer of pressure from process fluid to fill fluid)[13]. For this reason, isolating diaphragm systems for pressure instruments are usually "packed" with fill fluid at the point and time of manufacture, then sealed in such a way that they cannot be opened for any form of maintenance. Consequently, any fill fluid leak in such a system immediately ruins it.

### 10.7.4 Remote and chemical seals

Isolating diaphragms have merit even in scenarios where pressure pulsations are not a problem. Consider the case of a food-processing system where pressure must be remotely measured inside a mixing vessel:

[13]This concept will be immediately familiar to anyone who has ever had to "bleed" air bubbles out of an automobile brake system. With air bubbles in the system, the brake pedal has a "spongy" feel when depressed, and much pedal motion is required to achieve adequate braking force. After bleeding all air out of the brake fluid tubes, the pedal motion feels much more "solid" than before, with minimal motion required to achieve adequate braking force. Imagine the brake pedal being the isolating diaphragm, and the brake pads being the pressure sensing element inside the instrument. If enough gas bubbles exist in the tubes, the brake pedal might stop against the floor when fully pressed, preventing full force from ever reaching the brake pads! Likewise, if the isolating diaphragm hits a hard motion limit due to gas bubbles in the fill fluid, the sensing element will not experience full process pressure!

Figure 10.52: Diagram of remote pressure monitoring

The presence of the tube connecting the vessel to the pressure gauge poses a hygiene problem. Stagnant process fluid (in this case, some liquid food product) inside the tube can support microbial growth, which will eventually contaminate the vessel no matter how well or how often the vessel is cleaned. Even automated *Clean-In-Place* and *Steam-In-Place* (*CIP* and *SIP*, respectively) protocols where the vessel is chemically purged between batches cannot prevent this problem because the cleaning agents never purge the entire length of the tubing (ultimately, to the Bourdon tube or other sensing element inside the gauge).

Here, a valid application of an isolating diaphragm and fill fluid is seen. If an isolating diaphragm is mounted to the vessel in such a way that the process fluid directly contacts the diaphragm, sealed fill fluid will be the only material inside the tubing carrying that pressure to the instrument. Furthermore, the isolating diaphragm will be directly exposed to the vessel interior, and therefore cleaned with every CIP cycle. Thus, the problem of microbial contamination is completely avoided:

Figure 10.53: Diagram of remote pressure monitoring using an isolating diaphragm and filled capillary

Such systems are often referred to as *remote seals*, and they are available on a number of different pressure instruments including gauges, transmitters, and switches. If the purpose of an isolating diaphragm and fill fluid is to protect the sensitive instrument from corrosive or otherwise harsh chemicals, it is often referred to as a *chemical seal*.

The following photograph shows a pressure gauge equipped with a chemical seal diaphragm. Note that the chemical seal on this particular gauge is close-coupled to the gauge, since the only goal here is protection of the gauge from harsh process fluids, not the ability to remotely mount the gauge:

Figure 10.54: A pressure gauge with an isolating diaphragm

A view facing the bottom of the flange reveals the thin metal isolating diaphragm keeping process fluid from entering the gauge mechanism. Only inert fill fluid occupies the space between this diaphragm and the gauge's Bourdon tube:

Figure 10.55: Closeup of the isolating diaphragm

The only difference between this chemical-seal gauge and a remote-seal gauge is that a remote-seal gauge uses a length of very small-diameter tubing called *capillary tubing* to transfer fill fluid pressure from the sealing diaphragm to the gauge mechanism.

Direct-reading gauges are not the only type of pressure instrument that may benefit from having remote seals. Electronic pressure *transmitters* are also manufactured with remote seals for the same reasons: protection of the transmitter sensor from harsh process fluid, or prevention of "dead-end" tube lengths where organic process fluid would stagnate and harbor microbial growths. The following photograph shows a pressure transmitter equipped with a remote sealing diaphragm. The capillary tube is protected by a coiled metal ("armor") sheath:

Figure 10.56: A pressure transmitter with a capillary and remote diaphragm seal

A close-up view of the sealing diaphragm shows its corrugated design, allowing the metal to easily flex and transfer pressure to the fill fluid within the capillary tubing[14]:

[14]Like all instrument diaphragms, this one is sensitive to damage from contact with sharp objects. If the diaphragm ever becomes nicked, dented, or creased, it will tend to exhibit hysteresis in its motion, causing calibration errors for

Figure 10.57: Closeup of a remote diaphragm seal

Just like the isolating diaphragms of the pressure-sensing capsule, these remote diaphragms need only transfer process fluid pressure to the fill fluid and (ultimately) to the taut sensing diaphragm inside the instrument. Therefore, these diaphragms perform their function best if they are designed to easily flex. This allows the taut sensing diaphragm to provide the vast majority of the opposing force to the fluid pressure, as though it were the only spring element in the fluid system.

The connection point between the capillary tube and the transmitter's sensor capsule is labeled with a warning never to disassemble, since doing so would allow air to enter the filled system (or fill fluid to escape from the system) and thereby ruin its accuracy:

the instrument. For this reason, isolating diaphragms are often protected from contact by a plastic plug when the instrument is shipped from the manufacturer. This plug must be removed from the instrument before placing it into service.

Figure 10.58: Closeup of the capillary connection to a remote diaphragm seal

In order for a remote seal system to work, the hydraulic "connection" between the sealing diaphragm and the pressure-sensing element must be completely gas-free so there will be a "solid" transfer of motion from one end to the other.

A potential problem with using remote diaphragms is the hydrostatic pressure generated by the fill fluid if the pressure instrument is located at an elevation other than the process connection point. For example, a pressure gauge located far below the vessel it connects to will register a *greater* pressure than what is actually inside the vessel, because the hydrostatic pressure caused by the liquid in the tubing adds to the vessel's pressure:

Figure 10.59: The problem of hydrostatic head when measuring pressure

This pressure may be calculated by the formula $\rho gh$ or $\gamma h$ where $\rho$ is the mass density of the fill liquid or $\gamma$ is the weight density of the fill liquid. For example, a 12 foot capillary tube height filled with a fill liquid having a weight density of 58.3 lb/ft$^3$ will generate an elevation pressure of almost 700 lb/ft$^2$, or 4.86 psig. If the pressure instrument is located below the process connection point, this 4.86 psig offset must be incorporated into the instrument's calibration range. If this pressure instrument must accurately measure a process pressure range of 0 to 50 psig, it must be calibrated for an actual range of 4.86 to 54.86 psig.

The reverse problem exists where the pressure instrument is located *higher* than the process connection: here the instrument will register a *lower* pressure than what is actually inside the vessel, offset by the amount predicted by the hydrostatic pressure formulae $\rho gh$ or $\gamma h$.

In all fairness, this problem is not limited to remote seal systems – even non-isolated systems where the tubing is filled with process liquid will exhibit this offset error. However, in filled-capillary systems a vertical offset is *guaranteed* to produce a pressure offset because fill fluids are always liquid, and liquids generate pressure in direct proportion to the vertical height of the liquid column (and to the density of that liquid).

A similar problem unique to isolated-fill pressure instruments is measurement error caused by temperature extremes. Suppose the liquid-filled capillary tube of a remote seal pressure instrument comes too near a hot steam pipe, furnace, or some other source of high temperature. The expansion of the fill fluid may cause the isolation diaphragm to extend to the point where it begins to tense and add a pressure to the fill fluid above and beyond that of the process fluid. Cold temperatures may wreak havoc with filled capillary tubes as well, if the fill fluid congeals or even freezes such that it no longer flows as it should.

Proper mounting of the instrument and proper selection of the fill fluid[15] will help to avoid such problems. All in all, the potential for trouble with remote- and chemical-seal pressure instruments is greatly offset by their benefits in the right applications.

Some pressure transmitters are equipped with close-coupled seals rather than remote seals, for applications where it is best to avoid impulse tube connections to the process (e.g., liquids that tend to coagulate or in hygienic processes). A Rosemount extended-diaphragm pressure transmitter appears in the left-hand photograph, while a Yokogawa transmitter of the same basic design is shown installed in a working process in the right-hand photograph:

### 10.7.5 Filled impulse lines

An alternate method for isolating a pressure-sensing instrument from direct contact with process fluid is to either *fill* or *purge* the impulse lines with a harmless fluid. Filling impulse tubes with a static fluid works when gravity is able to keep the fill fluid in place, such as in this example of a pressure transmitter connected to a water pipe by a glycerin-filled impulse line:

[15] Most pressure instrument manufacturers offer a range of fill fluids for different applications. Not only is temperature a consideration in the selection of the right fill fluid, but also potential contamination of or reaction with the process if the isolating diaphragm ever suffers a leak!

Figure 10.60: Diagram showing use of filled impulse lines

A reason someone might do this is for freeze protection, since glycerin freezes at a lower temperature than water. If the impulse line were filled with process water, it might freeze solid in cold weather conditions (the water in the pipe cannot freeze so long as it is forced to flow). The greater density of glycerin keeps it placed in the impulse line, below the process water line. A fill valve is provided near the transmitter so a technician may re-fill the impulse line with glycerin (using a hand pump) if ever needed.

As with a remote diaphragm, a filled impulse line will generate its own pressure proportional to the height difference between the point of process connection and the pressure-sensing element. If the height difference is substantial, the pressure offset resulting from this difference in elevation will require compensation by means of an intentional "zero shift" of the pressure instrument when it is calibrated.

### 10.7.6 Purged impulse lines

Continuous purge of an impulse line is an option when the line is prone to plugging. Consider this example, where pressure is measured at the bottom of a sedimentation vessel:

Figure 10.61: Diagram showing use of purged impulse lines

A continuous flow of clean water enters through a "purge valve" and flows through the impulse line, keeping it clear of sediment while still allowing the pressure instrument to sense pressure at the bottom of the vessel. A *check valve* guards against reverse flow through the purge line, in case process fluid pressure ever exceeds purge supply pressure. Purged systems are very useful, but a few details are necessary to consider before deciding to implement such a strategy:

- How reliable is the supply of purge fluid? If this stops for any reason, the impulse line may plug!
- Is the purge fluid supply pressure guaranteed to exceed the process pressure at all times, for proper direction of purge flow?
- What options exist for purge fluids that will not adversely react with the process?
- What options exist for purge fluids that will not contaminate the process?
- How expensive will it be to maintain this constant flow of purge fluid into the process?

Also, it is important to limit the flow of purge fluid to a rate that will not create a falsely high pressure measurement due to restrictive pressure drop across the length of the impulse line, yet flow freely enough to achieve the goal of plug prevention. In many installations, a visual flow indicator is installed in the purge line to facilitate optimum purge flow adjustment. Such flow indicators are also helpful for troubleshooting, as they will indicate if anything happens to stop the purge flow.

In the previous example, the purge fluid was clean water. Many options exist for purge fluids other than water, though. Gases such as air, nitrogen, or carbon dioxide are often used in purged systems, for both gas and liquid process applications.

Purged impulse lines, just like filled lines and diaphragm-isolated lines, will generate hydrostatic pressure with vertical height. If the purge fluid is a liquid, this elevation-dependent pressure may be an offset to include in the instrument's calibration. If the purge fluid is a gas (such as air), however, any height difference may be ignored because the density of the gas is negligible.

### 10.7.7 Heat-traced impulse lines

If impulse lines are filled with liquid, there may exist a possibility for that liquid to freeze in cold-weather conditions. This possibility depends, of course, on the type of liquid filling the impulse lines and how cold the weather gets in that geographic location.

One safeguard against impulse line freezing is to *trace* the impulse lines with some form of active heating medium, steam and electrical being the most common. "Steam tracing" consists of a copper tube carrying low-pressure steam, bundled alongside one or more impulse tubes, enclosed in a thermally insulating jacket.

Figure 10.62: Diagram showing use of heated impulse lines

Steam flows through the shut-off valve, through the tube in the insulated bundle, transferring heat to the impulse tube as it flows past. Cooled steam condenses into water and collects in the *steam trap* device located at the lowest elevation on the steam trace line. When the water level builds up to a certain level inside the trap, a float-operated valve opens to vent the water. This allows more steam to flow into the tracing tube, keeping the impulse line continually heated.

The steam trap naturally acts as a sort of thermostat as well, even though it only senses condensed water level and not temperature. The rate at which steam condenses into water depends on how cold the impulse tube is. The colder the impulse tube (caused by colder ambient conditions), the more heat energy drawn from the steam, and consequently the faster condensation rate of steam into water. This means water will accumulate faster in the steam trap, which means it will "blow down" more often. More frequent blow-down events means a greater flow rate of steam into the tracing tube, which adds more heat to the tubing bundle and raises its temperature. Thus, the system is naturally regulating, with its own negative feedback loop to maintain bundle temperature at a relatively stable point[16].

The following photograph shows a picture of a steam trap:

[16] In fact, the blow-down frequency can serve as a crude ambient temperature thermometer! Steam traps seldom blow down during warm weather, but their "popping" is much more regular (one every minute or less) when ambient temperatures drop well below the freezing point of water.

Figure 10.63: A steam trap

Steam traps are not infallible, being susceptible to freezing (in *very* cold weather) and sticking open (wasting steam by venting it directly to atmosphere). However, they are generally reliable devices, capable of adding tremendous amounts of heat to impulse tubing for protection against freezing.

Electrically traced impulse lines are an alternative solution for cold-weather problems. The "tracing" used is a twin-wire cable (sometimes called *heat tape*) that acts as a resistive heater. When power is applied, the cable heats up, thus imparting thermal energy to the impulse tubing it is bundled with.

Heat tape may be self-regulating, or controlled with an external thermostat. Self-regulating heat tape exhibits an electrical resistance that varies with temperature, automatically self-regulating its own temperature without the need for external controls.

Both steam and electrical heat tracing are used to protect instruments themselves from cold weather freezing, not just the impulse lines. In these applications it is important to remember that only the liquid-filled portions of the instrument need freeze protection, not the electronics portions!

### 10.7.8 Water traps and pigtail siphons

Many industrial processes utilize high-pressure steam for direct heating, performing mechanical work, combustion control, and as a chemical reactant. Measuring the pressure of steam is important both for its end-point use and its generation (in a boiler). One problem with doing this is the relatively high temperature of steam at the pressures common in industry, which can cause damage to the sensing element of a pressure instrument if directly connected.

A simple yet effective solution to this problem is to intentionally create a "low" spot in the impulse line where condensed steam (water) will accumulate and act as a liquid barrier to prevent hot steam from reaching the pressure instrument. The principle is much the same as a plumber's trap used underneath sinks, creating a liquid seal to prevent noxious gases from entering a home from the sewer system. A loop of tube or pipe called a *pigtail siphon* achieves the same purpose:

Figure 10.64: Protecting instruments from process steam

The following photograph shows a pigtail siphon connected to a pressure gauge sensing pressure on a steam line:

Figure 10.65: A pigtail siphon

### 10.7.9 Mounting brackets

An accessory specifically designed for differential pressure transmitters, but useful for other field-mounted instruments as well, is the *2 inch pipe mounting bracket*. Such a bracket is manufactured from heavy-gauge sheet metal and equipped with a U-bolt designed to clamp around any 2 inch black iron pipe. Holes stamped in the bracket match mounting bolts on the capsule flanges of most common differential pressure transmitters, providing a mechanically stable means of attaching a differential pressure transmitter to a framework in a process area.

The following photographs show several different instruments mounted to pipe sections using these brackets:

### 10.7.10 Heated enclosures

In installations where the ambient temperature may become very cold, a protective measure against fluid freezing inside a pressure transmitter is to house the transmitter in an insulated, heated enclosure. The next photograph shows just such an enclosure with the cover removed:

Figure 10.66: A transmitter inside a heated enclosure (lid removed)

Not surprisingly, this installation works well to protect all kinds of temperature-sensitive instruments from extreme cold. Here, an explosive gas sensor is shown mounted inside a slightly different style of insulated enclosure, with the lid opened up for inspection:

Figure 10.67: A transmitter inside a heated enclosure (lid open)

## 10.8 Process/instrument suitability

On a fundamental level, pressure is universal. Regardless of the fluid in question; liquid or gas, hot or cold, corrosive or inert, pressure is nothing more than the amount of force exerted by that fluid over a unit area:

$$P = \frac{F}{A}$$

It should come as no surprise, then, that the common mechanical sensing elements for measuring pressure (bellows, diaphragm, Bourdon tube, etc.) are equally applicable to all fluid pressure measurement applications, at least in principle. It is normally a matter of proper material selection and element strength (material thickness) to make a pressure instrument suitable for any range of process fluids.

Fill fluids used in pressure instruments – whether it be the dielectric liquid inside a differential capacitance sensor, the fill liquid of a remote or chemical seal system, or liquid used to fill a vertical section of impulse tubing – must be chosen so as to not adversely react with or contaminate the process.

Pure oxygen processes require that no system component have traces of hydrocarbon fluids present. While oxygen itself is not explosive, it greatly accelerates the combustion and explosive potential of any flammable substance. Therefore, a pressure gauge calibrated using oil as the working fluid in a deadweight tester would definitely *not* be suitable for pure oxygen service! The same may be said for a differential pressure transmitter with a hydrocarbon-based fill inside its pressure-sensing capsule[17].

Pharmaceutical, medical, and food manufacturing processes require strict purity and the ability to disinfect all elements in the process system at will. Stagnant lines are not allowed in such processes, as microbe cultures may flourish in such "dead end" piping. Remote seals are very helpful in overcoming this problem, but the fill fluids used in remote systems must be chosen such that a leak in the isolating diaphragm will not contaminate the process.

Manometers, of course, are rather limited in their application, as their operation depends on direct contact between process fluid and manometer liquid. In the early days of industrial instrumentation, liquid mercury was a very common medium for process manometers, and it was not unusual to see a mercury manometer used in direct contact with a process fluid such as oil or water to provide pressure indication:

[17] Although this fluid would not *normally* contact pure oxygen in the process, it could if the isolating diaphragm inside the transmitter were to ever leak.

Figure 10.68: Diagram of a mercury manometer

Thankfully, those days are gone. Mercury (chemical symbol "Hg") is a toxic metal and therefore hazardous to use. Calibration of these manometers was also challenging due to the column height of the process liquid in the impulse line and the range tube. When the process fluid is a gas, the difference in mercury column height directly translates to sensed pressure by the hydrostatic pressure formula $P = \rho g h$ or $P = \gamma h$. When the process fluid is a liquid, though, the shifting of mercury columns also creates a change in height of the process liquid column, which means the indicated pressure is a function of the height difference ($h$) *and* the difference in density between the process liquid and mercury. Consequently, the indications provided by mercury manometers in liquid pressure applications were subject to correction according to process liquid density.

# References


[1] e. a. A. v. Beckerath, WIKA-Handbook, Pressure and Temperature Measurement. Klingenberg, Germany: WIKA Alexander Wiegand GmbH & Co., 1995.

[2] Rosemount 1199 Fill Fluid Specifications, 00840-2100-4016, Emerson Process Management Rosemount Division, Chanhassen, MN, 2005.

[3] A. E. Fribance, Industrial Instrumentation Fundamentals. New York, NY: McGraw-Hill, 1962.

[4] H. P. Kallen, Handbook of Instrumentation and Controls. New York, NY: McGraw-Hill, 1961.

[5] D. R. Patrick and S. R. Patrick, Pneumatic Instrumentation. Albany, NY: Delmar Publishers, Inc., 1993.

[6] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1.

# Chapter 11

# Temperature and Humidity Measurement

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Discuss the basic functions of temperature detectors.
2. Draw P&ID symbols used for temperature and humidity detection equipment, including sensing elements, transmitters, and protective wells.
3. Discuss the mechanisms that allow bi-metal and filled-bulb devices to sense temperature.
4. Explain how semiconductor devices may be used to sense temperature.
5. Explain how metallic RTDs may be used to sense temperature.
6. Describe the construction of a basic thermocouple including the materials used.
7. Identify different thermocouples by insulation color codes.
8. State and apply the laws of thermoelectric circuits.
9. Describe the use of cold-junction compensation in thermocouple circuits.
10. Explain how an optical pyrometer provides an output representative of the measured temperature.
11. Describe the environmental conditions that can affect the accuracy and reliability of temperature detection instrumentation.
12. Explain the methods of bridge circuit compensation for changes in ambient temperature.
13. Describe calibration methods for temperature transmitters.
14. Discuss and sketch methods of temperature sensor installation, including advantages and disadvantages.
15. Describe various sensing methods for airborne moisture such as relative humidity and dewpoint.
16. Describe calibration methods for humidity transmitters.
17. Discuss and sketch methods of humidity sensor installation, including advantages and disadvantages.

*Temperature* is the measure of average molecular kinetic energy within a substance. The concept is easiest to understand for gases under low pressure, where gas molecules randomly shuffle about. The average kinetic (motional) energy of these gas molecules defines temperature for that quantity of gas. There is even a formula expressing the relationship between average kinetic energy ($\overline{E_k}$) and temperature ($T$) for a monatomic (single-atom molecule) gas:

$$\overline{E_k} = \frac{3kT}{2}$$

Where,
$\overline{E_k}$ = Average kinetic energy of the gas molecules (joules)
$k$ = Boltzmann's constant ($1.38 \times 10^{-23}$ joules/Kelvin)
$T$ = Absolute temperature of gas (Kelvin)

*Thermal energy* is a different concept: the quantity of *total kinetic energy* for this random molecular motion. If the average kinetic energy is defined as $\frac{3kT}{2}$, then the total kinetic energy for all the molecules in a monatomic gas must be this quantity times the total number of molecules ($N$) in the gas sample:

$$E_{\text{thermal}} = \frac{3NkT}{2}$$

This may be equivalently expressed in terms of the number of *moles* of gas rather than the number of molecules (a staggeringly large number for any realistic sample):

$$E_{\text{thermal}} = \frac{3nRT}{2}$$

Where,
$E_{\text{thermal}}$ = Total thermal energy for a gas sample (joules)
$n$ = Quantity of gas in the sample (moles)
$R$ = Ideal gas constant (8.315 joules per mole-Kelvin)
$T$ = Absolute temperature of gas (Kelvin)

*Heat* is defined as the exchange of thermal energy from one sample to another, by way of conduction (direct contact), convection (transfer via a moving fluid), or radiation (emitted energy); although the terms *thermal energy* and *heat* are often used interchangeably.

Temperature is a more easily detected quantity than heat. There are many different ways to measure temperature, from a simple glass-bulb mercury thermometer to sophisticated infra-red optical sensor systems. Like all other areas of measurement, there is no single technology that is best for all applications. Each temperature-measurement technique has its own strengths and weaknesses. One responsibility of the I&C engineer is to know these pros and cons so as to choose the best technology for the application, and this knowledge is best obtained through understanding the operational principles of each technology.

## 11.1 Bi-metal temperature sensors

Solids tend to expand when heated. The amount that a solid sample will expand with increased temperature depends on the size of the sample, the material it is made of, and the amount of temperature rise. The following formula relates linear expansion to temperature change:

$$l = l_0(1 + \alpha\Delta T)$$

Where,

$l$ = Length of material after heating
$l_0$ = Original length of material
$\alpha$ = Coefficient of linear expansion
$\Delta T$ = Change in temperature

Here are some typical values of $\alpha$ for common metals:

- Aluminum = $25 \times 10^{-6}$ per °C
- Copper = $16.6 \times 10^{-6}$ per °C
- Iron = $12 \times 10^{-6}$ per °C
- Tin = $20 \times 10^{-6}$ per °C
- Titanium = $8.5 \times 10^{-6}$ per °C

As can be seen, the values for $\alpha$ are quite small. This means the amount of expansion (or contraction) for modest temperature changes are almost too small to see unless the sample size ($l_0$) is huge. The effects of thermal expansion can readily be seen in structures such as bridges, where expansion joints must be incorporated into the design to prevent serious problems due to changes in ambient temperature. However, for a sample the size of a human hand the change in length from a cold day to a warm day will be microscopic.

One way to amplify the motion resulting from thermal expansion is to bond two strips of dissimilar metals together, such as copper and iron. If two equally-sized strips of copper and iron were laid side by side, and then both of them heated to a higher temperature, the copper strip would lengthen slightly more than the iron strip:

Figure 11.1: Relative expansion of two different metals

If these two strips of metal were bonded together, this differential growth results in a bending motion that greatly exceeds the linear expansion. This device is called a *bimetal strip*:

Figure 11.2: Expansion of two different metals bonded together

This bending motion is significant enough to drive a pointer mechanism, activate an electromechanical switch, or perform any number of other mechanical tasks, making this a very simple and useful *primary sensing element* for temperature.

If a bimetallic strip is twisted over a long length, it will tend to un-twist as it heats up. This twisting motion may be used to directly drive the needle of a temperature gauge. This is the operating principle of the temperature gauge shown in the following photograph:

Figure 11.3: A temperature gauge using bimetallic strip sensing

## 11.2 Filled-bulb temperature sensors

Filled-bulb systems exploit the principle of fluid expansion to measure temperature. If a fluid is enclosed in a sealed system and then heated, the molecules in that fluid will exert a greater pressure on the walls of the enclosing vessel. By measuring this pressure, and/or by allowing the fluid to expand under constant pressure, the temperature of the fluid may be inferred.

Class I and Class V systems use a liquid fill fluid (class V is mercury). Here, the volumetric expansion of the liquid drives an indicating mechanism to show temperature:

Figure 11.4: Diagram of a liquid filled-bulb temperature sensor

Class III systems use a gas fill fluid instead of liquid. Here, the change in pressure with temperature (as described by the Ideal Gas Law) allows the bulb's temperature to be sensed:

Figure 11.5: Diagram of a gas filled-bulb temperature sensor

In these systems, it is quite critical that the tube connecting the sensing bulb to the indicating element be of minimal volume, so the fluid expansion is primarily due to changes in temperature

at the bulb rather than changes in temperature along the length of the tube. It is also important to realize that the fluid volume contained by the bellows (or bourdon tube or diaphragm, etc.) is also subject to expansion and contraction due to temperature changes at the indicator. This means the temperature indication varies somewhat as the indicator temperature changes, which is not desirable, since the device is intended to measure temperature (exclusively) at the bulb. Various methods of compensation exist for this effect (for example, a bi-metal spring inside the indicator mechanism to automatically offset the indication as ambient temperature changes), but it may be permanently offset through a simple "zero" adjustment provided that the ambient temperature at the indicator does not change much.

A fundamentally different class of filled-bulb system is the Class II, which uses a volatile liquid/vapor combination to generate a temperature-dependent fluid expansion:

Figure 11.6: Diagrams of Class II filled-bulb temperature sensors

Given that the liquid and vapor are in direct contact with each other, the pressure in the system will be precisely equal to the *saturated vapor pressure* at the vapor/liquid interface. This makes the Class II system sensitive to temperature only at the bulb and nowhere else along the system's volume. Because of this phenomenon, a Class II filled-bulb system requires no compensation for temperature changes at the indicator.

Class II systems do have one notable idiosyncrasy, though: they have a tendency to switch from Class IIA to Class IIB when the temperature of the sensing bulb crosses the ambient temperature at the indicator. Simply put, the liquid tends to seek the colder portion of a Class II system while the vapor tends to seek the warmer portion. This causes problems when the indicator and sensing bulb exchange identities as warmer/colder. The rush of liquid up (or down) the capillary tubing as the system tries to reach a new equilibrium causes intermittent measurement errors. Class II filled-bulb systems designed to operate in either IIA or IIB mode are classified as *IIC*.

One calibration problem common to all systems with liquid-filled capillary tubes is an offset in temperature measurement due to hydrostatic pressure (or suction) resulting from a different in height between the measurement bulb and the indicator. This represents a "zero" shift in calibration, which may be permanently offset by a "zero" adjustment at the time of installation. Class III (gas-filled) and Class IIB (vapor-filled) systems, of course, suffer no such problem because there is no liquid in the capillary tube to generate a pressure.

## 11.3 Resistance Temperature Elements

One of the simplest classes of temperature sensor is one where temperature effects a change in electrical resistance. With this type of primary sensing element, a simple ohmmeter is able to function as a thermometer, interpreting the resistance as a temperature measurement:

Figure 11.7: Diagram of a resistance temperature sensor

### 11.3.1 Thermistors

*Thermistors* are devices made of metal oxide which either increase in resistance with increasing temperature (a *positive temperature coefficient*) or decrease in resistance with increasing temperature (a *negative temperature coefficient*). *RTD*s are devices made of pure metal (usually platinum or copper) which always increase in resistance with increasing temperature. The major difference between thermistors and RTDs is linearity: thermistors are highly sensitive and nonlinear, whereas RTDs are relatively insensitive but very linear. For this reason, thermistors are typically used where high accuracy is unimportant. Many consumer-grade devices use thermistors for temperature sensors.

### 11.3.2 Resistance Temperature Detectors (RTDs)

Resistance Temperature Detectors (RTDs) relate resistance to temperature by the following simplified formula:

$$R_T = R_{ref}[1 + \alpha(T - T_{ref})]$$

Where,
$R_T$ = Resistance of RTD at given temperature $T$ ($\Omega$)
$R_{ref}$ = Resistance of RTD at the reference temperature $T_{ref}$ ($\Omega$)
$\alpha$ = Temperature coefficient of resistance ($^\circ\mathrm{C}^{-1}$)

The following example shows how to use this formula to calculate the resistance of a 100 Ωplatinum RTD with a temperature coefficient value of 0.00392 °C$^{-1}$ at a temperature of 35 °C:

$$R_T = 100\,\Omega[1 + (0.00392\ °C^{-1})(35\ °C - 0\ °C)]$$

$$R_T = 100\,\Omega[1 + 0.1372]$$

$$R_T = 100\,\Omega[1.1372]$$

$$R_T = 113.72\,\Omega$$

Due to nonlinearities in RTD behavior, this linear RTD formula is only an approximation. A better approximation is the *Callendar-van Dusen formula*, which introduces second, third, and fourth-degree terms for a better fit: $R_T = R_{ref}(1 + AT + BT^2 - 100CT^3 + CT^4)$ for temperatures ranging -200 °C $< T <$ 0 °C and $R_T = R_{ref}(1 + AT + BT^2)$ for temperatures ranging 0 °C $< T <$ 661 °C, both assuming $T_{ref} = 0$ °C.

Water's melting/freezing point is the standard reference temperature for most RTDs. Here are some typical values of $\alpha$ for common metals used in RTDs:

- Nickel = 0.00672 °C$^{-1}$
- Platinum (US) = 0.00392 °C$^{-1}$
- Platinum (EU) = 0.00385 °C$^{-1}$

A very common reference resistance ($R_{ref}$) for industrial RTDs is 100 Ω. Another common reference resistance 1000 Ω, is mostly found in HVAC applications. Compared to thermistors with their tens or even hundreds of thousands of ohms' nominal resistance, an RTD's resistance is comparatively small. This can cause problems with measurement, since the wires connecting an RTD to its ohmmeter possess their own resistance, which will be a more substantial percentage of the total circuit resistance than for a thermistor.

Figure 11.8 is a schematic diagrams show the relative effects of 2 Ω total wire resistance on a thermistor circuit and on an RTD circuit.

Figure 11.8: Two-wire temperature sensor outputs are affected by lead resistance

Clearly, wire resistance is more problematic for low-resistance RTDs than for high-resistance thermistors. In the RTD circuit, wire resistance counts for 1.96% of the total circuit resistance. In the thermistor circuit, the same 2 Ω of wire resistance counts for only 0.004% of the total circuit resistance. The thermistor's huge reference resistance value "swamps"[1] the wire resistance to the point that the latter becomes insignificant by comparison.

In HVAC (Heating, Ventilation, and Air Conditioning) systems, where the temperature measurement range is relatively narrow, the nonlinearity of thermistors is not a serious concern and their relative immunity to wire resistance is a decided advantage over RTDs. In industrial temperature measurement applications where the temperature ranges are usually much wider, the nonlinearity of thermistors is a significant problem, so use of RTDs is preferred but requires overcoming the (lesser) problem of wire resistance.

A *three-wire* RTD connection is superior to the two-wire connection because the lead resistances cancel out when they are balanced (i.e., equal in each wire). A three-wire connection is shown in Figure 11.9.

[1] "Swamping" is the term given to the overshadowing of one effect by another. Here, the normal resistance of the high-value RTD greatly overshadows any wire resistance, such that wire resistance becomes negligible.

Figure 11.9: Three-wire temperature sensor outputs are not affected by lead resistance

In a three-wire RTD circuit, voltmeter "A" measures the voltage dropped across the RTD (plus the voltage dropped across the bottom current-carrying wire). Voltmeter "B" measures just the voltage dropped across the top current-carrying wire. Both current-carrying wires can be cut to length to have (very nearly) the same resistance so their effects cancel, and subtracting the indication of voltmeter "B" from the indication given by voltmeter "A" yields the voltage dropped across the RTD.

$$V_{RTD} = V_{\text{meter(A)}} - V_{\text{meter(B)}}$$

Of course, real RTD instruments do not typically employ direct-indicating voltmeters. Most often, the voltage-measuring element is an analog-to-digital converter (ADC) which sends a digital output to a microprocessor for processing and signal output and/or display. Analog electronic RTD instruments have also been built using operational amplifiers to convert the RTD's voltage drop into a standard instrument output signal, such as 4-20 mA DC. The voltmeters shown in the previous diagrams serve only to illustrate the basic concepts.

A very old electrical technique known as the *Kelvin* or *four-wire* method is a one of the best solutions for the problem of lead resistance. Commonly employed to make precise resistance measurements for scientific experiments in laboratory conditions, the four-wire technique uses four wires to connect the resistance under test (in this case, the RTD) to the measuring instrument:

Figure 11.10: Four-wire temperature sensor outputs are not affected by lead resistance

Current is supplied to the RTD from a current source, whose job it is to precisely regulate current regardless of circuit resistance. A high-impedance input voltmeter measures the voltage dropped across the RTD, and Ohm's Law is used to calculate the resistance of the RTD ($R = \frac{V}{I}$).

None of the wire resistances are consequential in this circuit. The two wires carrying current to the RTD will drop some voltage along their length, but this is of no concern because the voltmeter only "sees" the voltage dropped across the RTD. The two wires connecting the voltmeter to the RTD have resistance, but drop negligible voltage because the voltmeter draws so little current through them (remember an ideal voltmeters has infinite input impedance, and modern semiconductor-amplified voltmeters have impedances of several $M\Omega$ or more).

The only disadvantage of the four-wire method is the sheer number of wires necessary. Four wires *per RTD* can add up to a sizeable wire count when many different RTDs are involved on the same process. Wires cost money, and occupy expensive conduit, so there are situations where the four-wire method is a burden.

Connecting the element leads to a temperature transmitter in a protective connection head mitigates most of the problems caused by lead resistance by making the leads to the element very short (hence low resistance) and transmitting a current signal to the receiving device instead of trying to read voltage or resistance directly. RTDs are often purchased as a complete unit made up of sensing element, transmitter, and protection head.

One problem inherent to both thermistors and RTDs is *self-heating*. In order to measure the resistance of either device, an electric current must pass through it. Unfortunately, this results in the generation of heat at the resistance according to Joule's Law:

$$P = I^2 R$$

This dissipated power causes the thermistor or RTD to increase in temperature beyond its surrounding environment, introducing a positive measurement error. The effect may be minimized by limiting excitation current to a bare minimum, but this results in less voltage dropped across the device. The smaller the developed voltage, the more sensitive the voltage-measuring instrument must be to accurately sense the condition of the resistive element. Furthermore, a decreased signal

voltage means a decreased signal-to-noise ratio, for any given amount of noise induced in the circuit from external sources.

One clever way to circumvent the self-heating problem without diminishing excitation current to the point of uselessness is to *pulse* current through the resistive sensor and digitally sample the voltage only during those brief time periods while the thermistor or RTD is powered. This technique works well when slow sample rates from the temperature instrument are able to be tolerated, which is often the case because most temperature measurement applications are slow-changing by nature. The pulsed-current technique enjoys the further advantage of reducing power consumption for the instrument, an important factor in battery-powered temperature measurement applications.

## 11.4 Thermocouples

When two dissimilar metal wires are joined together at one end, a voltage is produced at the free end that is approximately proportional to the temperature gradient that exists between the two ends. This form of electrical temperature sensor is called a *thermocouple*. Note, RTDs are completely passive sensing elements, requiring the application of an externally-sourced electric current in order to function as temperature sensors. Thermocouples, however, generate their own electric potential. In some ways, this makes thermocouple systems simpler because the device receiving the thermocouple's signal does not have to supply electric power to the thermocouple. The self-powering nature of thermocouples also means they do not suffer from the same "self-heating" effect as RTDs. In other ways, thermocouple circuits are more complex than RTD circuits because the generation of voltage actually occurs in two different locations within the circuit, not simply at the sensing point. This means the receiving circuit must "compensate" for temperature in another location in order to accurately measure temperature in the desired location. Though typically not as accurate as RTDs, thermocouples are more rugged, have greater temperature measurement spans, and are easier to manufacture in different physical forms.

### 11.4.1 The Seebeck Effect

The dominant physical phenomenon that makes thermocouples function is known as the Seebeck effect. Electromotive force is generated in a homogeneous wire when the two ends are exposed to a temperature gradient.

$$E = \int_{T_b}^{T_a} \sigma dT$$

where $T_a$ and $T_b$ are the temperatures at the two ends of a material and $\sigma$ is the Seebeck coefficient of the material as a function of temperature:

$$\sigma\left(T\right) = \frac{dE}{dT}$$

The chart below shows the variation of the Seebeck coefficient as a function of temperature for platinum:

Figure 11.11: Plot of the Seebeck coefficient vs. temperature for platinum

The previous equation can also be rewritten:

$$dE = \sigma\left(T\right) dT$$

Differentiating with respect to an incremental length of the material $dx$ over which the differential temperature $dx$ must occur produces:

$$\frac{dE}{dx} = \sigma\left(T\right) \frac{dT}{dx}$$

This equation shows that the emf is produced along the entire length of the wire as a function of the temperature gradient.

The Seebeck effect is produced by a combination of two other effects: the Peltier effect and the Thomson effect.

### 11.4.2 The Peltier Effect

A junction of two dissimilar metals is heated or cooled by electric current flowing through it.

### 11.4.3 The Thomson Effect

A homogeneous current-carrying conductor in a temperature gradient absorbs or produces heat independent of resistive heating effects.

### 11.4.4 Metal Junctions

If two similar homogeneous wires are joined together at one end and a temperature gradient exists between the junction end and the free end, emfs that are produced by each wire are equal and therefore cancel to produce no net emf at the free end.

However, when two heterogeneous wires are joined together at one end and a temperature gradient exists between the end where they are joined (known as the *junction*) and the free end of the wires, the emfs that are produced by each wire are not equal due to the unequal Seebeck coefficients of the materials. Therefore, a net emf is produced at the free end. This is the fundamental principle of thermocouple thermometry. This phenomenon provides a simple and direct way to electrically infer temperature: simply measure the voltage produced at the free end of the wires (at a different temperature than the junction), and the temperature of the junction end can be determined.

Figure 11.12: Diagram of a thermocouple

### 11.4.5 Laws of Thermocouples

#### Law of Homogeneous Metals

"A thermoelectric current cannot be sustained in a circuit of a single homogeneous material, however varying in cross-section, by the application of heat alone."

Temperature measurement would be simple using thermocouples if it were not for an unavoidable consequence of electric circuits: when any kind of electrical instrument is connected to the wires, it inevitably produces another junction of dissimilar metals. The schematic in Figure 11.13 shows this fact:

Figure 11.13: Diagram of a thermocouple showing additional junctions created by other metals

Junction $J_1$ is a junction of iron and copper – two dissimilar metals – and these wires will generate a voltage related to the temperature gradient between the junction and their ends. Note that junction $J_2$, which is necessary to connect the copper-wired voltmeter to the iron wire, is also a dissimilar-metal junction connecting wires that will generate a voltage related to temperature gradient between that junction and the ends of its wires. Note also how the polarity of the emf produced by the wires of junction $J_2$ stands opposed to the polarity of the emf produced by the wires of junction $J_1$ (iron = positive; copper = negative). A third junction ($J_3$) also exists between wires, but it is of no consequence because it is a junction of two identical metals and therefore these wires not generate a temperature-dependent voltage at all.

The presence of this second junction ($J_2$) helps explain why the voltmeter registers 0 volts when the entire system is at room temperature: any voltage generated by the iron-copper wires will be equal in magnitude and opposite in polarity, resulting in a net (series-total) voltage of zero. It is only when the two junctions $J_1$ and $J_2$ are at different temperatures that the voltmeter registers any voltage at all.

This relationship may be expressed mathematically as follows:

$$V_{meter} = V_{J1} - V_{J2}$$

With the measurement ($J_1$) and reference ($J_2$) junctions created from wires generating emfs with polarities opposed to each other, the voltmeter only "sees" the difference between these two voltages.

Thus, thermocouple systems are fundamentally *differential* temperature sensors. That is, they provide an electrical output proportional to the temperature gradient between two different points. For this reason, the wire junction used to measure the temperature of interest is called the *measurement junction* while the other junction (which cannot be eliminated from the circuit) is called the *reference junction* (or the *cold junction*, because it is typically at a cooler temperature than the process measurement junction).

### Law of Intermediate Temperatures

"If two dissimilar homogeneous metals produce a thermal emf of $E_A$ when the thermal gradient is defined by temperatures $T_1$ and $T_2$, and a thermal emf of $E_B$ when the thermal gradient is defined by $T_2$ and $T_3$, the emf generated when the thermal gradient is defined by $T_1$ and $T_3$ will be $E_A + E_B$."

**Law of Intermediate Metals**

"The algebraic sum of the thermoelectromotive forces in a circuit composed of any number of dissimilar materials is zero if all of the circuit is at a uniform temperature."

It is critical to realize that the phenomenon of a "reference junction" is an inevitable effect of having to close the electric circuit loop in a circuit made of dissimilar metals. This is true regardless of the number of metals involved. In the last example, only two metals were involved: iron and copper. This formed one iron-copper junction ($J_1$) at the measurement end and one iron-copper junction ($J_2$) at the indicator end. Recall that the copper-copper junction $J_3$ was of no consequence because its identical metallic composition generates no thermal voltage:

Figure 11.14: Diagram of a thermocouple with two dissimilar wiring junctions

Here there are *three* voltage-generating junctions: $J_1$ of iron and constantan, $J_2$ of iron and copper, and $J_3$ of copper and constantan which just happens to be the metallic combination for a type T thermocouple. Upon first inspection it would seem to be a much more complex situation than the one with just two metals (iron and copper), but the situation is actually just as simple as it was before.

A principle of thermo-electric circuits called the *Law of Intermediate Metals* helps illustrate this clearly. According to this law, intermediate metals in a series of junctions are of no consequence to

the overall (net) voltage so long as those intermediate junctions are all at the same temperature. Representing this pictorially, the net effect of having four different metals (A, B, C, and D) joined together in series is the same as just having the first and last metal in that series (A and D) joined with one junction, if all intermediate junctions are at the same temperature:

Figure 11.15: Illustration of the Law of Intermediate Metals

In the Type J thermocouple circuit where iron and constantan both join to copper, copper is an intermediate metal so long as junctions $J_2$ and $J_3$ are at the same temperature. Since those two junctions are located next to each other on the indicating instrument, identical temperature is a reasonable assumption, and junctions $J_2$ and $J_3$ may be treated as a single iron-constantan reference junction. In other words, the Law of Intermediate Metals says to treat these two circuits identically:

Figure 11.16: Illustration of the Law of Intermediate Metals

The practical importance of this Law is that the reference junction(s) can always be treated as a single junction made from the same two metal types as the measurement junction, so long as all dissimilar metal junctions at the reference location are at the same temperature.

This fact is extremely important in the age of semiconductor circuitry, where the connection of a thermocouple to an electronic amplifier involves many different junctions, from the thermocouple wires to the amplifier's silicon. Here there are a multitude of reference junctions, inevitably formed by the necessary connections from thermocouple wire to the silicon substrate inside the amplifier chip:

Figure 11.17: Illustration of the Law of Intermediate Metals with multiple reference junctions

It should be obvious that each complementary junction pair cancels if each pair is at the same temperature (e.g., gold-silicon junction $J_{12}$ cancels with silicon-gold junction $J_{13}$ because they generate the exact same amount of voltage with opposing polarities). The Law of Intermediate Metals goes one step further by indicating that junctions $J_2$ through $J_{13}$ taken together in series are of the same effect as a single reference junction of iron and constantan. Automatic reference junction compensation is as simple as counter-acting the voltage produced by this equivalent iron-constantan junction at whatever temperature exists at junctions $J_2$ through $J_{13}$.

**Reference junction compensation** Multiple techniques exist to deal with the influence of the reference junction's temperature. One technique is to physically fix the temperature of that junction at some constant value so it is always stable. This way, any changes in measured voltage *must* be due to changes in temperature at the measurement junction, since the reference junction has been rendered incapable of changing temperature. This may be accomplished by immersing the reference junction in a bath of ice and water:

Figure 11.18: Reference junction compensation using an ice bath

In fact, this is how thermocouple temperature/voltage tables are referenced: describing the amount of voltage produced for given temperatures at the measurement junction with the reference junction held at the freezing point of water (0 °C = 32 °F).

However, this is not a very practical solution for dealing with the reference junction's voltage. Instead, an additional electrical circuit can be applied to counteract the voltage produced by the reference junction. This is called a *reference junction compensation* or *cold junction compensation (CJC)* circuit:

Figure 11.19: Cold junction compensation using a voltage source

Please note that "cold junction" is just a synonymous label for "reference junction." In fact the "cold" reference junction may very well be at a warmer temperature than the so-called "hot" measurement junction! Nothing prevents anyone from using a thermocouple to measure temperatures below freezing.

This compensating voltage source ($V_{rjc}$ in the above schematic) uses some other temperature-sensing device such as a thermistor or RTD to sense the local temperature at the terminal block where junction $J_2$ is formed, and produce a counter-voltage that is precisely equal and opposite to $J_2$'s voltage ($V_{rjc} = V_{J2}$). Having canceled the effect of the reference junction, the voltmeter now only registers the voltage produced by the measurement junction $J_1$:

$$V_{meter} = V_{J1} - V_{J2} + V_{rjc}$$

$$V_{meter} = V_{J1} + 0$$

$$V_{meter} = V_{J1}$$

At first it may seem pointless to go through the trouble of building a reference junction compensation circuit. After all, why bother to do this just to be able to use a thermocouple to accurately measure temperature, when this "other" device (thermistor, RTD, etc.) could simply be used to directly measure the temperature of interest? In other words, isn't the usefulness of a thermocouple invalidated if it requires integrating another type of electrical temperature sensor in the circuit just to compensate for an idiosyncrasy of thermocouples?

The answer to this very good question is that thermocouples enjoy certain advantages over these other sensor types. Thermocouples are extremely rugged and have far greater temperature-measurement ranges than thermistors, RTDs, and other primary sensing elements. However, if the application does not demand extreme ruggedness or large measurement ranges, a thermistor or RTD would likely be the better choice.

**Software compensation** Previously, it was suggested this automatic compensation could be accomplished by intentionally inserting a temperature-dependent voltage source in series with the circuit, oriented in such a way as to oppose the reference junction's voltage. This technique is known as *hardware* compensation. A stand-alone circuit designed to do this is sometimes called an *ice point*, because it electrically accomplishes the same thing as physically placing the reference junction(s) in a bath of ice-water.

A more modern technique for reference junction compensation is called *software* compensation. This is applicable only where the indicating device is microprocessor-based, and where an additional analog input channel exists:

Figure 11.20: Cold junction compensation using software

Instead of canceling the effect of the reference junction electrically, it can be canceled mathematically inside the microprocessor. In other words, let the meter see the difference in voltage between the measurement and reference junctions ($V_{meter} = V_{J1} - V_{J2}$). After digitizing this voltage measurement, the microprocessor adds the equivalent voltage value corresponding to the ambient temperature sensed by the RTD or thermistor ($V_{rjc}$):

$$\text{Compensated total} = V_{meter} + V_{rjc}$$

$$\text{Compensated total} = (V_{J1} - V_{J2}) + V_{rjc}$$

Since the calculated value of $V_{rjc}$ should be equal to the real reference junction voltage ($V_{J2}$), the result of this digital addition should be a compensated total equal only to the measurement junction voltage $V_{J1}$:

$$\text{Compensated total} = V_{J1} - V_{J2} + V_{rjc}$$

$$\text{Compensated total} = V_{J1} + 0$$

$$\text{Compensated total} = V_{J1}$$

Perhaps the greatest advantage of software compensation is flexibility. Being able to re-program the compensation function means this instrument may easily interpret the voltage output of different thermocouple types with no modifications to the hardware. So long as the microprocessor memory is programmed with look-up tables relating voltage values to temperature values, it may accurately measure (and compensate for the reference junction of) any thermocouple type. With hardware-based compensation (an "ice point" circuit), rewiring or replacement is necessary to accommodate different thermocouple types.

**Side-effects of reference junction compensation** Reference junction compensation is a necessary part of any precision thermocouple circuit, due to the inescapable fact of the reference junction's existence. When forming a complete circuit of dissimilar metals, the circuit *will* form both a measurement junction and a reference junction, with those two junctions' polarities opposed to one another. This is why reference junction compensation – whether it takes the form of a hardware circuit or an algorithm in software – must exist within every precision thermocouple instrument.

The presence of reference junction compensation in every precision thermocouple instrument leads to an interesting phenomenon: if the thermocouple input terminals of such an instrument are directly short-circuited it will register ambient temperature. This behavior may be illustrated by example, first showing a normal operating temperature measurement system and then with that same system short-circuited. Here a temperature indicator received a 4-20 mA current signal from a temperature transmitter, which is receives a millivoltage signal from a type "K" thermocouple sensing a process temperature of 780 °F:

Figure 11.21: A thermocouple system with temperature transmitter having CJC

The transmitter's internal reference junction compensation feature compensates for the ambient temperature of 68 °F. If the ambient temperature rises or falls, the compensation will automatically adjust for the change in reference junction potential, such that the output will still register the process (measurement junction) temperature of 780 °F. This is what the reference junction compensation is designed to do.

Now, disconnect the thermocouple from the temperature transmitter and short-circuit the transmitter's input:

Figure 11.22: A temperature transmitter that has CJC with its inputs shorted

With the input short-circuited, the transmitter "sees" no voltage at all from the thermocouple circuit. There is no measurement junction nor a reference junction to compensate for, just a piece of wire making both input terminals electrically common. This means the reference junction compensation inside the transmitter no longer performs a useful function. However, the transmitter does not "know" it is no longer connected to the thermocouple, so the compensation keeps on working even though it has nothing to compensate for. Recall the voltage equation relating measurement, reference, and compensation voltages in a hardware-compensated thermocouple instrument:

$$V_{meter} = V_{J1} - V_{J2} + V_{rjc}$$

Disconnecting the thermocouple wire and connecting a shorting jumper to the instrument eliminates the $V_{J1}$ and $V_{J2}$ terms, leaving only the compensation voltage to be read by the meter[2]:

$$V_{meter} = 0 + V_{rjc}$$

$$V_{meter} = V_{rjc}$$

[2]The effect will be exactly the same for an instrument with software compensation rather than hardware compensation. With software compensation, there is no literal $V_{rjc}$ voltage source, but the equivalent millivolt value is digitally added to the zero input measured at the thermocouple connection terminals, resulting in the same effect of measuring ambient temperature.

This is why the instrument registers the equivalent temperature created by the reference junction compensation feature: this is the only signal it "sees" with its input short-circuited. This phenomenon is true regardless of which thermocouple type the instrument is configured for, which makes it a convenient "quick test" of instrument function in the field. If one short-circuits the input terminals of any thermocouple instrument, it should respond as though it is sensing ambient temperature.

While this interesting trait is a somewhat useful side-effect of reference junction compensation in thermocouple instruments, there are other effects that are not quite so useful. The presence of reference junction compensation becomes quite troublesome, for example, if one tries to simulate a thermocouple using a precision millivoltage source. Suppose, for example, a technician wished to simulate a type K thermocouple at 300 °F by setting a millivolt source to 6.094 millivolts (the voltage corresponding to 300 °F for type K thermocouples according to the ITS-90 standard). Connecting the millivolt source to the instrument will *not* result in an instrument response appropriate for 300 °F:

Figure 11.23: Improper simulation of a thermocouple with a millivolt source

Instead, the instrument registers 339 degrees because its internal reference junction compensation feature is still active, compensating for a reference junction voltage that no longer exists. The millivolt source's output of 6.094 mV gets *added* to the compensation voltage (inside the transmitter) of 0.865 mV – the necessary millivolt value to compensate for a type K reference junction at 71 °F – with the result being a larger millivoltage (6.959 mV) interpreted by the transmitter as a temperature of 339 °F.

The only way to properly use a millivoltage source to simulate a desired temperature is for the instrument technician to "out-think" the transmitter's compensation feature by specifying a millivolt signal that is offset by the amount of equivalent voltage generated by the transmitter's compensation. In other words, instead of setting the millivolt source to a value of 6.094 mV, the technician should set the source to only 5.229 mV so the transmitter will add 0.865 mV to this value to arrive at 6.094 mV and register as 300 °F:

Figure 11.24: Correct simulation of a thermocouple with a millivolt source

Years ago, the only suitable piece of test equipment available for generating the precise millivoltage signals necessary to calibrate thermocouple instruments was a device called a *precision potentiometer*. These potentiometers used a stable *mercury cell* battery (sometimes called a *standard cell*) as a voltage reference and a potentiometer with a calibrated knob to output low-voltage signals. Photographs of two vintage precision potentiometers are shown in Figure 11.25.

(a) Rubicon precision potentiometer (b) Sheppard precision potentiometer

Figure 11.25: Vintage precision potentiometers

Of course, modern thermocouple calibrators also provide direct entry of temperature and automatic compensation to "un-compensate" the transmitter such that any desired temperature may be easily simulated:

Figure 11.26: Diagram showing use of a thermocouple simulator

In this example, when the technician sets the calibrator for 300 °F (type K), it measures the ambient temperature and automatically subtracts 0.865 mV from the output signal, so only 5.229 mV is sent to the transmitter terminals instead of the full 6.094 mV. The transmitter's internal

reference junction compensation adds the 0.865 mV offset value (thinking it must compensate for a reference junction that in reality is not there) and "sees" a total signal voltage of 6.094 mV, interpreting this properly as 300 °F.

The following photograph shows the display of a modern thermocouple calibration device (a Fluke model 744 documenting process calibrator) being used to generate a thermocouple signal. In this particular example, the thermocouple type is set to type "S" (Platinum-Rhodium/Platinum) at a temperature of 2650 °F:

Figure 11.27: Display of a digital thermocouple simulator

The ITS-90 thermocouple standard declares a millivoltage signal value of 15.032 mV for a type S thermocouple junction at 2650 °F (with a reference junction temperature of 32 °F). Note how the calibrator does *not* output 15.032 mV even though the simulated temperature has been set to 2650 °F. Instead, it outputs 14.910 mV, which is 0.122 mV less than 15.032 mV. This offset of 0.122 mV corresponds to the calibrator's local temperature of 70.8 °F (according to the ITS-90 standard for type S thermocouple junctions).

When the calibrator's 14.910 mV signal reaches the thermocouple instrument being calibrated (be it an indicator, transmitter, or even a controller equipped with a type S thermocouple input), the instrument's own internal reference junction compensation will add 0.122 mV to the received signal of 14.910 mV, "thinking" it needs to compensate for a real reference junction. The result will be a perceived measurement junction signal of 15.032 mV, which is exactly what the instrument should "think" it sees if the goal is to simulate connection to a real type S thermocouple at a temperature of 2650 °F.

### 11.4.6 Thermocouple types

Thermocouples exist in many different types depending on the metals used for each wire. Thermocouples are differentiated by giving each positive wire its own color codes for the dissimilar-metal wires. In the U.S., the color-coded insulation is defined by ISA MC96.1-1982 "Temperature Measurement Thermocouples." The negative wire of a pair is always red. An overall insulation layer is also provided. It is either brown or brown with a color stripe that matches the positive wire insulation color. Here is a table showing the more common thermocouple types:

| Type | Positive wire | Negative wire | Temperature range |
|---|---|---|---|
| J | iron (white) | constantan (red) | 32 to 1382 °F |
| K | chromel (yellow) | alumel (red) | -328 to 2282 °F |
| T | copper (blue) | constantan (red) | -328 to 662 °F |
| E | chromel (purple) | constantan (red) | -328 to 1652 °F |
| N | Ni/Cr/Si (orange) | Platinum (red) | -450 to 2372 °F |

When thermocouple signals need to be run long distances without using a temperature transmitter, thermocouple-grade extension wire must be used that matches the type of thermocouple[3]. Extension wire properly transfers the junction to the other end of the wiring without creating emfs from reference junctions since it is made of the same materials (though cheaper alloys) as the thermocouple itself. Extension wire for a given type of thermocouple is generally indicated by an "X" appended to the type letter, e.g., "JX" is extension wire for use with type J thermocouples. Since extension wires can act like antennas and pick up noise in the millivolt range, shielded thermocouple extension wire with the shield grounded at one end should be used in installations where the wire can be exposed to electromagnetic interference.

### 11.4.7 Burnout detection

Another consideration for thermocouples is *burnout detection*. The most common failure mode for thermocouples is to fail open, otherwise known as "burning out." An open thermocouple is problematic for any voltage-measuring instrument with high input impedance because the lack of a complete circuit on the input makes it possible for electrical noise from surrounding sources (power lines, electric motors, variable-frequency motor drives) to be detected by the instrument and falsely interpreted as a wildly varying temperature.

For this reason it is prudent to design into the thermocouple instrument some provision for generating a consistent state in the absence of a complete circuit. This is called the *burnout mode* of a thermocouple instrument.

[3] To avoid induced voltages from noise and other concerns, always use a temperature transmitter that has CJC as close to the thermocouple as possible instead of extension wires.

Figure 11.28: Diagram showing a thermocouple burnout mode

The resistor in this circuit provides a path for current in the event of an open thermocouple. It is sized in the megohm range to minimize its effect during normal operation when the thermocouple circuit is complete. Only when the thermocouple fails open will the miniscule current through the resistor have any substantial effect on the voltmeter's indication. The SPDT switch provides a selectable burnout mode: in the event of a burnt-out thermocouple, it can configure the meter to either read high temperature (sourced by the instrument's internal milli-voltage source) or low temperature (grounded), depending on what failure mode is deemed safest for the application.

### 11.4.8 Connector and tip styles

### 11.4.9 Connector and tip styles

In its simplest form, a thermocouple is nothing more than a pair of dissimilar-metal wires joined together. However, in industrial practice, we often need to package thermocouples in a way that optimizes their ruggedness and reliability. For instance, most industrial thermocouples are manufactured in such a way that the dissimilar-metal wires are protected from physical damage by a stainless steel or ceramic *sheath*, and they are often equipped with molded-plastic plugs for quick connection to and disconnection from a thermocouple-based instrument.

A photograph of a type K industrial thermocouple (approximately 20 inches in length) reveals this "sheathed" and "connectorized" construction:

The stainless steel sheath of this particular thermocouple shows signs of discoloration from previous service in a hot process. Note the different diameters of the plug terminals. This "polarized" design makes it difficult[4] to insert backward into a matching socket.

[4]It should be noted that no amount of engineering or design is able to *completely* prevent people from doing the wrong thing. I have seen this style of thermocouple plug forcibly mated the wrong way to a socket. The amount of insertion force necessary to make the plug fit backward into the socket was quite extraordinary, yet this apparently was not enough of a clue for this wayward individual to give them pause.

A miniature version of this same plug (designed to attach to thermocouple wire by screw terminals, rather than be molded onto the end of a sheathed assembly) is shown here, situated next to a ballpoint pen for size comparison:

Industrial-grade thermocouples are available with this miniature style of molded plug end as an alternative to the larger (standard) plug. Miniature plug-ends are often the preferred choice for laboratory applications, while standard-sized plugs are often the preferred choice for field applications.

Some industrial thermocouples have no molded plug at all, but terminate simply in a pair of open wire ends. The following photograph shows a type J thermocouple of this construction:

If the electronic measuring instrument (e.g., temperature transmitter) is located near enough for the thermocouple's wires to reach the connection terminals, no plug or socket is needed at all in the circuit. If, however, the distance between the thermocouple and measuring instrument is too far to span with the thermocouple's own wires, a common termination technique is to attach a special terminal block and connection "head" to the top of the thermocouple allowing a pair of thermocouple extension wires to join and carry the millivoltage signal to the measuring instrument.

This next photograph shows a close-up view of such a thermocouple "head":

As you can see from this photograph, the screws directly press against the solid metal wires to make a firm connection between each wire and the brass terminal block. Since the "head" attaches directly to one end of the thermocouple, the thermocouple's wires will be trimmed just long enough to engage with the terminal screws inside the head. A threaded cover provides easy access to these connection points for installation and maintenance, while ensuring the connections are covered and protected from ambient weather conditions the rest of the time.

Thermocouple wires are most often manufactured in solid form rather than stranded form. A common mistake made with thermocouple wires is for technicians to crimp compression-style terminals ("lugs") onto the solid wires. While this may form a usable connection at first, compression-style terminals are simply unable to maintain adequate compression when applied to solid wire of any type, thermocouple wire included. Over time, solid wires *will* loosen inside compression terminals leading to circuit problems. In the case of a thermocouple circuit, bad wire connections lead to a situation where the receiving instrument "thinks" the thermocouple has failed open. This situation is commonly called *burnout*, referring to the phenomenon where a thermocouple junction fails open from being "burned out" by excessive temperature.

You will most often find compression terminals (improperly) applied to solid thermocouple wire tips where those wires must terminate under the head of a screw. Compression terminals are correct to use in applications where stranded wire terminates at a screw head, but not solid wire. The

proper termination technique for solid wire under a screw head is to wrap the solid wire clockwise at least three-quarters of the way around the screw post and directly clamp it under the screw head.

At the other end of the thermocouple, we have a choice of tip styles. For maximum sensitivity and fastest response, the dissimilar-metal junction may be unsheathed (bare). This design, however, makes the thermocouple more fragile. Sheathed tips are typical for industrial applications, available in either *grounded* or *ungrounded* forms:

Grounded-tip thermocouples exhibit faster response times and greater sensitivity than ungrounded-tip thermocouples, but they are vulnerable to *ground loops*: circuitous paths for electric current between the conductive sheath of the thermocouple and some other point in the thermocouple circuit. In order to avoid this potentially troublesome effect, most industrial thermocouples are of the ungrounded design.

## 11.5 Optical temperature sensing

Virtually any mass above absolute zero temperature will emit electromagnetic radiation (photons, or light) as a function of that temperature. The *Stefan-Boltzmann Law* of radiated energy says that the rate of heat lost by radiant emission from a hot object is proportional to the fourth power of the absolute temperature:

$$\frac{dQ}{dt} = e\sigma AT^4$$

Where,
$\frac{dQ}{dt}$ = Radiant heat loss rate (watts)
$e$ = Emissivity factor (unitless)
$\sigma$ = Stefan-Boltzmann constant ($5.67 \times 10^{-8}$ W / m$^2 \cdot$ K$^4$)
$A$ = Surface area (square meters)
$T$ = Absolute temperature (Kelvin)

This phenomenon provides a way to infer an object's absolute temperature by sensing the radiation it emits. Such a measurement technique holds obvious advantages, perhaps the greatest being the lack of need for direct contact to the process with a sensing element such as an RTD or thermocouple.

Using an array of radiation sensors it is possible to build a *thermal imager*, providing a graphic display of objects in its view according to their temperatures. Each object is artificially colored in the display on a chromatic scale that varies with temperature, hot objects typically registering as red tones and cold objects typically registering as blue tones. Thermal imaging is very useful in the electric power distribution industry, where technicians can check power line insulators and other objects at elevated potential for "hot spots" without having to make physical contact with those objects. Thermal imaging is also useful in performing "energy audits" of buildings and other heated structures, providing a means of revealing points of heat escape through walls, windows, and roofs.

Perhaps the main disadvantage of optical temperature sensors is their inaccuracy. The emissivity factor ($e$) in the Stefan-Boltzmann equation varies with the composition of a substance, but beyond that there are several other factors (surface finish, shape, etc.) that affect the amount of radiation an optical sensor will receive from an object. For this reason, emissivity is not a very practical way to gauge the effectiveness of an optical temperature sensor. Instead, a more comprehensive measure of an object's "thermal-optical measureability" is *emittance*.

A perfect emitter of thermal radiation is known as a *blackbody*. Emittance for a blackbody is unity (1), while emittance figures for any real object is a value between 1 and 0. The only certain way to know the emittance of an object is to test that object's thermal radiation at a known temperature. This assumes that the object's temperature can be measured by direct contact, which of course renders void one of the major purposes of optical thermometry: to be able to measure an object's temperature without having to touch it. Not all hope is lost for optical techniques, though. All that is needed is to obtain an emittance value for that object *one time*, and then an optical temperature sensor may be calibrated for that object's particular emittance so as to measure its temperature in the future without contact.

Beyond the issue of emittance, other idiosyncrasies plague optical techniques as well. Objects also have the ability to *reflect* and *transmit* radiation from other bodies, which taints the accuracy of any optical device sensing the radiation from that body. An example of the former is trying to measure the temperature of a silver mirror using an optical pyrometer: the radiation received by the pyrometer is mostly from other objects, merely *reflected* by the mirror. An example of the latter is trying to measure the temperature of a gas or a clear liquid, and instead primarily measuring the temperature of a solid object in the background (*through* the gas or liquid).

Nevertheless, optical techniques for measuring temperature have been and will continue to be useful in specific applications where other, contact-based techniques are impractical.

## 11.6 Temperature switches

A *temperature switch* is one detecting the temperature of an object. Temperature switches often use bimetallic strips as the temperature-sensing element, the motion of which actuates one or more switch contacts. An alternative design uses a metal bulb filled with a fluid that expands with temperature, causing the switch mechanism to actuate based on the pressure this fluid exerts against a diaphragm or bellows. This latter temperature switch design is really a pressure switch, whose pressure is a direct function of process temperature by virtue of the physics of the entrapped fluid inside the sensing bulb.

Recall from section 9.1 that the "normal" status of a switch is the resting condition of *no stimulation*. A temperature switch will be in its "normal" status when it senses minimum

temperature (i.e., cold, in some cases a condition colder than ambient)[5]. For a temperature switch, "normal" status is any sensed temperature *below* the trip threshold of the switch.

Figure 11.29: Schematic symbol used for temperature switches

The following photograph shows a temperature-actuated switch manufactured by the Ashcroft corporation:

[5]If the trip setting of a temperature switch is below ambient temperature, then it will be "actuated" at ambient temperature and in its "normal" status only when the temperature falls below that trip point (i.e., colder than ambient).

Figure 11.30: A temperature switch

For discrete temperature-sensing applications demanding high accuracy and repeatability, *electronic* temperature switch circuits using thermocouples, RTDs, or thermistors may be used instead of a mechanical (bi-metallic or filled bulb) sensing element. The operation and configuration of discrete electronic temperature switches is very similar to that of continuous electronic temperature transmitters.

## 11.7 Temperature sensor accessories

One of the most important accessories for any temperature-sensing element is a pressure-tight sheath known as a *thermowell*. This may be thought of as a thermally conductive protrusion into a process vessel or pipe that allows a temperature-sensitive instrument to detect process temperature without opening a hole in the vessel or pipe. Thermowells are critically important for installations where the temperature element (RTD, thermocouple, thermometer, etc.) must be replaceable without de-pressurizing the process.

Thermowells may be made out of any material that is thermally conductive, pressure-tight, and

not chemically reactive with the process. A simple diagram showing a thermowell in use with a temperature gauge is shown here:

Figure 11.31: Diagram of a thermometer installed in a protective thermowell

If the temperature gauge is removed for service or replacement, the thermowell maintains pressure integrity of the pipe (no process fluid leaking out, and no air leaking in):

Figure 11.32: Diagram of a thermometer removed from its protective thermowell

Photographs of a real (stainless steel) thermowell are shown here, the left-hand photo showing the entire length of the thermowell, and the right-hand photo showing the end where the temperature-sensing device is inserted:

A photo of a complete RTD assembly (connection head, RTD, and thermowell) appears in the next photograph:

Figure 11.33: A complete RTD assembly with connection head, RTD, and thermowell

Another photo shows an RTD installed in a thermowell on the side of a commercial freezer, using a Rosemount model 3044C temperature transmitter to output a 4-20 mA signal to an operator display:

Figure 11.34: An RTD assembly connected to a temperature transmitter

As useful as thermowells are, they are not without their caveats. First and foremost is the first-order time lag they add to the temperature measurement system by virtue of their mass and specific heat value. It should be intuitively obvious that one or more pounds of metal will not heat up and cool down as fast as a few ounces' worth of RTD or thermocouple, and therefore that the presence of a thermowell will decrease the response time of any temperature-sensing element.

A potential problem with thermowells is incorrect installation of the temperature-sensing element. The element *must* be inserted with full contact at the bottom of the thermowell's blind hole. If any air gap is allows to exist between the end of the temperature element and the bottom of the thermowell's hole, this will add a *second* time lag to the measurement system[6]. Some thermowells include a spring clip in the bottom of the blind hole to help maintain constant metal-to-metal contact between the sensing element and the thermowell wall. Thermal paste or grease can also be used to provide good heat transfer between the thermowell and the sensing element.

There are four main types of connections for thermowells and the process equipment: threaded, flanged, welded, and clamped. The engineer must ensure the type of connection is suitable for the process requirements, e.g., welded connections can withstand the most pressure. Sanitary clamped connections may be required for bioprocessing systems to prevent pockets of biological contamination from forming.

Thermowells should always be analyzed for potential failure due to the von Karman effect. The von Karman effect is caused by turbulent wake vortices that form downstream of the thermowell. These vortices cause pressure fluctuations that impact the thermowell, and the resulting vibrations

[6]The air gap acts as a thermal *resistance* while the mass of the element itself acts as a thermal *capacitance*. Thus, the inclusion of an air gap forms a thermal "RC time constant" delay network secondary to the thermal delay incurred by the thermowell.

can cause failure of the thermowell. Since the thermowell is a pressure boundary, this can result in catastrophic leakage.

## 11.8 Humidity sensors

The dominant type of humidity sensor used in industry today is the capacitive style sensor, though resistive and thermal conductivity sensors are also available.

### 11.8.1 Capacitive humidity sensing

A capacitive humidity sensor is constructed of a thin film of dielectric material (often a porous aluminum oxide) sandwiched between two conductive surfaces (often gold on one side and aluminum substrate on the other side) to form a capacitor. The entire assembly is encased in a porous material that allows moisture to move into the dielectric, thus altering the capacitance of the sensor. This capacitance change is proportional (almost linearly) to the relative humidity of the environment. Typical accuracy reported by manufacturers is around 2%.

### 11.8.2 Resistive humidity sensing

A resistive humidity sensor is constructed of a hygroscopic material. As the material absorbs or releases moisture from the surrounding atmosphere, its resistance changes. The resistance is approximately inversely proportional to the relative humidity. Typical accuracy reported by manufacturers is around 2%.

### 11.8.3 Thermal conductivity humidity sensing

A thermal conductivity humidity sensor is constructed of a two matched thermistor elements in the same leg of a bridge circuit, one of which is sealed in a dry environment (such as nitrogen), the other is exposed to the surrounding atmosphere. Current is passed through each element, and the heat dissipated from the encapsulated thermistor is greater than the heat dissipated from the other thermistor. Since the heat dissipation difference produces a difference in temperature of the elements, a difference in resistance is produced, causing the bridge circuit output voltage to change. Unlike capacitive and resistive humidity sensors, thermal conductivity humidity sensors measure the absolute moisture in the surrounding atmosphere. Typical accuracy reported by manufacturers is around 5% at 40 °C.

## 11.9 Humidity switches

A *humidity switch* is one that switches when the humidity of an object exceeds a setpoint.

Recall from section 9.1 that the "normal" status of a switch is the resting condition of *no stimulation*. A humidity switch will be in its "normal" status when it senses minimum humidity (i.e., dry, in some cases a condition dryer than ambient)[7]. For a humidity switch, "normal" status is any sensed humidity *below* the trip threshold of the switch.

[7] If the trip setting of a humidity switch is below ambient humidity, then it will be "actuated" at ambient humidity and in its "normal" status only when the humidity falls below that trip point (i.e., dryer than ambient).

## 11.10 Process/instrument suitability

The primary consideration for selecting a proper temperature sensing element for any application is the expected temperature range. Mechanical (bi-metal) and filled-system temperature sensors are limited to relatively low process temperatures, and cannot relay signals very far from the point of measurement.

Thermocouples are by far the most rugged and wide-ranging of the contact-type temperature sensors. Accuracies vary with thermocouple type and installation quality.

RTDs are more fragile than thermocouples, but they require no reference compensation and are inherently more linear.

Optical sensors lack the ability to measure temperature of fluids inside vessels unless a transparent window is provided in the vessel for light emissions to reach the sensor. Otherwise, the best an optical sensor can do is report the skin temperature of a vessel. For monitoring surface temperatures of solid objects, especially objects that would be impractical or even dangerous to contact (e.g., electrical insulators on high-voltage power lines), optical sensors are the only appropriate solution.

Chemical reactivity is a concern for contact-type sensors. If the sensing element is held inside a thermowell, that thermowell must be selected for minimum reaction with the process fluid(s). Bare thermocouples are particularly vulnerable to chemical reactions given the nature of most thermocouple metals (iron, nickel, copper, etc.), and must be carefully chosen for the particular process chemistry to avoid reliability problems later.

# References


[1] e. a. A. v. Beckerath, WIKA-Handbook, Pressure and Temperature Measurement. Klingenberg, Germany: WIKA Alexander Wiegand GmbH & Co., 1995.

[2] A. E. Fribance, Industrial Instrumentation Fundamentals. New York, NY: McGraw-Hill, 1962.

[3] J. D. Irwin, Ed., The Industrial Electronics Handbook. Boca Raton, FL: CRC Press, 1997.

[4] H. P. Kallen, Handbook of Instrumentation and Controls. New York, NY: McGraw-Hill, 1961.

[5] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1.

# Chapter 12

# Level Measurement

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Identify the principle of operation of the following types of level instrumentation:
   (a) Level gauges or glasses
   (b) Ball or chain float
   (c) Bubblers
   (d) Differential pressure
   (e) Ultrasonic probes
   (f) Radar level sensors
   (g) Laser level sensors
   (h) Weighing level sensors
   (i) Capacitance probes
   (j) Radiation level sensors
   (k) Resistance probes
   (l) Level switches
2. Draw P&ID symbols used for level detection equipment, including sensing elements, transmitters, and protective diaphragms.
3. Explain the process of density compensation in level detection systems.
4. Given a basic diagram of a differential pressure detector level instrument, discuss the functions of the following:
   (a) Differential pressure (D/P) transmitter
   (b) Amplifier
   (c) Indication
5. Describe the operation and applications of the following types of differential pressure detector level instruments:
   (a) Open tank differential pressure transmitter
   (b) Closed tank, dry reference leg
   (c) Closed tank, wet reference leg

6. Describe the environmental concerns that can affect the accuracy and reliability of level detection instrumentation.
7. Discuss the failure modes of the various types of level indication instruments.

Many industrial processes require the accurate measurement of fluid or solid (powder, granule, etc.) height within a vessel. Some process vessels hold a stratified combination of fluids, naturally separated into different layers by virtue of differing densities, where the height of the *interface* point between liquid layers is of interest.

A wide variety of technologies exist to measure the level of substances in a vessel, each exploiting a different principle of physics. This chapter explores the major level-measurement technologies in current use.

## 12.1 Level gauges (sightglasses)

The *level gauge*, or *sightglass* is to liquid level measurement as manometers are to pressure measurement: a very simple and effective technology for direct visual indication of process level. In its simplest form, a level gauge is nothing more than a clear tube through which process liquid may be seen. The following photograph shows a simple example of a sightglass:

Figure 12.1: A sightglass

A functional diagram of a sightglass shows how it visually represents the level of liquid inside a vessel such as a storage tank:

Figure 12.2: A diagram showing the use of a sightglass

A level gauge is not unlike a U-tube manometer, with equal pressures applied to both liquid columns (one column being the liquid in the gauge sightglass, the other column being the liquid in the vessel).

Level gauge valves exist to allow replacement of the glass tube without emptying or depressurizing the process vessel. These valves are usually equipped with flow-limiting devices in the event of a tube rupture, so too much process fluid does not escape even when the valves are fully open.

Some level gauges called *reflex gauges* are equipped with special optics to facilitate the viewing of clear liquids, which is problematic for simple glass-tube sightglasses.

As simple and apparently trouble-free as level gauges may seem, there are special circumstances where they will register incorrectly. One such circumstance is in the presence of a lighter liquid layer existing between the connection ports of the gauge. If a lighter (less dense) liquid exists above a heavier (denser) liquid in the process vessel, the level gauge may not show the proper interface, if at all:

Figure 12.3: Incorrect use of a sightglass in a tank with a liquid-liquid interface

This figure shows how a column of water in the sightglass shows less (total) level than the combination of water and oil inside the process vessel. Since the oil lies between the two level gauge ports into the vessel (sometimes called *nozzles*), it cannot enter the sightglass tube, and therefore the level gauge will continue to show just water.

If by chance some oil does find its way into the sightglass tube – either by the interface level dropping below the lower nozzle or the total level rising above the upper nozzle – the oil/water interface shown inside the level gauge may not continue to reflect the true interface inside the vessel once the interface and total levels return to their previous positions:

Figure 12.4: A second diagram showing the incorrect use of a sightglass

Recall that the level gauge and vessel together form a U-tube manometer. So long as the pressures from each liquid column are the same, the columns balance each other. The problem is, many different liquid-liquid interface columns can have the same hydrostatic pressure without being identical to one another:

Figure 12.5: A diagram showing various manometer conditions for liquid-liquid balance

The only way to ensure proper two-part liquid interface level indication in a sightglass is to keep both ports (nozzles) submerged:

Figure 12.6: Correct use of a sightglass in a tank with a liquid-liquid interface

Another troublesome scenario for level gauges is when the liquid inside the vessel is substantially hotter than the liquid in the gauge, causing the densities to be different. This is commonly seen on boiler level gauges, where the water inside the sightglass cools off substantially from its former temperature inside the boiler drum:

Figure 12.7: Incorrect use of a sightglass measuring boiler water

Looking at the sightglass as a U-tube manometer again, it can be seen that unequal-height liquid columns may indeed balance each other's hydrostatic pressures if the two columns are comprised of liquids with different densities. The weight density of water is 62.4 lb/ft$^3$ at standard temperature, but may be as low as only 36 lb/ft$^3$ at temperatures common for power generation boilers.

## 12.2 Float

Perhaps the simplest form of solid or liquid level measurement is with a *float*: a device that rides on the surface of the fluid or solid within the storage vessel. The float itself must be of substantially lesser density than the substance of interest, and it must not corrode or otherwise react with the substance.

Floats may be used for manual "gauging" of level, as illustrated here:

Figure 12.8: Diagram showing manual level measurement using a float and tape

A person lowers a float down into a storage vessel using a flexible measuring tape, until the tape goes slack due to the float coming to rest on the material surface. At that point, the person notes the length indicated on the tape (reading off the lip of the vessel access hole). This distance is called the *ullage*, being the distance from the top of the vessel to the surface of the process material. *Fillage* of the vessel may be determined by subtracting this "ullage" measurement from the known height of the vessel.

Obviously, this method of level measurement is tedious and may pose risk to the person conducting the measurement. If the vessel is pressurized, this method is simply not applicable.

If the person's function is automated using a small winch controlled by a computer – having the computer automatically lower the float down to the material surface and measure the amount of cable played out at each measurement cycle – better results are achieved without human intervention. Such a level gauge may be enclosed in such a way to allow pressurization of the vessel, too:

Figure 12.9: Diagram showing automatic level measurement using a float and tape

A simpler version of this technique uses a spring-reel to constantly tension the cable holding the float, such that the float continuously rides on the surface of the liquid in the vessel[1]:

Figure 12.10: Diagram showing automatic level measurement using a float and spring-loaded tape

[1] A spring-loaded cable float only works with liquid level measurement, while a retracting float will measure liquids and solids with equal ease. The reason for this limitation is simple: a float that always contacts the material surface is likely to become buried if the material in question is a solid (powder or granules), which must be fed into the vessel from above.

The following photograph shows the "measurement head" of a spring-reel tape-and-float liquid level transmitter, with the vertical pipe housing the tape on its way to the top of the storage tank where it will turn 180 degrees via two pulleys and attach to the float inside the tank:

Figure 12.11: A spring-loaded tape level measurement indicator

The spring reel's angular position may be measured by a multi-turn potentiometer or a rotary encoder (located inside the "head" unit), then converted to an electronic signal for transmission to a remote display, control, and/or recording system. Such systems are used extensively for measurement of water and fuel in storage tanks.

If the liquid inside the vessel is subject to turbulence, *guide wires* may be necessary to keep the float cable in a vertical orientation:

Figure 12.12: Diagram showing a float with guide wires

The guide wires are anchored to the floor and roof of the vessel, passing through ring lugs on the float to keep it from straying laterally.

One of the potential disadvantages of tape-and-float level measurement systems is fouling of the tape (and guide wires) if the substance is sticky or unclean.

A variation on the theme of float level measurement is to place a small float inside the tube of a sightglass-style level gauge:

Figure 12.13: Diagram showing a float used inside a sightglass

The float's position inside the tube may be readily detected by ultrasonic waves, magnetic sensors or any other applicable means. Locating the float inside a tube eliminates the need for guide wires

or a sophisticated tape retraction or tensioning system. If no visual indication is necessary, the level gauge tube may be constructed out of metal instead of glass, greatly reducing the risk of tube breakage. All the problems inherent to sightglasses, however, still apply to this form of float instrument.

## 12.3 Hydrostatic pressure

A vertical column of fluid exerts a pressure due to the column's weight. The relationship between column height and fluid pressure at the bottom of the column is constant for any particular fluid with a uniform density profile regardless of vessel width or shape.

This principle makes it possible to infer the height of liquid in a vessel by measuring the pressure generated at the bottom:

Figure 12.14: Pressure of a given water column height is independent of container shape

The mathematical relationship between liquid column height and pressure is as follows:

$$P = \rho g h \qquad\qquad P = \gamma h$$

Where,

$P$ = Hydrostatic pressure

$\rho$ = Mass density of fluid in kilograms per cubic meter (metric) or slugs per cubic foot (British)

$g$ = Acceleration of gravity

$\gamma$ = Weight density of fluid in newtons per cubic meter (metric) or pounds per cubic foot (British)

$h$ = Height of vertical fluid column above point of pressure measurement

For example, the pressure generated by a column of oil 12 feet high having a weight density ($\gamma$) of 40 pounds per cubic foot is:

$$P = \gamma h$$

$$P = \left(\frac{12 \text{ ft}}{1}\right)\left(\frac{40 \text{ lb}}{\text{ft}^3}\right)$$

$$P = \frac{480 \text{ lb}}{\text{ft}^2}$$

Note the cancellation of units, resulting in a pressure value of 480 pounds per square foot (PSF). To convert into the more common pressure unit of pounds per square inch, multiply by the proportion of square feet to square inches, eliminating the unit of square feet by cancellation and leaving square inches in the denominator:

$$P = \left(\frac{480 \text{ lb}}{\text{ft}^2}\right)\left(\frac{1^2 \text{ ft}^2}{12^2 \text{ in}^2}\right)$$

$$P = \left(\frac{480 \text{ lb}}{\text{ft}^2}\right)\left(\frac{1 \text{ ft}^2}{144 \text{ in}^2}\right)$$

$$P = \frac{3.\overline{33} \text{ lb}}{\text{in}^2} = 3.\overline{33} \text{ psig}$$

Thus, a pressure gauge attached to the bottom of the vessel holding a 12 foot column of this oil would register $3.\overline{33}$ psig. It is possible to customize the scale on the gauge to read directly in feet of oil (height) instead of psig, for convenience of the operator who must periodically read the gauge. Since the mathematical relationship between oil height and pressure is both linear and direct, the gauge's indication will always be proportional to height.

Any type of pressure-sensing instrument may be used as a liquid level transmitter by means of this principle. In the following photograph, Rosemount model 1151 pressure transmitter is shown being used to measure the height of colored water inside a clear plastic tube:

Figure 12.15: An industrial pressure transmitter used to infer level

The critically important factor in liquid level measurement using hydrostatic pressure is liquid density. One must accurately know the liquid's density in order to have any hope of measuring that liquid's level using hydrostatic pressure, since density is an integral part of the height/pressure relationship ($P = \rho gh$ and $P = \gamma h$). Having an accurate assessment of liquid density also implies that density must remain relatively constant despite other changes in the process. If the liquid density is subject to random variation, the accuracy of any hydrostatic pressure-based level instrument will correspondingly vary.

It should be noted, though, that changes in liquid density will have absolutely no effect on hydrostatic measurement of liquid *mass*, so long as the vessel has a constant cross-sectional area throughout its entire height. A simple thought experiment proves this: imagine a vessel partially full of liquid, with a pressure transmitter attached to the bottom to measure hydrostatic pressure. Now imagine the temperature of that liquid increasing, such that its volume expands and has a lower density than before. Assuming no addition or loss of liquid to or from the vessel, any increase in liquid level will be strictly due to volume expansion (density decrease). Liquid level inside this vessel will rise, but the transmitter will sense the exact same hydrostatic pressure as before, since the rise in level is precisely countered by the decrease in density (if $h$ increases by the same factor that $\gamma$ decreases, then $P = \gamma h$ must remain the same!). In other words, hydrostatic pressure is seen to be a direct indication of the liquid *mass* contained within the vessel, regardless of changes in liquid density.

Differential pressure transmitters are the most common pressure-sensing device used in this capacity to infer liquid level within a vessel. In the hypothetical case of the oil vessel just considered, the transmitter would connect to the vessel in this manner (with the high side toward the process and the low side vented to atmosphere):

Figure 12.16: Diagram showing level measurement using a differential pressure transmitter

Connected as such, the differential pressure transmitter functions as a gauge pressure transmitter, responding to hydrostatic pressure exceeding ambient (atmospheric) pressure. As liquid level increases, the hydrostatic pressure applied to the "high" side of the differential pressure transmitter also increases, driving the transmitter's output signal higher.

Some pressure-sensing instruments are built specifically for hydrostatic measurement of liquid level in vessels, doing away with impulse tubing altogether in favor of a special kind of sealing diaphragm extending slightly into the vessel through a flanged pipe entry (commonly called a *nozzle*). A Rosemount hydrostatic level transmitter with an extended diaphragm is shown here:

Figure 12.17: An industrial level transmitter with a diaphragm for tank mounting

The calibration table for a transmitter close-coupled to the bottom of an oil storage tank would

be as follows, assuming a zero to twelve foot measurement range for oil height, an oil density of 40 pounds per cubic foot, and a 4-20 mA transmitter output signal range:

| Oil level | Percent of range | Hydrostatic pressure | Transmitter output |
|---|---|---|---|
| 0 ft | 0 % | 0 psig | 4 mA |
| 3 ft | 25 % | 0.833 psig | 8 mA |
| 6 ft | 50 % | 1.67 psig | 12 mA |
| 9 ft | 75 % | 2.50 psig | 16 mA |
| 12 ft | 100 % | 3.33 psig | 20 mA |

### 12.3.1 Bubbler systems

An interesting variation on this theme of direct hydrostatic pressure measurement is the use of a purge gas to measure hydrostatic pressure in a liquid-containing vessel. This eliminates the need for direct contact of the process liquid against the pressure-sensing element, which can be advantageous if the process liquid is corrosive.

Such systems are often called *bubble tube* or *dip tube* systems, the former name being appropriately descriptive for the way purge gas bubbles out the end of the tube as it is submerged in process liquid. A key detail of a bubble tube system is to provide a means of limiting gas flow through the tube, so the purge gas backpressure properly reflects hydrostatic pressure at the end of the tube with no additional pressure due to frictional losses of purge flow through the length of the tube. Most bubble tube systems, therefore, are provided with some means of monitoring purge gas flow, typically with a *rotameter* or with a *sightfeed bubbler*:

Figure 12.18: Diagram showing level measurement using the bubbler method

If the purge gas flow is not too great, gas pressure measured anywhere in the tube system downstream of the needle valve will be equal to the hydrostatic pressure of the process liquid at the bottom of the tube where the gas escapes. In other words, the purge gas acts to transmit the liquid's hydrostatic pressure to some remote point where a pressure-sensing instrument is located. A general rule-of-thumb is to limit purge gas flow to the point where individual bubbles exiting the bubble tube may be easily counted (or inside the sightfeed bubbler if one is provided on the system).

As with all purged systems, certain criteria must be met for successful operation. Listed here are a few pertinent questions to consider for a bubble tube system:

- How reliable is the supply of purge fluid? If this stops for any reason, the level measurement may be in error!
- Is the purge fluid supply pressure guaranteed to exceed the hydrostatic pressure at all times, to ensure continuous purging (bubbling)?
- What options exist for purge gases that will not adversely react with the process?
- What options exist for purge gases that will not contaminate the process?
- How expensive will it be to maintain this constant flow of purge gas into the process?

One measurement artifact of a bubble tube system is a slight variation in pressure each time a new bubble breaks away from the end of the tube. The amount of pressure variation is approximately equal to the hydrostatic pressure of process fluid at a height equal to the diameter of the bubble, which in turn will be approximately equal to the diameter of the bubble tube. For example, a 1/4 inch diameter dip tube will experience pressure oscillations with a peak-to-peak amplitude of approximately 1/4 inch elevation of process liquid. The frequency of this pressure oscillation, of

course, will be equal to the rate at which individual bubbles escape out the end of the dip tube.

Usually, this is a small variation when considered in the context of the measured liquid height in the vessel. A pressure oscillation of approximately 1/4 inch compared to a measurement range of 0 to 10 feet, for example, is only about 0.2% of span. Modern pressure transmitters have the ability to "filter" or "damp" pressure variations over time, which is a useful feature for minimizing the effect such a pressure variation will have on system performance.

### 12.3.2 Transmitter suppression and elevation

A very common scenario for liquid level measurement is where the pressure-sensing instrument is not located at the same level as the 0% measurement point. The following photograph shows an example of this, where a Rosemount model 3051 differential pressure transmitter is being used to sense hydrostatic pressure of colored water inside a (clear) vertical plastic tube:

Figure 12.19: Hydrostatic level transmitter requiring an offset zero

Consider the example of a pressure sensor measuring the level of liquid ethanol in a storage tank. The measurement range for liquid height in this ethanol storage tank is 0 to 40 feet, but the transmitter is located 30 feet below the tank:

Figure 12.20: Diagram showing a hydrostatic level transmitter requiring an offset zero

This means the transmitter's impulse line contains a 30-foot elevation head of ethanol, so the transmitter "sees" 30 feet of ethanol when the tank is empty and 70 feet of ethanol when the tank is full. A 3-point calibration table for this instrument would look like this, assuming a 4 to 20 mA DC output signal range:

| Ethanol level in tank | Percent of range | Pressure (inches of water) | Pressure (psig) | Output (mA) |
|---|---|---|---|---|
| 0 ft | 0 % | 284 "w.c. | 10.3 psig | 4 mA |
| 20 ft | 50 % | 474 "w.c. | 17.1 psig | 12 mA |
| 40 ft | 100 % | 663 "w.c. | 24.0 psig | 20 mA |

Another common scenario is where the transmitter is mounted at or near the vessel's bottom, but the desired level measurement range does not extend to the vessel bottom:

Figure 12.21: Diagram showing a need for zero suppression

In this example, the transmitter is mounted exactly at the same level as the vessel bottom, but the level measurement range goes from 4 feet to 9 feet (a 5 foot span). At the level of castor oil deemed 0%, the transmitter "sees" a hydrostatic pressure of 1.68 psig (46.5 inches of water column) and at the 100% castor oil level the transmitter "sees" a pressure of 3.78 psig (105 inches water column). Thus, these two pressure values would define the transmitter's lower and upper range values (LRV and URV), respectively.

The term for describing either of the previous scenarios, where the lower range value (LRV) of the transmitter's calibration is a positive number, is called *zero suppression*[2]. If the zero offset is reversed (e.g. the transmitter mounted at a location *higher* than the 0% process level), it is referred to as *zero elevation*[3].

If the transmitter is elevated above the process connection point, it will most likely "see" a negative pressure (vacuum) with an empty vessel owing to the pull of liquid in the line leading down from the instrument to the vessel. It is vitally important in elevated transmitter installations to use a *remote seal* rather than an open impulse line, so liquid cannot dribble out of this line and into the vessel[4]:

[2]Or alternatively, zero *depression*.

[3]There is some disagreement among instrumentation professionals as to the definitions of these two terms. According to Béla G. Lipták's *Instrument Engineers' Handbook, Process Measurement and Analysis* (Fourth Edition, page 67), "suppressed zero range" refers to the transmitter being located below the 0% level (the LRV being a positive pressure value), while "suppression," "suppressed range," and "suppressed span" mean exactly the opposite (LRV is a negative value). The Yokogawa Corporation defines "suppression" as a condition where the LRV is a positive pressure ("Autolevel" Application Note), as does the Michael MacBeth in his CANDU Instrumentation & Control course (lesson 1, module 4, page 12), Foxboro's technical notes on bubble tube installations (pages 4 through 7), and Rosemount's product manual for their 1151 Alphaline pressure transmitter (page 3-7). Interestingly, the Rosemount document defines "zero range suppression" as synonymous with "suppression," which disagrees with Lipták's distinction. When faced with conflicting information like this, draw a picture to help the other person to clearly understand what is meant!

[4]The calibration of an elevated transmitter depends on knowing how much hydrostatic pressure (or vacuum, in this case) is generated within the tube connecting the transmitter to the process vessel. If liquid were to ever escape from this tube, the hydrostatic pressure would be unpredictable, and so would be the accuracy of the transmitter as a level-measuring instrument. A remote seal diaphragm guarantees no fill fluid will be lost if and when the process

Figure 12.22: Diagram showing the need for a remote seal to prevent fill fluid from entering tank

In this example, a remote seal system is shown with a fill fluid having a density of 58.3 lb/ft$^3$, and a process level measurement range of 0 to 11 feet of sea water (density = 64 lb/ft$^3$). The transmitter elevation is 6 feet, which means it will "see" a vacuum of -2.43 psig (-67.2 inches of water column) when the vessel is completely empty. This, of course, will be the transmitter's calibrated lower range value (LRV). The upper range value (URV) will be the pressure "seen" with 11 feet of sea water in the vessel. This much sea water will contribute an additional 4.89 psig of hydrostatic pressure at the level of the remote seal diaphragm, causing the transmitter to experience a pressure of +2.46 psig[5].

### 12.3.3 Compensated leg systems

The simple and direct relationship between liquid height in a vessel and pressure at the bottom of that vessel is ruined if another source of pressure exists inside the vessel other than hydrostatic (elevation head). This is virtually guaranteed to be the case if the vessel in question is unvented. Any gas or vapor pressure accumulation in an enclosed vessel will add to the hydrostatic pressure at the bottom, causing any pressure-sensing instrument to falsely register a high level:

vessel goes empty.

[5]The sea water's positive pressure at the remote seal diaphragm adds to the negative pressure already generated by the downward length of the capillary tube's fill fluid (-2.43 psig), which explains why the transmitter only "sees" 2.46 psig of pressure at the 100% full mark.

Figure 12.23: Diagram showing the need for pressure compensation in a closed tank

A pressure transmitter has no way of "knowing" how much of the sensed pressure is due to liquid elevation and how much of it is due to pressure existing in the vapor space above the liquid. Unless a way can be found to compensate for any non-hydrostatic pressure in the vessel, this extra pressure will be interpreted by the transmitter as additional liquid level.

Moreover, this error will change as gas pressure inside the vessel changes, so it cannot simply be "calibrated away" by a static zero shift within the instrument. The only way to hydrostatically measure liquid level inside an enclosed (non-vented) vessel is to continuously compensate for gas pressure.

Fortunately, the capabilities of a *differential* pressure transmitter make this a simple task. By connecting a second impulse line (called a *compensating leg*), from the "Low" port of the transmitter to the top of the vessel, the "Low" side of the transmitter experiences nothing but the gas pressure enclosed by the vessel, while the "High" side experiences the *sum* of gas and hydrostatic pressures. Since a differential pressure transmitter responds only to *differences* in pressure between "High" and "Low" sides, it will naturally subtract the gas pressure ($P_{gas}$) to yield a measurement based solely on hydrostatic pressure ($\gamma h$):

Figure 12.24: Pressure compensation used for hydrostatic level measurement in a closed tank

$$(P_{gas} + \gamma h) - P_{gas} = \gamma h$$

The amount of gas pressure inside the vessel now becomes completely irrelevant to the transmitter's indication, because its effect is canceled at the differential pressure instrument's sensing element. If gas pressure inside the vessel were to increase while liquid level remained constant, the pressure sensed at *both* ports of the differential pressure transmitter would increase by the exact same amount, with the pressure *difference* between the "high" and "low" ports remaining absolutely constant with the constant liquid level. This means the instrument's output signal is a representation of hydrostatic pressure only, which represents liquid height (assuming a known liquid density $\gamma$).

Unfortunately, it is common for enclosed vessels to hold condensible vapors, which may over time fill a compensating leg full of liquid. If the tube connecting the "Low" side of a differential pressure transmitter fills completely with a liquid, this will add a hydrostatic pressure to that side of the transmitter, causing another calibration shift. This *wet leg* condition makes level measurement more complicated than a *dry leg* condition where the only pressure sensed by the transmitter's "Low" side is gas pressure ($P_{gas}$):

Figure 12.25: Hydrostatic level measurement in a closed tank is affected by wet leg column

$$(P_{gas} + \gamma_1 h_1) - (P_{gas} + \gamma_2 h_2) = \gamma_1 h_1 - \gamma_2 h_2$$

Gas pressure still cancels due to the differential nature of the pressure transmitter, but now the transmitter's output indicates a difference of hydrostatic pressures between the vessel and the wet leg, rather than just the hydrostatic pressure of the vessel's liquid level. Fortunately, the hydrostatic pressure generated by the wet leg will be constant, so long as the density of the condensed vapors filling that leg ($\gamma_2$) is constant. If the wet leg's hydrostatic pressure is constant, it can be removed from the equation by calibrating the transmitter with an intentional zero shift, so it indicates as though it were measuring hydrostatic pressure on a vented vessel.

$$\text{Differential pressure } = \gamma_1 h_1 - \text{Constant}$$

A constant density of wet leg liquid may be ensured by intentionally filling that leg with a liquid known to be denser than the densest condensed vapor inside the vessel. A differential pressure transmitter with remote seals and capillary tubes filled with liquid of known density could also be used:

Figure 12.26: Wet leg implementation methods

The following example shows the calibration table for a compensated-leg (wet) hydrostatic level measurement system, for a gasoline storage vessel and water as the wet leg fill fluid. Here, a density of 41.0 lb/ft$^3$ for gasoline and 62.4 lb/ft$^3$ for water is assumed, with a 0 to 10 foot measurement range and an 11 foot wet leg height:

| Gasoline level | Percent of range | Pressure at transmitter | Transmitter output |
|---|---|---|---|
| 0 ft | 0 % | -4.77 psig | 4 mA |
| 2.5 ft | 25 % | -4.05 psig | 8 mA |
| 5 ft | 50 % | -3.34 psig | 12 mA |
| 7.5 ft | 75 % | -2.63 psig | 16 mA |
| 10 ft | 100 % | -1.92 psig | 20 mA |

Note that due to the superior density and height of the wet (water) leg, the transmitter *always* sees a negative pressure (pressure on the "Low" side exceeds pressure on the "High" side). With some older differential pressure transmitter designs, this was a problem. Consequently, it is common to see "wet leg" hydrostatic transmitters installed with the "Low" port connected to the bottom of the vessel and the "High" port connected to the compensating leg. In fact, it is *still* common to see modern differential pressure transmitters installed in this manner[6], although modern transmitters may be calibrated for negative pressures just as easily as for positive pressures. It is vitally important to recognize that any differential pressure transmitter connected as such (for any reason) will respond in reverse fashion to increases in liquid level. That is to say, as the liquid level in the vessel rises, the transmitter's output signal will *decrease* instead of increase:

[6]Sometimes this is done out of habit, other times because I&C engineers do not know the capabilities of new technology.

Figure 12.27: A reverse-acting hydrostatic level transmitter on a closed tank

Either way of connecting the transmitter to the vessel will suffice for measuring liquid level, so long as the instrumentation receiving the transmitter's signal is properly configured to interpret the signal. The choice of which way to connect the transmitter to the vessel should be driven by fail-safe system design, which means to design the measurement system such that the most probably system failures – including broken signal wires – result in the control system "seeing" the most dangerous process condition and therefore taking the safest action.

### 12.3.4 Tank expert systems

An alternative to using a compensating leg to subtract gas pressure inside an enclosed vessel is to simply use a second pressure transmitter and electronically subtract the two pressures in a computing device:

This approach enjoys the distinct advantage of avoiding a potentially wet compensating leg, but suffers the disadvantages of extra cost and greater error due to the potential calibration drift of *two* transmitters rather than just one. Such a system is also impractical in applications where the gas pressure is substantial compared to the hydrostatic (elevation head) pressure[7]. This measurement arrangement is often used to measure level in tanks that are tall, limiting the length of impulse tubing required.

If a third pressure transmitter is added to this system and located a known distance ($x$) above the bottom transmitter, all the pieces are in place to create what is called a *tank expert system*. These systems are used on large storage tanks operating at or near atmospheric pressure, and have the ability to measure infer liquid height, liquid density, total liquid volume, and total liquid mass stored in the tank:

[7]This is due to limited transmitter resolution. Imagine an application where the elevation head was 10 psig (maximum) yet the vapor space pressure was 200 psig. The majority of each transmitter's working range would be "consumed" measuring gas pressure, with hydrostatic head being a mere 5% of the measurement range. This would make precise measurement of liquid level very difficult, akin to trying to measure the sound intensity of a whisper in a noisy room.

Figure 12.28: Diagram of a tank expert system

The pressure difference between the bottom and middle transmitters will change only if the liquid density changes[8], since the two transmitters are separated by a known and fixed height difference.

Algebraic manipulation shows how the measured pressures may be used by the level computer (LY) to continuously calculate liquid density ($\gamma$):

$$P_{bottom} - P_{middle} = (P_{gas} + \gamma h) - [P_{gas} + \gamma(h - x)]$$

$$P_{bottom} - P_{middle} = P_{gas} + \gamma h - P_{gas} - \gamma(h - x)$$

$$P_{bottom} - P_{middle} = P_{gas} + \gamma h - P_{gas} - \gamma h + \gamma x$$

$$P_{bottom} - P_{middle} = \gamma x$$

[8] Assuming the liquid level is equal to or greater than $x$. Otherwise, the pressure difference between $P_{bottom}$ and $P_{middle}$ will depend on liquid density *and* liquid height. However, this condition is easy to check: the level computer simply checks to see if $P_{middle}$ and $P_{top}$ are unequal. If so, then the computer knows the liquid level exceeds $x$ and it is safe to calculate density. If not, and $P_{middle}$ registers the same as $P_{top}$, the computer knows those two transmitters are both registering gas pressure only, and it knows to stop calculating density.

$$\frac{P_{bottom} - P_{middle}}{x} = \gamma$$

Once the computer knows the value of $\gamma$, it may calculate the height of liquid in the tank with great accuracy based on the pressure measurements taken by the bottom and top transmitters:

$$P_{bottom} - P_{top} = (P_{gas} + \gamma h) - P_{gas}$$

$$P_{bottom} - P_{top} = \gamma h$$

$$\frac{P_{bottom} - P_{top}}{\gamma} = h$$

With all the computing power available in the LY, it is possible to characterize the tank such that this height measurement converts to a precise volume measurement[9] ($V$), which may then be converted into a total mass ($m$) measurement based on the mass density of the liquid ($\rho$) and the acceleration of gravity ($g$). First, the computer calculates mass density based on the proportionality between mass and weight (shown here starting with the equivalence between the two forms of the hydrostatic pressure formula):

$$\rho g h = \gamma h$$

$$\rho g = \gamma$$

$$\rho = \frac{\gamma}{g}$$

Armed with the mass density of the liquid inside the tank, the computer may now calculate total liquid mass stored inside the tank:

$$m = \rho V$$

Dimensional analysis shows how units of mass density and volume cancel to yield only units of mass in this last equation:

$$[\mathrm{kg}] = \left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right] [\mathrm{m}^3]$$

This is a vivid example of how several measurements may be inferred from just a few actual process (in this case, pressure) measurements. Three pressure measurements on this tank allow the computation of four inferred variables: liquid density, liquid height, liquid volume, and liquid mass.

[9] The details of this math depend entirely on the shape of the tank. For vertical cylinders – the most common shape for vented storage tanks – volume and height are related by the simple formula $V = \pi r^2 h$ where $r$ is the radius of the tank's circular base. Other tank shapes and orientations may require much more sophisticated formulae to calculate stored volume from height. See section 15.2, beginning on page 624, for more details on this subject.

The accurate measurement of liquids in storage tanks is not just useful for process operations, but also for conducting business affairs. Whether the liquid represents raw material purchased from a supplier, or a processed product ready to be pumped out to a customer, both parties have a vested interest in knowing the exact quantity of liquid bought or sold. Measurement applications such as this are known as *custody transfer*, because they represent the transfer of custody (ownership) of a substance exchanged in a business agreement. In some instances, both buyer and seller operate and maintain their own custody transfer instrumentation, while in other instances there is but one instrument, the calibration of which validated by a neutral party.

### 12.3.5 Hydrostatic interface level measurement

Hydrostatic pressure sensors may be used to detect the level of a liquid-liquid interface, if and only if the total height of liquid sensed by the instrument is fixed. A single hydrostatic-based level instrument cannot discern between a changing interface level and a changing total level, so the latter must be fixed in order to measure the former.

One way of fixing total liquid height is to equip the vessel with an overflow pipe, and ensure that drain flow is always less than incoming flow (forcing some flow to always go through the overflow pipe). This strategy naturally lends itself to separation processes, where a mixture of light and heavy liquids are separated by their differing densities:

Figure 12.29: Tank level measurement when a liquid-liquid interface exists

Here is a practical application for liquid-liquid interface level measurement. If the goal is to separate two liquids of differing densities from one another, only the light liquid needs to exit out

the overflow pipe and only the heavy liquid needs to exit out the drain pipe. This means the interface level must be controlled to stay between those two piping points on the vessel. If the interface drifts too far up, heavy liquid will be carried out the overflow pipe; and if the interface drifts too far down, light liquid will flow out of the drain pipe. The first step in controlling any process variable is to measure that variable, and so this problem requires measuring the interface point between the light and heavy liquids.

Another way of fixing the total height seen by the transmitter is to use a compensating leg located at a point on the vessel always lower than the total liquid height. In this example, a transmitter with remote seals is used:

Figure 12.30: Another type of tank level measurement when a liquid-liquid interface exists

Since both sides of the differential pressure transmitter "see" the hydrostatic pressure generated by the liquid column above the top connection point ($\gamma_2 h_3$), this term is naturally canceled:

$$(\gamma_1 h_1 + \gamma_2 h_2 + \gamma_2 h_3) - (\gamma_4 h_4 + \gamma_2 h_3)$$

$$\gamma_1 h_1 + \gamma_2 h_2 + \gamma_2 h_3 - \gamma_4 h_4 - \gamma_2 h_3$$

$$\gamma_1 h_1 + \gamma_2 h_2 - \gamma_4 h_4$$

The hydrostatic pressure in the compensating leg is constant ($\gamma_4 h_4 = \text{Constant}$), since the fill fluid never changes density and the height never changes. This means the transmitter's sensed pressure will differ from that of an uncompensated transmitter merely by a constant offset, which may be "calibrated out" so as to have no impact on the measurement:

$$\gamma_1 h_1 + \gamma_2 h_2 - \text{Constant}$$

At first, it may seem as though determining the calibration points (lower- and upper-range values: LRV and URV) for a hydrostatic interface level transmitter is impossibly daunting given all the different pressures involved. A recommended problem-solving technique to apply here is that of a *thought experiment*, imagining what the process will "look like" at lower-range value condition and at the upper-range value condition, drawing two separate illustrations:

Figure 12.31: Graphic of a thought experiment used to determine level transmitter range values

For example, suppose a differential pressure transmitter must be calibrated to measure the interface level between two liquids having specific gravities of 1.1 and 0.78, respectively, over a span of 3 feet. The transmitter is equipped with remote seals, each containing a halocarbon fill fluid with a specific gravity of 1.09. The physical layout of the system is as follows:

Figure 12.32: Layout of level measurement system

As the first step in the "thought experiment," imagine what the process would look like with the

interface at the LRV level, calculating hydrostatic pressures seen at each side of the transmitter:

Figure 12.33: Determining the lower range value for level transmitter calibration

From previous exploration of this setup, any hydrostatic pressure resulting from liquid level *above* the top remote seal location is irrelevant to the transmitter, since it is "seen" on both sides of the transmitter and thus cancels out. All that must be done, then, is to calculate hydrostatic pressures as though the total liquid level stopped at that upper diaphragm connection point.

First, calculating the hydrostatic pressure "seen" at the high port of the transmitter[10]:

$$P_{high} = 4.5 \text{ feet of heavy liquid } + 4.5 \text{ feet of light liquid}$$

$$P_{high} = 54 \text{ inches of heavy liquid } + 54 \text{ inches of light liquid}$$

$$P_{high} \text{ "W.C. } = (54 \text{ inches of heavy liquid })(1.1) + (54 \text{ inches of light liquid })(0.78)$$

$$P_{high} \text{ "W.C. } = 59.4 \text{ "W.C. } + 42.12 \text{ "W.C.}$$

$$P_{high} = 101.52 \text{ "W.C.}$$

Next, calculating the hydrostatic pressure "seen" at the low port of the transmitter:

$$P_{low} = 9 \text{ feet of fill fluid}$$

[10]Here all hydrostatic pressures are calculated in units of inches water column. This is relatively easy because the specific gravities of each liquid are known, which makesS it easy to translate actual liquid column height into column heights of pure water.

$$P_{low} = 108 \text{ inches of fill fluid}$$

$$P_{low} \text{ "W.C. } = (108 \text{ inches of fill fluid })(1.09)$$

$$P_{low} = 117.72 \text{ "W.C.}$$

The differential pressure applied to the transmitter in this condition is the difference between the high and low port pressures, which becomes the lower range value (LRV) for calibration:

$$P_{LRV} = 101.52 \text{ "W.C. } - 117.72 \text{ "W.C. } = -16.2 \text{ "W.C.}$$

As the second step in the "thought experiment," imagine what the process would look like with the interface at the URV level, calculating hydrostatic pressures seen at each side of the transmitter:

Figure 12.34: Determining the upper range value for level transmitter calibration

$$P_{high} = 7.5 \text{ feet of heavy liquid } + 1.5 \text{ feet of light liquid}$$

$$P_{high} = 90 \text{ inches of heavy liquid } + 18 \text{ inches of light liquid}$$

$$P_{high} \text{ "W.C. } = (90 \text{ inches of heavy liquid })(1.1) + (18 \text{ inches of light liquid })(0.78)$$

$$P_{high} \text{ "W.C. } = 99 \text{ "W.C. } + 14.04 \text{ "W.C.}$$

$$P_{high} = 113.04 \text{ "W.C.}$$

The hydrostatic pressure of the compensating leg is exactly the same as it was before: 9 feet of fill fluid having a specific gravity of 1.09, which means there is no need to calculate it again. It will still be 117.72 inches of water column. Thus, the differential pressure at the URV point is:

$$P_{URV} = 113.04 \text{ "W.C.} - 117.72 \text{ "W.C.} = -4.68 \text{ "W.C.}$$

Using these two pressure values and some interpolation, generate a 5-point calibration table (assuming a 4-20 mA transmitter output signal range) for this interface level measurement system:

| Interface level | Percent of range | Pressure at transmitter | Transmitter output |
|---|---|---|---|
| 4.5 ft | 0 % | -16.2 "W.C. | 4 mA |
| 5.25 ft | 25 % | -13.32 "W.C. | 8 mA |
| 6 ft | 50 % | -10.44 "W.C. | 12 mA |
| 6.75 ft | 75 % | -7.56 "W.C. | 16 mA |
| 7.5 ft | 100 % | -4.68 "W.C. | 20 mA |

When the time comes to bench-calibrate this instrument in the shop, the easiest way to do so will be to set the two remote diaphragms on the workbench (at the same level), then apply 16.2 to 4.68 inches of water column pressure to the *low* remote seal diaphragm with the other diaphragm at atmospheric pressure to simulate the desired range of negative differential pressures[11].

The more mathematically inclined reader will notice that the span of this instrument (URV − LRV) is equal to the span of the interface level (3 feet, or 36 inches) multiplied by the difference in specific gravities (1.1 − 0.78):

$$\text{Span in "W.C.} = (36 \text{ inches})(1.1 - 0.78)$$

$$\text{Span} = 11.52 \text{ "W.C.}$$

Looking at the two "thought experiment" illustrations, it can be seen that the only difference between the two scenarios is the type of liquid filling that 3-foot region between the LRV and URV marks. Therefore, the only difference between the transmitter's pressures in those two conditions will be the difference in height multiplied by the difference in density. Not only is this an easy way to quickly calculate the necessary transmitter span, but it also is a way to check the previous work: the difference between the LRV and URV pressures is indeed a difference of 11.52 inches water column just as this method predicts.

## 12.4 Displacement

*Displacer* level instruments exploit *Archimedes' Principle* to detect liquid level by continuously measuring the weight of a rod immersed in the process liquid. As liquid level increases, the displacer

[11] Remember that a differential pressure instrument cannot "tell the difference" between a positive pressure applied to the low side, an equal vacuum applied to the high side, or an equivalent difference of two positive pressures with the low side's pressure exceeding the high side's pressure. Simulating the exact process pressures experienced in the field to a transmitter on a workbench would be exceedingly complicated, so an easier method is to "cheat" by simplifying the calibration setup and applying the equivalent difference of pressure only to the "low" side.

rod experiences a greater buoyant force, making it appear lighter to the sensing instrument, which interprets the loss of weight as an increase in level and transmits a proportional output signal.

In practice a displacer level instrument usually takes the following form:

Figure 12.35: Diagram of a displacer level measurement system

The following photograph shows a Fisher "LevelTrol" model pneumatic transmitter measuring condensate level in a *knockout drum*[12] for natural gas service. The instrument itself appears on the right-hand side of the photo, topped by a grey-colored "head" with two pneumatic pressure gauges visible. The displacer "cage" is the vertical pipe immediately behind and below the head unit. Note that a sightglass level gauge appears on the left-hand side of the knockout chamber (or *condensate boot*) for visual indication of condensate level inside the process vessel:

[12]So-called for its ability to "knock out" (separate and collect) condensible vapors from the gas stream. This particular photograph was taken at a natural gas compression facility, where it is very important the gas to be compressed is dry (since liquids are essentially incompressible). Sending even relatively small amounts of liquid into a compressor may cause the compressor to catastrophically fail!

Figure 12.36: An installed displacer level measurement system

Two photos of a disassembled LevelTrol displacer instrument appear here, showing how the displacer fits inside the cage pipe:

The cage pipe is coupled to the process vessel through two block valves, allowing isolation from the process. A drain valve allows the cage to be emptied of process liquid for instrument service and zero calibration.

Full-range calibration may be performed by flooding the cage with process liquid (a *wet* calibration), or by suspending the displacer with a string and precise scale (a *dry* calibration), pulling upward on the displacer at just the right amount to simulate buoyancy at 100% liquid level:

Figure 12.37: Diagram showing calibration of a displacer level measurement system

Calculation of this buoyant force is a simple matter. According to Archimedes' Principle, buoyant force is always equal to the weight of the fluid volume displaced. In the case of a displacer-based level instrument at full range, this usually means the entire volume of the displacer element is submerged in the liquid. Simply calculate the volume of the displacer (if it is a cylinder, $V = \pi r^2 l$, where $r$ is the cylinder radius and $l$ is the cylinder length) and multiply that volume by the weight density ($\gamma$):

$$F_{buoyant} = \gamma V$$

$$F_{buoyant} = \gamma \pi r^2 l$$

For example, if the weight density of the process fluid is 57.3 pounds per cubic foot and the displacer is a cylinder measuring 3 inches in diameter and 24 inches in length, the necessary force to simulate a condition of buoyancy at full level may be calculated as follows:

$$\gamma = \left(\frac{57.3 \text{ lb}}{\text{ft}^3}\right)\left(\frac{1 \text{ ft}^3}{12^3 \text{ in}^3}\right) = 0.0332\frac{\text{lb}}{\text{in}^3}$$

$$V = \pi r^2 l = \pi(1.5 \text{ in})^2(24 \text{ in}) = 169.6 \text{ in}^3$$

$$F_{buoyant} = \gamma V = \left(\frac{0.0332\text{lb}}{\text{in}^3}\right)\left(169.6 \text{ in}^3\right) = 5.63 \text{ lb}$$

Note how important it is to maintain consistency of units! The liquid density was given in units of pounds per cubic *foot* and the displacer dimensions in *inches*, which would have caused serious problems without a conversion between feet and inches. In this example work, density was convertinto units of pounds per cubic inch, butthe displacer dimensions could have just as easily been converted into feet to arrive at a displacer volume in units of cubic feet.

### 12.4.1 Displacement interface level measurement

Displacer level instruments may be used to measure liquid-liquid interfaces just the same as hydrostatic pressure instruments. One important requirement is that the displacer always be fully submerged. If this rule is violated, the instrument will not be able to "tell" the difference between a low (total) liquid level and a low interface level.

If the displacer instrument has its own "cage," it is important that both pipes connecting the cage to the process vessel (sometimes called "nozzles") be submerged. This ensures the liquid interface inside the cage matches the interface inside the vessel. If the upper nozzle ever goes dry, the same problem can happen with a caged displacer instrument as with a "sightglass" level gauge (see page 404 for a detailed explanation of this problem.).

Calculating buoyant force on a displacer element due to a combination of two liquids is not as difficult as it may sound. Archimedes' Principle still holds: that buoyant force is equal to the weight of the fluid(s) displaced. Simply calculate the combined weights and volumes of the displaced liquids to calculate buoyant force. For a single liquid, buoyant force is equal to the weight density of that liquid ($\gamma$) multiplied by the volume displaced ($V$):

$$F_{buoyant} = \gamma V$$

For a two-liquid interface, the buoyant force is equal to the sum of the two liquid weights displaced, each liquid weight term being equal to the weight density of that liquid multiplied by the displaced volume of that liquid:

$$F_{buoyant} = \gamma_1 V_1 + \gamma_2 V_2$$

Assuming a displacer of constant cross-sectional area throughout its length, the volume for each liquid's displacement is simply equal to the same area ($\pi r^2$) multiplied by the length of the displacer submerged in that liquid:

Figure 12.38: Diagram of a displacer level measurement system detecting a liquid-liquid interface

$$F_{buoyant} = \gamma_1 \pi r^2 l_1 + \gamma_2 \pi r^2 l_2$$

Since the area ($\pi r^2$) is common to both buoyancy terms in this equation, it may be factored out for simplicity:

$$F_{buoyant} = \pi r^2 (\gamma_1 l_1 + \gamma_2 l_2)$$

Calculating the LRV buoyant force is as simple as setting $l_1$ equal to zero and $l_2$ equal to the total displacer length ($L$):

$$F_{buoyant} \text{ (LRV) } = \pi r^2 \gamma_2 L$$

Calculating the URV buoyant force is as simple as setting $l_2$ equal to zero and $l_1$ equal to the total displacer length ($L$):

$$F_{buoyant} \text{ (URV) } = \pi r^2 \gamma_1 L$$

The buoyancy for any measurement percentage between the LRV (0%) and URV (100%) may be calculated by interpolation. Sample calculations are shown below for a displacer instrument measuring the interface level between two liquids having specific gravities of 0.850 and 1.10, with a displacer length of 30 inches and a displacer diameter of 2.75 inches (radius = 1.375 inches):

$$\gamma_1 = \left(62.4 \frac{\text{lb}}{\text{ft}^3}\right)(1.10) = 68.6 \frac{\text{lb}}{\text{ft}^3} = 0.0397 \frac{\text{lb}}{\text{in}^3}$$

$$\gamma_2 = \left(62.4 \frac{\text{lb}}{\text{ft}^3}\right)(0.85) = 53.0 \frac{\text{lb}}{\text{ft}^3} = 0.0307 \frac{\text{lb}}{\text{in}^3}$$

$$F_{buoyant}\ (\text{LRV}) = \pi(1.375\ \text{in})^2 \left(0.0307\ \frac{\text{lb}}{\text{in}^3}\right)(30\ \text{in}) = 5.47\ \text{lb}$$

$$F_{buoyant}\ (\text{URV}) = \pi(1.375\ \text{in})^2 \left(0.0397\ \frac{\text{lb}}{\text{in}^3}\right)(30\ \text{in}) = 7.08\ \text{lb}$$

| Interface level (inches) | Buoyant force (pounds) |
|---|---|
| 0 | 5.47 |
| 7.5 | 5.87 |
| 15 | 6.27 |
| 22.5 | 6.68 |
| 30 | 7.08 |

## 12.5 Echo

A completely different way of measuring liquid level in vessels is to bounce a traveling wave off the surface of the liquid – typically from a location at the top of the vessel – using the time-of-flight for the waves as an indicator of distance[13], and therefore an indicator of liquid height inside the vessel. Echo-based level instruments enjoy the distinct advantage of immunity to changes in liquid density, a factor crucial to the accurate calibration of hydrostatic and displacement level instruments. In this regard, they are quite comparable with float-based level measurement systems.

The single most important factor to the accuracy of an echo-based level instrument is the speed at which the wave travels en route to the liquid surface and back. This wave propagation speed is as fundamental to the accuracy of an echo instrument as liquid density is to the accuracy of a hydrostatic or displacer instrument. So long as this velocity is known and stable, good level measurement accuracy is possible.

From a historical perspective, hydrostatic and displacement level instruments have a richer pedigree. These instruments are simpler in nature than echo-based instruments, and were practical long before the advent of modern electronic technology. Echo-based instruments require precision timing and wave-shaping circuitry, plus sensitive (and rugged!) transceiver elements, demanding a much higher level of technology. However, modern electronic design and instrument manufacturing practices are making echo-based level instruments more and more practical for industrial applications. At the time of this writing (2008), it is common practice in some industries to replace old displacer level instruments with guided-wave radar instruments, even in demanding applications operating at high pressures[14].

Liquid-liquid interfaces may also be measured with some types of echo-based level instruments, most commonly guided-wave radar.

Echo-based level instruments may be "fooled" by layers of foam resting on top of the liquid, and the liquid-to-liquid interface detection models may have difficulty detecting non-distinct interfaces (such as emulsions). Irregular structures residing within the vapor space of a vessel (such as access

[13] The general term for this form of measurement is *time domain reflectometry*.

[14] My own experience with this trend is within the oil refining industry, where legacy displacer instruments (typically Fisher brand "LevelTrol" units) are being replaced with new guided-wave radar transmitters, both for single-liquid and liquid-liquid interface applications.

portals, mixer paddles and shafts, ladders, etc.) may wreak havoc with echo-based level instruments by casting false echoes back to the instrument, although this problem may be mitigated by installing guide tubes for the waves to travel in, or using wave probes as in the cases of guided-wave radar instruments. Liquid streams pouring in to the vessel through the vapor space may similarly cause problems for an echo instrument. Additionally, all echo-based instruments have *dead zones* where liquid level is too close to the transceiver to be accurately measured or even detected (the echo time-of-flight being too short for the receiving electronics to distinguish from the incident pulse).

### 12.5.1 Ultrasonic level measurement

*Ultrasonic* level instruments measure the distance from the transmitter (located at some high point) to the surface of a process material located further below. The time-of-flight for a sound pulse indicates this distance, and is interpreted by the transmitter electronics as process level. These transmitters may output a signal corresponding either to the fullness of the vessel (*fillage*) or the amount of empty space remaining at the top of a vessel (*ullage*).

Figure 12.39: Diagram of an ultrasonic level measurement system

Ullage is the "natural" mode of measurement for this sort of level instrument, because the sound wave's time-of-flight is a direct function of how much empty space exists between the liquid surface and the top of the vessel. Total tank height will always be the sum of fillage and ullage, though. If the ultrasonic level transmitter is programmed with the vessel's total height, it may calculate fillage via simple subtraction:

$$\text{Fillage} = \text{Total height} - \text{Ullage}$$

The instrument itself consists of an electronics module containing all the power, computation, and signal processing circuits; plus an ultrasonic transducer[15] to send and receive the sound waves.

[15] In the industrial instrumentation world, the word "transducer" usually has a very specific meaning: a device used to process or convert standardized instrumentation signals, such as 4-20 mA converted into 3-15 psig, etc. In

This transducer is typically piezoelectric in nature, being the equivalent of a very high-frequency audio speaker. A typical example is shown in the following photograph:

Figure 12.40: An industrial ultrasonic level transducer

If the ultrasonic transducer is rugged enough, and the process vessel sufficiently free of sludge and other sound-damping materials accumulating at the vessel bottom, the transducer may be mounted at the bottom of the vessel, bouncing sound waves off the liquid surface through the liquid itself rather than through the vapor space:

the general scientific world, however, the word "transducer" describes any device converting one form of energy into another. It is this latter definition of the word used to describe an ultrasonic "transducer" – a device used to convert electrical energy into ultrasonic sound waves, and visa-versa.

Figure 12.41: Diagram of an ultrasonic level measurement system with bottom placement

This arrangement makes fillage the natural measurement, and ullage a derived measurement (calculated by subtraction from total vessel height).

$$\text{Ullage} = \text{Total height} - \text{Fillage}$$

Whether the ultrasonic transducer is mounted above or below the liquid level, the principle of detection is any significant difference in material *density*. If the detection interface is between a gas and a liquid, the abrupt change in density is enough to create a strong reflected signal. However, it is possible for foam and floating solids to also cause echos when the transducer is above-mounted, which may or may not be desirable depending on the application[16].

Ultrasonic level instruments enjoy the advantage of being able to measure the height of solid materials such as powders and grains stored in vessels, not just liquids. Certain challenges unique to these level measurement applications include low material density (not causing strong reflections) and uneven profiles (causing reflections to be scattered laterally instead of straight back to the ultrasonic instrument. A classic problem encountered when measuring the level of a powdered or granular material in a vessel is the *angle of repose* formed by the material as a result of being fed into the vessel at one point:

[16] If the goal is to only detect the liquid, then reflections from foam or solids would be bad. However, if the goal of measuring level is to prevent a vessel from overflowing, it is good to measure *anything* floating on the liquid surface!

Figure 12.42: Difficulty using ultrasonic level measurement system to measure solids

This angled surface is difficult for an ultrasonic device to detect because it tends to scatter the sound waves laterally instead of reflecting them strongly back toward the instrument. However, even if the scattering problem is not significant, there still remains the problem of interpretation: what is the instrument actually measuring? The detected level near the vessel wall will certainly register less than at the center, but the level detected mid-way between the vessel wall and vessel center may not be an accurate average of those two heights. Moreover, this angle may decrease over time if mechanical vibrations cause the material to "flow" and tumble from center to edge.

In fact, the angle will probably reverse itself if the vessel empties from a center-located chute:

Figure 12.43: Additional difficulty using ultrasonic level measurement system to measure solids

For this reason, solids storage measurement applications demanding high accuracy generally use other techniques, such as weight-based measurement (see section 12.7 for more information).

Since the speed of sound is so important to accurate distance calculations for ultrasonic instruments, some ultrasonic level instruments are equipped with temperature sensors to measure the temperature of the fluid through which the sound waves travel. A formula programmed into the transmitter calculates the speed of sound based on temperature, so the instrument may continuously compensate for changes in sound velocity rooted in temperature changes, and therefore maintain superior accuracy over a wide range of ambient conditions. In the vast majority of ultrasonic level transmitter installations (where the instrument is mounted above the liquid level such that the sound waves travel through air, bounce off liquid, and travel back through air), it is the speed of sound through air that matters. The speed of sound through liquid is irrelevant in these applications, since most of the acoustic energy reflects off the liquid surface and therefore never travels through it.

### 12.5.2 Radar level measurement

*Radar*[17] level instruments measure the distance from the transmitter (located at some high point) to the surface of a process material located further below in much the same way as ultrasonic transmitters – by measuring the time-of-flight of a traveling wave. The fundamental difference between a radar instrument and an ultrasonic instrument is the type of wave used: radio waves instead of sound waves. Radio waves are electromagnetic in nature (comprised of alternating electric and magnetic fields), and very high frequency (in the microwave frequency range – GHz). Sound waves are *mechanical* vibrations (transmitted from molecule to molecule in a fluid or solid substance) and of much lower frequency (tens or hundreds of kilohertz – still too high for a human being to detect as a tone) than radio waves.

Some radar level instruments use waveguide "probes" to guide the radio waves into the process liquid while others send radio waves out through open space to reflect off the process material. The instruments using waveguides are called *guided-wave radar* instruments, whereas the radar instruments relying on open space for signal propagation are called *non-contact radar*. The differences between these two varieties of radar instruments is shown in the following illustration:

[17] "Radar" is an acronym: RAdio Detection And Ranging. First used as a method for detecting enemy ships and aircraft at long distances over the ocean in World War II, this technology is used for detecting the presence, distance, and/or speed of objects in a wide variety of applications.

Figure 12.44: Types of radar level measurement

Non-contact radar transmitters are always mounted on the top side of a storage vessel. Modern radar transmitters are quite compact, as this photograph shows:

Figure 12.45: An industrial radar level transmitter

Probes used in guided-wave radar instruments may be single metal rods, parallel pairs of metal rods, or a coaxial metal rod-and-tube structure. Single-rod probes radiate the most energy, whereas coaxial probes do the best job guiding the microwave energy to the liquid interface and back. However, single-rod probes are much more tolerant of process fouling than two-rod or (especially) coaxial probes, where sticky masses of viscous liquid and/or solid matter cling to the probe. Such fouling deposits, if severe enough, will cause radio energy reflections that "look" to the transmitter like the reflection from an actual liquid level or interface.

Non-contact radar instruments rely on an antenna to direct microwave energy into the vessel, and to receive the echo (return) energy. These antennae must be kept clean and dry, which may be a problem if the liquid being measured emits condensible vapors. For this reason, non-contact radar instruments are often separated from the vessel interior by means of a *dielectric window* (made of some substance that is relatively "transparent" to radio waves yet acts as an effective vapor barrier):

Figure 12.46: A radar level transmitter with a dielectric window

Radio waves travel at the velocity of light[18], 2.9979 $\times 10^8$ meters per second in a perfect vacuum. The velocity of a radio wave through space depends on the dielectric permittivity (symbolized by the Greek letter "epsilon," $\epsilon$) of that space. A formula relating wave velocity to relative permittivity (the ratio of a substance's permittivity to that of a perfect vacuum, symbolized as $\epsilon_r$ and sometimes called the *dielectric constant* of the substance) and the velocity of light in a perfect vacuum ($c$) is shown here[19]:

$$v = \frac{c}{\sqrt{\epsilon_r}}$$

The relative permittivity of air at standard pressure and temperature is very nearly unity (1). The permittivity of any gas is a function of both pressure and temperature, as shown by the following formula:

$$\epsilon_r = 1 + (\epsilon_{ref} - 1)\frac{PT_{ref}}{P_{ref}T}$$

Where,

[18]In actuality, both radio waves and light waves are electromagnetic in nature. The only difference between the two is frequency: while the radio waves used in radar systems are classified as "microwaves" with frequencies in the gigahertz (GHz) region, visible light waves range in the hundred of terahertz (THz)!

[19]This formula assumes lossless conditions: that none of the wave's energy is converted to heat while traveling through the dielectric. For many situations, this is true enough to assume.

$\epsilon_r$ = Relative permittivity of a gas at a given pressure ($P$) and temperature ($T$)
$\epsilon_{ref}$ = Relative permittivity of the same gas at standard pressure ($P_{ref}$) and temperature ($T_{ref}$)
$P$ = Absolute pressure of gas (bars)
$P_{ref}$ = Absolute pressure of gas under standard conditions ($\approx$ 1 bar)
$T$ = Absolute temperature of gas (Kelvin)
$T_{ref}$ = Absolute temperature of gas under standard conditions ($\approx$ 273 K)

If a radio wave encounters a sudden change in dielectric permittivity, some of that wave's energy will be reflected in the form of another wave traveling the opposite direction. In other words, the wave will "echo" when it reaches a discontinuity. This is the basis of all radar devices:

Figure 12.47: A radar level transmitter showing interface detection

This same principle explains reflected signals in copper transmission lines as well. Any discontinuities (sudden changes in characteristic impedance) along the length of a transmission line will reflect a portion of the electrical signal's power back to the source. In a transmission line, discontinuities may be formed by pinches, breaks, or short-circuits. In a radar level measurement system, any sudden change in permittivity is a discontinuity that reflects some of the incident radio energy back to the source.

The ratio of reflected power to incident (transmitted) power at any interface of materials is called the *power reflection factor* ($R$). This may be expressed as a unitless ratio, or more often as a decibel figure. The relationship between dielectric permittivity and reflection factor is as follows:

$$R = \frac{\left(\sqrt{\epsilon_{r2}} - \sqrt{\epsilon_{r1}}\right)^2}{\left(\sqrt{\epsilon_{r2}} + \sqrt{\epsilon_{r1}}\right)^2}$$

Where,

$R$ = Power reflection factor at interface, as a unitless ratio
$\epsilon_{r1}$ = Relative permittivity (dielectric constant) of the first medium
$\epsilon_{r2}$ = Relative permittivity (dielectric constant) of the second medium

The fraction of incident power transmitted through the interface ($\frac{P_{forward}}{P_{incident}}$) is, of course, the mathematical complement of the power reflection factor: $1 - R$.

For situations where the first medium is air or some other low-permittivity gas, the formula simplifies to the following form (with $\epsilon_r$ being the relative permittivity of the reflecting substance):

$$R = \frac{\left(\sqrt{\epsilon_r} - 1\right)^2}{\left(\sqrt{\epsilon_r} + 1\right)^2}$$

In the previous illustration, the two media were air ($\epsilon_r \approx 1$) and water ($\epsilon_r \approx 80$) – a nearly ideal scenario for strong signal reflection. Given these relative permittivity values, the power reflection factor has a value of 0.638 (63.8%), or -1.95 dB. This means that well over half the incident power reflects off the air/water interface, with the remaining 0.362 (36.2%) of the wave's power making it through the air-water interface and propagating into water. If the liquid in question is gasoline rather than water (having a rather low relative permittivity value of approximately 2), the power reflection ratio will only be 0.0294 (2.94%) or -15.3 dB, with the vast majority of the wave's power successfully penetrating the air-gasoline interface.

The longer version of the power reflection factor formula suggests liquid-liquid interfaces should be detectable using radar, and indeed they are. All that is needed is a sufficiently large difference in relative permittivity between the two liquids to create a strong enough echo to reliably detect. Liquid-liquid interface level measurement with radar works best when the upper liquid has a substantially lesser permittivity value than the lower liquid[20]. A layer of hydrocarbon oil on top of water (or any aqueous solution such as an acid or a caustic) is a good candidate for guided-wave radar level measurement. An example of a liquid-liquid interface that would be very difficult for a radar instrument to detect is water ($\epsilon_r \approx 80$) above glycerin ($\epsilon_r \approx 42$).

If the radar instrument uses a digital network protocol to communicate information with a host system (such as HART or any number of "fieldbus" standards), it may perform as a multi-variable transmitter, transmitting *both* the interface level measurement and the total liquid level measurement simultaneously. This capability is rather unique to guided-wave radar transmitters, and is very useful in some processes because it eliminates the need for multiple instruments measuring multiple levels.

One reason why a lesser-$\epsilon$ fluid above a greater-$\epsilon$ fluid is easier to detect than the inverse is due to the necessity of the signal having to travel through a gas-liquid interface above the liquid-liquid interface. With gases and vapors having such small $\epsilon$ values, the signal would have to pass through the gas-liquid interface first in order to reach the liquid-liquid interface. This gas-liquid interface, having the greatest difference in $\epsilon$ values of any interface within the vessel, will be *most* reflective to radio energy *in both directions*. Thus, only a small portion of the incident wave will ever reach the liquid-liquid interface, and a similarly small portion of the wave reflected off the liquid-liquid interface (which itself is a fraction of the forward wave power that made it through the gas-liquid interface on its way down) will ever make it through the gas-liquid interface on its way *back up* to the instrument. The situation is much improved if the $\epsilon$ values of the two liquid layers are inverted,

[20] Rosemount's "Replacing Displacers with Guided Wave Radar" technical note states that the difference in dielectric constant between the upper and lower liquids *must* be at least 10.

as shown in this hypothetical comparison (all calculations assume no power dissipation along the way, only reflection at the interfaces[21]):

Figure 12.48: A radar level transmitter showing interface transmission and reflection

As can be seen in the illustration, the difference in power received back at the instrument is nearly two to one, just from the upper liquid having the lesser of two identical $\epsilon$ values. Of course, in real life does not offer the luxury of *choosing* which liquid will go on top of the other (this being determined by fluid density), but the appropriate liquid-liquid interface level measurement technology may be chosen, and as can be seen here, certain orientations of $\epsilon$ values are less detectable with radar than others.

Another factor working against radar as a liquid-liquid interface measurement technology for interfaces where the upper liquid has a greater dielectric constant is that fact that many high-$\epsilon$ liquids are aqueous in nature, and water readily dissipates microwave energy. This fact is exploited in microwave ovens, where microwave radiation excites water molecules in the food, dissipating energy in the form of heat. For a radar-based level measurement system consisting of gas/vapor over water over some other (heavier) liquid, the echo signal will be extremely weak because the signal must pass through the "lossy" water layer *twice* before it returns to the radar instrument.

Radio energy losses are important to consider in radar level instrumentation, even when the detected interface is simply gas (or vapor) over liquid. The power reflection factor formula only predicts the ratio of reflected power to incident power *at an interface of substances*. Just because an air-water interface reflects 63.8% of the incident power does not mean 63.8% of the incident power will actually return to the transceiver antenna! Any dissipative losses between the transceiver and the interface(s) of concern will weaken the radio signal, to the point where it may become difficult to distinguish from noise.

Another important factor in maximizing reflected power is the degree to which the microwaves spread out on their way to the liquid interface(s) and back to the transceiver. Guided-wave radar instruments receive a far greater percentage of their transmitted power than non-contact radar

[21] $R = 0.5285$ for the 1/40 interface; $R = 0.02944$ for the 40/80 interface; and $R = 0.6382$ for the 1/80 interface.

instruments because the metal probe used to guide the microwave signal pulses help prevent the waves from spreading (and therefore weakening) throughout the liquids as they propagate. In other words, the probe functions as a transmission line to direct and focus the microwave energy, ensuring a straight path from the instrument into the liquid, and a straight echo return path from the liquid back to the instrument. This is why guided-wave radar is the only practical radar technology for measuring liquid-liquid interfaces.

A critically important factor in accurate level measurement using radar instruments is that the relative permittivity of the upper substance(s) (all media between the radar instrument and the interface of interest) be accurately known. The reason for this is rooted in the dependence of electromagnetic wave propagation velocity to relative permittivity. Recalling the wave velocity formula shown earlier:

$$v = \frac{c}{\sqrt{\epsilon_r}}$$

Where,

$v$ = Velocity of electromagnetic wave through a particular substance

$c$ = Velocity of light in a perfect vacuum ($\approx 3 \times 10^8$ meters per second)

$\epsilon_r$ = Relative permittivity (dielectric constant) of the substance

In the case of a single-liquid application where nothing but gas or vapor exists above the liquid, the permittivity of that gas or vapor must be precisely known. In the case of a two-liquid interface with gas or vapor above, the relative permittivities of *both* gas and upper liquids must be accurately known in order to accurately measure the liquid-liquid interface. Changes in dielectric constant value of the medium or media through which the microwaves must travel and echo will cause the microwave radiation to propagate at different velocities. Since all radar measurement is based on time-of-flight through the media separating the radar transceiver from the echo interface, changes in wave velocity through this media will affect the amount of time required for the wave to travel from the transceiver to the echo interface, and reflect back to the transceiver. Therefore, changes in dielectric constant are relevant to the accuracy of any radar level measurement[22].

Factors influencing the dielectric constant of gases include pressure and temperature, which means the accuracy of a radar level instrument will vary as gas pressure and/or gas temperature vary! Whether or not this variation is substantial enough to consider for any application depends on the desired measurement accuracy and the degree of permittivity change from one pressure/temperature extreme to another. In no case should a radar instrument be considered for any level measurement application unless the dielectric constant value(s) of the upper media are precisely known. This is analogous to the dependence on liquid density that hydrostatic level instruments face. It is futile to attempt level measurement based on hydrostatic pressure if liquid density is unknown, and it is just as futile to attempt level measurement based on radar if the dielectric constants are unknown[23].

As with ultrasonic level instruments, radar level instruments have the ability to measure the level of solid substances in vessels (e.g., powders and granules). The same caveat of repose angle

[22] It should be noted that the dielectric constant of the lowest medium (the liquid in a simple, non-interface, level measurement application) is irrelevant for calibration purposes. The only concern is the propagation time of the signal to and from the level of interest, nothing below it.

[23] For vented-tank level measurement applications where air is the only substance above the point of interest, the relative permittivity is so close to a value of 1 that there is little need for further consideration on this point. Where the relative permittivity of fluids becomes a problem for radar is in high-pressure (non-air) gas applications and liquid-liquid interface applications, especially where the upper substance composition is subject to change.

applicable to ultrasonic level measurement (see page 446), however, is a factor for radar measurement as well. When the particulate solid is not very dense (i.e., much air between particles), the dielectric constant may be rather low, making the material surface more difficult to detect.

Modern radar level instruments provide a wealth of diagnostic information to aid in troubleshooting. One of the most informative is the *echo curve*, showing each reflected signal received by the instrument along the incident signal's path of travel. The following image is a screen capture of a computer display, from software used to configure a Rosemount model 3301 guided-wave radar level transmitter with a coaxial probe:

Figure 12.49: Reflections measured from a radar level transmitter

Pulse `P1` is the *reference* or *fiducial* pulse, resulting from the change in dielectric permittivity between the extended "neck" of the probe (connecting the transmitter to the probe tube) and the coaxial probe itself. This pulse marks the top of the probe, thereby establishing a point of reference for ullage measurement.

This next screen capture shows the same level transmitter measuring a water level that is 8 inches higher than before. Note how pulse `P2` is further to the left (indicating an echo received sooner in time), indicating a lesser ullage (greater level) measurement:

Figure 12.50: Reflections measured from a radar level transmitter when tank level is higher

Several *threshold* settings determine how the transmitter categorizes each received pulse. Threshold `T1` for this particular radar instrument defines which pulse is the reference (fiducial). Thus, the first echo in time to exceed the value of threshold `T1` is interpreted by the instrument to be the reference point. Threshold `T2` defines the upper product level, so the first echo in time to exceed this threshold value is interpreted as the vapor/liquid interface point. Threshold `T3` for this particular transmitter is used to define the echo generated by a liquid-liquid interface. However, threshold `T3` does not appear in this echo plot because the interface measurement option was disabled during this experiment. The last threshold, `T4`, defines the end-of-probe detection. Set at a negative value (just like the reference threshold `T1`), threshold `T4` looks for the first pulse in time to exceed that value and interprets that pulse as the one resulting from the signal reaching the probe's end.

Weak echo signals show up as bumps all along the echo curve. These echoes may be caused by discontinuities along the probe (solid deposits, vent holes, centering spacers, etc.), discontinuities in the process liquid (suspended solids, emulsions, etc.), or even discontinuities in the surrounding process vessel (for non-coaxial probes which exhibit varying degrees of sensitivity to surrounding objects). A challenge in configuring a radar level transmitter is to set the threshold values such that "false" echoes are not interpreted as real liquid or interface levels.

A simple way to eliminate false echoes near the reference point is to set a *null zone* where any echoes are ignored. The upper null zone (UNZ) setting on the Rosemount 3301 radar level transmitter whose screen capture image was shown previously was set to zero, meaning it would be sensitive to any and all echoes near the reference point. If a false echo from a tank nozzle or some other discontinuity near the probe's entry point into the process vessel created a measurement problem, the upper null zone (UNZ) value could be set just beyond that point so the false echo would not be interpreted as a liquid level echo, regardless of the threshold value for `T2`.

Some radar level instruments allow thresholds to be set as curves themselves rather than straight lines. Thus, thresholds may be set high during certain periods along the horizontal (time/distance) axis to ignore false echoes, and set low during other periods to capture legitimate echo signals.

Regardless of how null zones and thresholds are set for any guided-wave radar level transmitter, the technician must be aware of *transition zones* near the extremes of the probe length. Measurements of liquid level or interface level within these zones may not be accurate or even linearly responsive. Thus, it is strongly advised to range the instrument in such a way that the lower- and upper-range values (LRV and URV) lie between the transition zones:

Figure 12.51: Transition zones for a radar level transmitter

The size of these transition zones depends on both the process substances and the probe type[24]. The instrument manufacturer will provide the appropriate data for determining transition zone dimensions.

## 12.6 Laser level measurement

The least-common form of echo-based level measurement is *laser*, which uses pulses of laser light reflected off the surface of a liquid to detect the liquid level. Perhaps the most limiting factor with laser measurement is the necessity of having a sufficiently reflective surface for the laser light to "echo" off of. Many liquids are not reflective enough for this to be a practical measurement technique, and the presence of dust or thick vapors in the space between the laser and the liquid will disperse the light, weakening the light signal and making the level more difficult to detect.

However, lasers have been applied with great success in measuring distances between objects. Applications of this technology include motion control on large machines, where a laser points at a moving reflector, the laser's electronics calculating distance to the reflector based on the amount of time it takes for the laser "echo" to return. The advent of mass-produced, precision electronics has made this technology practical and affordable for many applications. It is even possible for the average American consumer to purchase laser "tape measures" for use in building construction.

[24]Probe mounting style will also influence the lower transition zone, in the case of flexible probes anchored to the bottom of the process vessel.

## 12.7 Weight

*Weight-based* level instruments sense process level in a vessel by directly measuring the weight of the vessel. If the vessel's empty weight (*tare weight*) is known, process weight becomes a simple calculation of total weight minus tare weight. Obviously, weight-based level sensors can measure both liquid and solid materials, and they have the benefit of providing inherently linear mass storage measurement[25]. *Load cells* (strain gauges bonded to a steel element of precisely known modulus) are typically the primary sensing element of choice for detecting vessel weight. As the vessel's weight changes, the load cells compress or relax on a microscopic scale, causing the strain gauges inside to change resistance. These small changes in electrical resistance become a direct indication of vessel weight.

The following photograph shows three bins, each one supported by pillars equipped with load cells near their bases:

Figure 12.52: Pillars showing load cells used for weight-based level measurement

One very important caveat for weight-based level instruments is to isolate the vessel from any

[25]Regardless of the vessel's shape or internal structure, the measurement provided by a weight-sensing system is based on the true mass of the stored material. Unlike height-based level measurement technologies (float, ultrasonic, radar, etc.), no characterization will ever be necessary to convert a measurement of height into a measurement of mass.

external mechanical stresses generated by pipes or machinery. The following illustration shows a typical installation for a weight-based measurement system, where all pipes attaching to the vessel do so through flexible couplings, and the weight of the pipes themselves is borne by outside structures through *pipe hangers*:

Figure 12.53: Isolation of the vessel is required for weight-based level measurement

Stress relief is very important because any forces acting upon the storage vessel will be interpreted by the load cells as more or less material stored in the vessel. The only way to ensure that the load cell's measurement is a direct indication of material held inside the vessel is to ensure that no other forces act upon the vessel except the gravitational weight of the material.

An interesting problem associated with load cell measurement of vessel weight arises if there are ever electric currents traveling through the load cell(s). This is not a normal state of affairs, but it can happen if maintenance workers incorrectly attach arc welding equipment to the support structure of the vessel, or if certain electrical equipment mounted on the vessel such as lights or motors develop ground faults. The electronic amplifier circuits interpreting a load cell's resistance will detect voltage drops created by such currents, interpreting them as changes in load cell resistance and therefore as changes in material level. Sufficiently large currents may even cause permanent damage to load cells, as is often the case when the currents in question are generated by arc welding equipment.

A variation on this theme is the so-called *hydraulic load cell* which is a piston-and-cylinder mechanism designed to translate vessel weight directly into hydraulic (liquid) pressure. A normal pressure transmitter then measures the pressure developed by the load cell and reports it as material

weight stored in the vessel. Hydraulic load cells completely bypass the electrical problems associated with resistive load cells, but are more difficult to network for the calculation of total weight (using multiple cells to measure the weight of a large vessel).

## 12.8 Capacitive

*Capacitive* level instruments measure electrical capacitance of a conductive rod inserted vertically into a process vessel. As process level increases, capacitance increases between the rod and the vessel walls, causing the instrument to output a greater signal.

The basic principle behind capacitive level instruments is the capacitance equation:

$$C = \frac{\epsilon A}{d}$$

Where,

$C$ = Capacitance

$\epsilon$ = Permittivity of dielectric (insulating) material between plates

$A$ = Overlapping area of plates

$d$ = Distance separating plates

The amount of capacitance exhibited between a metal rod inserted into the vessel and the metal walls of that vessel will vary only with changes in permittivity ($\epsilon$), area ($A$), or distance ($d$). Since $A$ is constant (the interior surface area of the vessel is fixed, as is the area of the rod once installed), only changes in $\epsilon$ or $d$ can affect the probe's capacitance.

Capacitive level probes come in two basic varieties: one for conductive liquids and one for non-conductive liquids. If the liquid in the vessel is conductive, it cannot be used as the dielectric (insulating) medium of a capacitor. Consequently, capacitive level probes designed for conductive liquids are coated with plastic or some other dielectric substance, so the metal probe forms one plate of the capacitor and the conductive liquid forms the other:

Figure 12.54: Diagram of capacative level measurement for conductive liquids

In this style of capacitive level probe, the variables are permittivity ($\epsilon$) and distance ($d$), since a rising liquid level displaces low-permittivity gas and essentially acts to bring the vessel wall electrically closer to the probe. This means total capacitance will be greatest when the vessel is full ($\epsilon$ is greatest and effective distance $d$ is at a minimum), and least when the vessel is empty ($\epsilon$ of the gas is in effect, and over a much greater distance).

If the liquid is non-conductive, it may be used as the dielectric itself, with the metal wall of the storage vessel forming the second capacitor plate:

Figure 12.55: Diagram of capacative level measurement for nonconductive liquids

In this style of capacitive level probe, the variable is permittivity ($\epsilon$), provided the liquid has a substantially greater permittivity than the vapor space above the liquid. This means total capacitance will be greatest when the vessel is full (average permittivity $\epsilon$ is at a maximum), and least when the vessel is empty.

Permittivity of the process substance is a critical variable in the non-conductive style of capacitance level probe, and so good accuracy may be obtained with this kind of instrument only if the process permittivity is accurately known. A clever way to ensure good level measurement accuracy when the process permittivity is not stable over time is to equip the instrument with a special *compensating probe* (sometimes called a *composition probe*) below the LRV point in the vessel that will always be submerged in liquid. Since this compensating probe is always immersed, and always experiences the same $A$ and $d$ dimensions, its capacitance is purely a function of the liquid's permittivity ($\epsilon$). This gives the instrument a way to continuously measure process permittivity, which it then uses to calculate level based on the capacitance of the main probe. The inclusion of a compensating probe to measure and compensate for changes in liquid permittivity is analogous to the inclusion of a third pressure transmitter in a hydrostatic *tank expert* system to continuously measure and compensate for liquid density. It is a way to correct for changes in the one remaining system variable that is not related to changes in liquid level.

Capacitive level instruments may be used to measure the level of solids (powders and granules) in addition to liquids. In these applications, and solid substance is almost always non-conductive, and therefore the permittivity of the substance becomes a factor in measurement accuracy. This can be problematic, as moisture content variations in the solid may greatly affect permittivity, as can variations in granule size. They are not known for great accuracy, though, primarily due to sensitivity to changes in process permittivity and errors caused by stray capacitance in probe cables.

## 12.9 Radiation

Certain types of nuclear radiation easily penetrates the walls of industrial vessels, but is attenuated by traveling through the bulk of material stored within those vessels. By placing a radioactive source on one side of the vessel and measuring the radiation making it through to the other side of the vessel, an approximate indication of level within that vessel may be obtained.

The three most common forms of nuclear radiation are *alpha particles* ($\alpha$), *beta particles* ($\beta$), and *gamma rays* ($\gamma$). Alpha particles are helium nuclei (2 protons bound together with 2 neutrons) ejected at high velocity from the nuclei of certain decaying atoms. They are easy to detect, but have very little penetrating power and so are not used for industrial level measurement. Beta particles are electrons[26] ejected at high velocity from the nuclei of certain decaying atoms. Like alpha particles, though, they have little penetrating power and so are not used for industrial level measurement. Gamma rays, on the other hand, are electromagnetic in nature (like X-rays and light waves) and have great penetrating power. This form of radiation is the most common used in industrial level measurement.

One of the most effective methods of shielding against gamma ray radiation is with very dense substances such as lead or concrete. This is why the source boxes holding gamma-emitting radioactive pellets are lined with lead, so the radiation escapes only in the direction intended:

Figure 12.56: Diagram of nuclear level measurement showing a collimated source

These "sources" may be locked out for testing and maintenance by moving a lever that hinges a lead shutter over the "window" of the box. This lead shutter acts as an on/off switch for the radioactive source. The lever actuating the shutter typically has provisions for lock-out/tag-out so a maintenance person may place a padlock on the lever and prevent anyone else from "turning on" the source during maintenance.

[26] Beta particles are *not* orbital electrons, but rather the product of elementary particle (neutron) decay in an atom's nucleus. These electrons are spontaneously generated and subsequently ejected from the nucleus of the atom.

The accuracy of radiation-based level instruments varies with the stability of process fluid density, vessel wall coating, source decay rates, and detector drift. Given these error variables and the additional need for NRC (Nuclear Regulatory Commission) licensing to operate such instruments at an industrial facility, radiation instruments are typically used where no other instrument can possibly function. Examples include the level measurement of highly corrosive or toxic process fluids where penetrations into the vessel must be minimized and where piping requirements make weight-based measurement impractical, as well as processes where the internal conditions of the vessel are too violent for any instrument to survive (e.g., delayed coking vessels in the oil refining industry).

## 12.10 Level switches

A *level switch* is one detecting the level of liquid or solid (granules or powder) in a vessel. Level switches often use floats as the level-sensing element, the motion of which actuates one or more switch contacts.

Recall from section 9.1 that the "normal" status of a switch is the resting condition of *no stimulation*. A level switch will be in its "normal" status when it senses minimum level (e.g., an empty vessel). For a level switch, "normal" status is any fluid level *below* the trip threshold of the switch.

### 12.10.1 Float-type level switches

Figure 12.57: Schematic symbols for level switches

Two water level switches appear in this photograph of a steam boiler. The switches sense water level in the steam drum of the boiler. Both water level switches are manufactured by the Magnetrol corporation:

Figure 12.58: An industrial level switch installation

The switch mechanism is a mercury tilt bulb, tilted by a magnet's attraction to a steel rod lifted into position by a float. The float directly senses liquid level, which positions the steel rod either closer to or further away from the magnet. If the rod comes close enough to the magnet, the mercury bottle will tilt and change the switch's electrical status.

### 12.10.2 Tuning fork level switches

This level switch uses a metal *tuning fork* structure to detect the presence of a liquid or solid (powder or granules) in a vessel:

Figure 12.59: An industrial tuning fork level switch

An electronic circuit continuously excites the tuning fork, causing it to mechanically vibrate. When the prongs of the fork contact anything with substantial mass, the resonant frequency of the structure dramatically decreases. The circuit detects this change and indicates the presence of material contacting the fork. The forks' vibrating motion tends to shake off any accumulated material, such that this style of level switch tends to be resistant to fouling.

### 12.10.3 Paddle level switches

A more primitive variation on the theme of a "tuning fork" level switch is the *rotating paddle* switch, used to detect the level of powder or granular solid material. This level switch uses an electric motor to slowly rotate a metal paddle inside the process vessel. If solid material rises to the level of the paddle, the material's bulk will place a mechanical load on the paddle. A torque-sensitive switch mechanically linked to the motor actuates when enough torsional effort is detected on the part of the motor. A great many level switches of this design sold in the United States under the trade-name *Bindicator* (so-called because they detected the level of solid material in storage *bins*).

A "Bindicator" style of level switch appears in this photograph (black, against a white-painted hopper), used to detect the presence of soda ash powder in a hopper at a water treatment plant:

Figure 12.60: An industrial paddle level switch

### 12.10.4 Ultrasonic level switches

Yet another style of electronic level switch uses ultrasonic sound waves to detect the presence of process material (either solid or liquid) at one point:

Figure 12.61: An industrial ultrasonic level switch

Sound waves pass back and forth within the gap of the probe, sent and received by piezoelectric transducers. The presence of any substance other than gas within that gap affects the received audio power, thus signaling to the electronic circuit within the bulkier portion of the device that process

level has reached the detection point. The lack of moving parts makes this probe quite reliable, although it may become "fooled" by heavy fouling.

### 12.10.5 Capacitive level switches

Another electronic liquid level switch technology is *capacitive*: sensing level by changes in electrical capacitance between the switch and the liquid. The following photograph shows a couple of capacitive switches sensing the presence of water in a plastic storage vessel:

Figure 12.62: Industrial capacitive level switches

### 12.10.6 Conductive level switches

Perhaps the simplest (and oldest) form of electrical level detection is where a set of metal electrodes contacts the process material to form a complete electrical circuit, actuating a relay. This type of switch, of course, only works with granular solids and liquids that are electrically conductive (e.g., potable or dirty water, acids, caustics, food liquids, coal, metal powders) and not with non-conducting materials (e.g., ultra-pure water, oils, ceramic powders).

A legacy design for conductive level switches is the model 1500 "induction relay" manufactured by B/W Controls (now part of Ametek), using a special transformer/relay to generate an isolated AC probe voltage and sense the presence of liquid:

Figure 12.63: Diagram of a conductive level switch

Line voltage (120 VAC) energizes the primary coil, sending a magnetic field through the laminated iron core of the relay. This magnetic field easily passes through the center of the secondary coil when the secondary circuit is open (no liquid closing the probe circuit), thus completing the magnetic "circuit" in the core. However, when a circuit is completed by liquid level rising to contact both probes, the secondary coil's resulting current "bucks" the magnetic flux through its center, causing more magnetic flux to bypass to the end poles where it attracts the iron armature toward the core frame[27]. This physical attraction actuates switch contacts which then signal the presence of liquid level at the probes.

[27]The reason for this opposition is rooted in the roles of primary and secondary coils as power *load* and *source*, respectively. The voltage across each coil is a function of Faraday's Law of Electromagnetic Induction: $V = N\frac{d\phi}{dt}$. However, since the primary coil acts as a load (drawing power from the 120 VAC source) and the secondary coil acts as a source (sending power to the probes), the directions of current through the two coils will be opposite despite

The "transformer" design of this particular conductive level switch not only provides electrical isolation between the probes and the energizing (120 VAC) circuit, but it also enables a wide range of detection voltages to be generated for the probes just by altering the number of wire "turns" in the secondary coil. The B/W Controls model 1500 inductive relay is available in a variety of AC voltage output levels, ranging from 12 VAC (for detecting metallic substances) to 800 VAC for use with demineralized water such as that found in steam boiler systems.

More modern variations on the same design theme use much lower AC voltages to energize the probes, employing sensitive semiconductor amplifier circuits to detect probe current and signal liquid level[28].

## 12.11 Level sensor accessories

Disturbances in the liquid tend to complicate liquid level measurement. These disturbances may result from liquid introduced into a vessel above the liquid level (splashing into the liquid's surface), the rotation of agitator paddles, and/or turbulent flows from mixing pumps. Any source of turbulence for the liquid surface (or liquid-liquid interface) is especially problematic for echo-type level sensors, which *only* sense interfaces between vapors and liquids, or liquids and liquids.

If it is not possible to eliminate disturbances inside the process vessel, a relatively simple accessory one may add to the process vessel is a vertical length of pipe called a *stilling well*. To understand the principle of a stilling well, first consider the application of a hydraulic oil reservoir where oil level must be continuously measured. The oil flow in and out of this reservoir will cause problems for the displacer element:

Figure 12.64: Disturbances affect the ability to properly measure level

their common voltage polarities. The secondary coil's opposite current direction causes an opposing magnetic force in the core section running through the center of the secondary coil, reducing the magnetic flux in that section. In a normal power transformer, this reduction in magnetic flux caused by secondary current is also felt by the primary coil (since there is but one magnetic "path" in a power transformer's core), which then causes the primary coil to draw more current and re-establish the core flux at its original magnitude. With the inductive relay, however, the opposing magnetic force created by the secondary coil simply forces more of the primary coil's magnetic flux to bypass to the alternate route: through the armature.

[28]The B/W Controls model 5200 solid-state relay, for example, uses only 8 volts AC at the probe tips.

A section of vertical pipe installed in the reservoir around the displacer will serve as a shield to all the turbulence in the rest of the reservoir. The displacer element will no longer be subject to a horizontal blast of oil entering the reservoir, nor any wave action to make it bob up and down. This section of pipe *quiets*, or *stills*, the oil surrounding the displacer, making it easier to measure oil level:

Figure 12.65: A stilling well eliminates disturbances

Stilling wells may be used in conjunction with many types of level instruments: floats, displacers, ultrasonic, radar, and laser to name a few. If the process application necessitates liquid-liquid interface measurement, however, the stilling well must be properly installed to ensure the interface level inside the well match the interface levels in the rest of the vessel. Consider this example of using a stilling well in conjunction with a tape-and-float system for interface measurement:

Figure 12.66: Correct and incorrect use of a stilling well when a liquid-liquid interface exists

In the left-hand installation where the stilling well is completely submerged, the interface levels will always match. In the right-hand installation where the top of the stilling well extends above the total liquid level, however, the two levels may not always match.

This potential problem for the non-submerged stilling well is graphically illustrated here:

Figure 12.67: Potential problems using a stilling well when a liquid-liquid interface exists

The problem here is analogous to what is seen with sightglass-style level gauges: interfaces may be reliably indicated if and only if both ends of the sightglass are submerged (see page 404 for an illustrated description of the problem).

If it is not possible or practical to ensure complete submersion of the stilling well, an alternative technique is to drill holes or cut slots in the well to allow interface levels to equalize inside and outside of the well tube:

Figure 12.68: Potential problems using a stilling well when a liquid-liquid interface exists

Such equalization ports are commonly found as a standard design feature on coaxial probes for guided-wave radar level transmitters, where the outer tube of the coaxial transmission line acts as a sort of stilling well for the fluid. Coaxial probes are typically chosen for liquid-liquid interface radar measurement applications because they do the best job of preventing dispersion of the radio energy, but the "stilling well" property of a coaxial probe practically necessitates these equalization ports to ensure the interface level within the probe always matches the interface level in the rest of the vessel[29].

[29] So much of the incident power is lost as the radar signal partially reflects off the gas-liquid interface, then the liquid-liquid interface, then *again* through the gas-liquid interface on its return trip to the instrument that every care must be taken to ensure optimum received signal strength. While twin-lead probes have been applied in liquid-liquid interface measurement service, the coaxial probe design is still the best for maintaining radar signal integrity.

## 12.12 Level instrument installation details

| BILL OF MATERIAL | | |
|---|---|---|
| ITEM | QTY. | DESCRIPTION |
| 1 | 1 | ¾"x½" SWAGE |
| 2 | 1 | ½" TEE |
| 3 | 2 | ½" CONNECTOR TUBE x MALE PIPE |
| 4 | 1 | ½" PLUG |
| 5 | A/R | ½" TUBING BUNDLE |
| 6 | 1 | ½" MANIFOLD VALVE |
| 7 | 1 | ½" VENT PROTECTOR |

GENERAL NOTES

A. TUBING AND FITTINGS SHALL BE 1/2" DIAMETER 316SS UNLESS OTHERWISE SPECIFIED.

B. LOCATE TRANSMITTER LEVEL WITH LOWER PIPE CONNECTION. IF THERE ARE TWO TRANSMITTERS, LOCATE AT SAME ELEVATION.

C. INSULATION MATERIAL SHALL MEET THE REQUIREMENTS OF THE PIPING SPECIFICATION.

Figure 12.69: Installation detail for a remotely-located inferential level transmitter on an open vessel

BILL OF MATERIAL

| ITEM | QTY. | DESCRIPTION |
|---|---|---|
| 1 | 2 | ¾"x½" SWAGE |
| 2 | 8 | ½" CONNECTOR TUBE x MALE PIPE |
| 3 | 4 | ½" TEE |
| 4 | 4 | ½" PLUG |
| 5 | 2 | ½" BALL VALVE |
| 6 | 2 | ½"x3" NIPPLE |
| 7 | A/R | ½" TUBING |
| 8 | 1 | ½" MANIFOLD VALVE |
| 9 | A/R | ½" TUBING BUNDLE |

GENERAL NOTES

A. TUBING AND FITTINGS SHALL BE 1/2" DIAMETER 316SS UNLESS OTHERWISE SPECIFIED.

B. LOCATE TRANSMITTER LEVEL WITH LOWER PIPE CONNECTION.

C. INSULATION MATERIAL SHALL MEET THE REQUIREMENTS OF THE PIPING SPECIFICATION.

Figure 12.70: Installation detail for a remotely-located inferential level transmitter on an closed vessel

# References


[1] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1.

[2] Boiler Drum Level Transmitter Calibration, 00800-0100-3055, Emerson Process Management Rosemount Division, Chanhassen, MN, 2001.

[3] Autolevel, AN 01C22A01-01E, Yokogawa Electric Corporation, Sugar Land, TX, 2006.

[4] D. Brumbi, Ed., Fundamentals of Radar Technology for Level Gauging, 4th ed. Duisburg, Germany: Krohne Messtechnik GmbH & Co. KG, 2003.

[5] Bubble Tube Installations For Liquid Level, Density, and Interface Measurements, MI 020-328, The Foxboro Company, Foxboro, MA, 1988.

[6] D. Brumbi, Ed., Fundamentals of Radar Technology for Level Gauging, 4th ed. Duisburg, Germany: Krohne Messtechnik GmbH & Co. KG, 2003.

[7] A. E. Fribance, Industrial Instrumentation Fundamentals. New York, NY: McGraw-Hill, 1962.

[8] H. P. Kallen, Handbook of Instrumentation and Controls. New York, NY: McGraw-Hill, 1961.

[9] Level Measurement Technology: Radar, 00816-0100-3209, Emerson Process Management Rosemount Division, Chanhassen, MN, 1999.

[10] H. P. Kallen, "Iaea candu instrumentation & control course," Shanghai, China, 1998.

[11] Model 1151 Alphaline Pressure Transmitters, 00809-0100-4360, Emerson Process Management Rosemount Division, Chanhassen, MN, 1997.

[12] Replacing Displacers with Guided Wave Radar, 3300_2_02_CA, Emerson Process Management Rosemount Division, Chanhassen, MN, 2003.

[13] "The art of tank gauging for safety and precision," IN 4416.650 Rev. 6, Enraf B.V., Delft, The Netherlands.

# Chapter 13

# Flow Measurement

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: rangeability, turndown ratio
2. Describe the basic construction and theory of operation of the following types of flow detectors:
   (a) Orifice plates
   (b) Venturi tube
   (c) Pitot tube
   (d) Flow nozzle
   (e) Laminar
   (f) Variable area (e.g., rotameter)
   (g) Turbine
   (h) Vortex
   (i) Electromagnetic flow meter
   (j) Ultrasonic flow detector
   (k) Thermal
   (l) Positive displacement
3. Draw P&ID symbols used for flow detection equipment, including sensing elements, transmitters, and protective wells.
4. Describe density compensation of a volumetric flow instrument to determine mass flow. Include a discussion of the reason for density compensation and the parameters used.
5. State the relationship between flow and differential pressure in a head-type flow element
6. Perform calculations to size flow elements
7. Describe the environmental concerns that can affect the accuracy and reliability of flow sensing instrumentation.
8. Discuss the failure modes of the various types of flow indication instruments.
9. Discuss and sketch methods of flow sensor installation, including advantages and disadvantages.

Fluid flow may be measured volumetrically or by mass. *Volumetric flow* is expressed in volume units (e.g., gallons, liters, cubic inches) per unit time. *Mass flow* is expressed in mass units (slugs, kilograms, pounds-mass) per unit time.

Liquids are essentially incompressible: that is, they do not easily yield in volume to applied pressure. Gases and vapors, however, easily change volume under the influence of changing pressure. In other words, a gas will yield to an increasing pressure by decreasing in volume as the gas molecules are forced closer together. This makes volumetric flow measurement more complex for gases than for liquids. To begin, a standardized volumetric measurement for a gas must be agreed upon, since the volume is so easily subject to change. This is done by agreeing on standard "base-line" pressures and temperatures under which a particular gas volume is specified.

For example, a gas flow rate of 900 scfm (*standard* cubic feet per minute) refers to 900 cubic feet of gas flowing per minute of time, if that gas flowstream were subjected to standard atmospheric pressure at 60 $^o$F and is free of moisture (0% relative humidity). The actual volume (acfm) of gas moving through that pipe each minute under pressurized conditions will likely occupy far less than 900 cubic feet, due to physical compression (reduction of volume resulting from increased pressure) of the gas. However, the unit of "standard" cubic feet per minute gives people a common frame of reference in which to specify flow rates of gases. Because the density of a dry gas is fixed by standard temperature and pressure when the units are in scfm, the measured value is actually a mass or molar flow rate, as opposed to a volumetric flow rate, even though the units appear to indicate volume.

Due to the relative incompressibility of liquids, no one bothers to express volumetric liquid flow in units such as *standard gallons per minute*. Though liquids do in fact expand and contract (slightly) with changes in temperature, changes in liquid density due to applied pressure are quite insignificant, and thus it is generally not a great enough factor to warrant the expression of flowing volume in "standard" units as it is with gas flows.

Most flowmeters have a range over which they can measure flow to within a given accuracy, known as *rangeability*. The *turndown ratio* is the ratio of the upper range value to the lower range value that are used to define the rangeability. Most flowmeters work on principles that will not allow them to measure down to zero flow, so a design engineer should understand the limits of the flowmeter. Many instrument vendors will provide an upper range value and a turndown ratio (instead of the accompanying lower range value) as though they don't want to admit that their device can't measure down to zero flow. However, since process conditions may require flow measurement at low values, the design engineer should always calculate the lower range value to be sure that a single flowmeter will be adequate.

**Example 13.13** A flow element/transmitter combination has a range specified as 2 gpm - 30 gpm. What is the turndown ratio of this instrument?

**Answer to Example 13.13**

$$Turndown\ Ratio = \frac{URV}{LRV} = \frac{30\ gpm}{2\ gpm}$$

Therefore, the turndown ratio is 15:1.

**Example 13.14** A flow element/transmitter combination can measure up to 30 gpm with ±1% accuracy with a turndown ratio of 3:1. What is the minimum accurate flow measurement?

**Answer to Example 13.14**

$$Turndown\ Ratio = \frac{URV}{LRV}$$

Rearranging:

$$LRV = \frac{URV}{Turndown\ Ratio} = \frac{30\ gpm}{\frac{3}{1}} = 30\ gpm \times \frac{1}{3}$$

Therefore, the minimum accurate lower range value is 10 gpm.

## 13.1 Pressure-based flowmeters

All masses require force to accelerate (think of this in terms of the mass generating a reaction force as a result of being accelerated). This is quantitatively expressed by Newton's Second Law of Motion:

Figure 13.1: The second law of motion

All fluids possess mass, and therefore require force to accelerate just like solid masses. When considering a quantity of fluid confined inside a pipe[1], with that fluid quantity having a mass equal to its volume multiplied by its mass density ($m = \rho V$, where $\rho$ is the fluid's mass per unit volume), the force required to accelerate that fluid "plug" is calculated just the same as for a solid mass:

[1] Sometimes referred to as a *plug* of fluid.

Figure 13.2: The second law of motion for plug flow

Since this accelerating force is applied on the cross-sectional area of the fluid plug, it may be expressed as a *pressure*, the definition of pressure being force per unit area:

$$F = \rho V a$$

$$\frac{F}{A} = \rho \frac{V}{A} a$$

$$P = \rho \frac{V}{A} a$$

Since the rules of algebra require *both* sides of the force equation to be divided by area, it leaves a fraction of volume over area ($\frac{V}{A}$) on the right-hand side of the equation. This fraction has a physical meaning, since the volume of a cylinder divided by the area of its circular face is simply the length of that cylinder:

$$P = \rho \frac{V}{A} a$$

$$P = \rho l a$$

When this is applied to the illustration of the fluid mass, it makes sense: the pressure described by the equation is actually a *differential*[2] pressure drop from one side of the fluid mass to the other, with the length variable ($l$) describing the spacing between the differential pressure ports:

[2]What really matters in Newton's Second Law equation is the *resultant* force causing the acceleration. This is the vector sum of all forces acting on the mass. Likewise, what really matters in this scenario is the *resultant* pressure acting on the fluid plug, and this resultant pressure is the difference of pressure between one face of the plug and the other, since those two pressures impart two forces on the fluid mass in direct opposition to each other.

Figure 13.3: A pressure drop is developed across a differential length of moving fluid

This means a fluid can be accelerated by applying a difference of pressure across its length. The amount of pressure applied will be in direct proportion to the density of the fluid and its rate of acceleration. Conversely, a fluid's rate of acceleration may be determined by measuring the pressure developed across a distance over which it accelerates.

A fluid may be forced to accelerate by altering its natural flow path. The difference of pressure generated by this acceleration will indirectly indicate the rate of acceleration. Since the acceleration seen from a change in flow path is a direct function of how fast the fluid was originally moving, the acceleration (and therefore the pressure drop) indirectly indicates fluid flow rate.

A very common way to cause linear acceleration in a moving fluid is to pass the fluid through a constriction in the pipe, thereby increasing its velocity (remember that the definition of acceleration is a change in velocity). The following illustrations show several devices used to linearly accelerate moving fluids when placed in pipes, with differential pressure transmitters connected to measure the pressure drop resulting from this acceleration:

Figure 13.4: Examples of flow elements that work by accelerating a fluid

Another way a fluid may be accelerateed is to force it to turn a corner through a pipe fitting called an *elbow*. This will generate radial acceleration, causing a pressure difference between the outside and inside of the elbow which may be measured by a differential pressure transmitter:

Figure 13.5: An elbow can be used as a differential flow element

The pressure tap located on the outside of the elbow's turn registers a greater pressure than the tap located on the inside of the elbow's turn, due to the inertial force of the fluid's mass being "flung" to the outside of the turn as it rounds the corner.

Yet another way to cause fluid acceleration is to force it to *decelerate* by bringing a portion of it to a full stop. Elements that work on this principle can measure the difference between the total pressure and the static pressure to determine the dynamic pressure, or measure the dynamic pressure by the measuring the drag force on an object. The dynamic pressure tells how fast it was originally flowing. A few devices working on this principle are shown here:

Figure 13.6: Examples of flow elements that work by decelerating a fluid

The following subsections in this flow measurement chapter explore different primary sensing elements (PSEs) used to generate differential pressure in a moving fluid stream. Despite their very different designs, they all operate on the same fundamental principle: causing a fluid to accelerate (or decelerate) by changing its flow path, and thus generating a measurable pressure difference. The following subsection will introduce a device called a *venturi tube* used to measure fluid flow rates, and derive mathematical relationships between fluid pressure and flow rate starting from basic physical conservation laws.

### 13.1.1 Venturi tubes and basic principles

The standard example used to demonstrate pressure change in a fluid stream is the *venturi tube*: a pipe purposefully narrowed to create a region of low pressure. If the fluid going through the venturi tube is a liquid under relatively low pressure, the pressure at different points in the tube may be shown by means of *piezometers*, which are transparent tubes allowing varying liquid column heights. The greater the height of liquid column in the piezometer, the greater the pressure at that point in the flowstream:

Figure 13.7: Diagram of a venturi tube

As indicated by the piezometer liquid heights, pressure at the constriction (point 2) is the least, while pressures at the wide portions of the venturi tube (points 1 and 3) are the greatest. This is a counter-intuitive result, but it has a firm grounding in the physics of mass and energy conservation. If no energy is assumed to be added (by a pump) or lost (due to friction) as fluid travels through this pipe, then the Law of Energy Conservation describes a situation where the fluid's energy must remain constant at all points in the pipe as it travels through. When assuming no fluid joins this flowstream from another pipe, or is lost from this pipe through any leaks, then the Law of Mass Conservation describes a situation where the fluid's mass flow rate must remain constant at all points in the pipe as it travels through.

So long as fluid density remains fairly constant[3], fluid velocity must increase as the cross-sectional area of the pipe decreases, as described by the Law of Continuity:

$$A_1\overline{v_1} = A_2\overline{v_2}$$

[3]This is a very sound assumption for liquids, and a fair assumption for gases when pressure changes through the venturi tube are modest.

Rearranging variables in this equation to place velocities in terms of areas:

$$\frac{\overline{v_2}}{\overline{v_1}} = \frac{A_1}{A_2}$$

This equation says that the ratio of fluid velocity between the narrow throat (point 2) and the wide mouth (point 1) of the pipe is the same ratio as the mouth's area to the throat's area. So, if the mouth of the pipe had an area five times as great as the area of the throat, then the fluid velocity in the throat may be expectedc to be five times as great as the velocity at the mouth. Simply put, the narrow throat causes the fluid to accelerate from a lower velocity to a higher velocity.

From the study of energy in physics, kinetic energy is proportional to the square of a mass's velocity ($E_k = \frac{1}{2}mv^2$). If the fluid molecules increase velocity as they travel through the venturi tube's throat, the kinetic energies of those molecules must increase as well. However, the total energy at any point in the fluid stream must remain constant, because no energy is added to or taken away from the stream in this simple fluid system. Therefore, if kinetic energy increases at the throat, potential energy must correspondingly decrease to keep the total amount of energy constant at any point in the fluid.

Potential energy may be manifest as height above ground, or as pressure in a fluid system. Since this venturi tube is level with the ground, there cannot be a height change to account for a change in potential energy. Therefore, there *must* be a change of pressure ($P$) as the fluid travels through the venturi throat. The Laws of Mass and Energy Conservation invariably lead to this conclusion: fluid pressure must decrease as it travels through the narrow throat of the venturi tube.

Conservation of energy at different points in a fluid stream is neatly expressed in *Bernoulli's Equation* as a constant sum of elevation, pressure, and velocity "heads":

$$z_1\rho g + \frac{v_1^2\rho}{2} + P_1 = z_2\rho g + \frac{v_2^2\rho}{2} + P_2$$

Where,

$z$ = Height of fluid (from a common reference point, usually ground level)
$\rho$ = Mass density of fluid
$g$ = Acceleration of gravity
$v$ = Velocity of fluid
$P$ = Pressure of fluid

Bernoulli's equation will be used to develop a precise mathematical relationship between pressure and flow rate in a venturi tube. To simplify the task, the following assumptions for the venturi tube system are made:

- No energy lost or gained in the venturi tube (all energy is conserved)
- No mass lost or gained in the venturi tube (all mass is conserved)
- Fluid is incompressible
- Venturi tube centerline is level (no height changes to consider)

Applying the last two assumptions to Bernoulli's equation, the "elevation head" term drops out of both sides, since $z$, $\rho$, and $g$ are equal at all points in the system:

$$\frac{v_1^2\rho}{2} + P_1 = \frac{v_2^2\rho}{2} + P_2$$

Now, algebraically rearrange this equation to show pressures at points 1 and 2 in terms of velocities at points 1 and 2:

$$\frac{v_2^2\rho}{2} - \frac{v_1^2\rho}{2} = P_1 - P_2$$

Factoring $\frac{\rho}{2}$ out of the velocity head terms:

$$\frac{\rho}{2}(v_2^2 - v_1^2) = P_1 - P_2$$

The Continuity equation shows the relationship between velocities $v_1$ and $v_2$ and the areas at those points in the venturi tube, assuming constant density ($\rho$):

$$A_1 v_1 = A_2 v_2$$

Specifically, this equation must be rearranged to define $v_1$ in terms of $v_2$ in order to substitute it into Bernoulli's equation:

$$v_1 = \left(\frac{A_2}{A_1}\right) v_2$$

Performing the algebraic substitution:

$$\frac{\rho}{2}(v_2^2 - \left[\left(\frac{A_2}{A_1}\right) v_2\right]^2) = P_1 - P_2$$

Distributing the "square" power:

$$\frac{\rho}{2}(v_2^2 - \left(\frac{A_2}{A_1}\right)^2 v_2^2) = P_1 - P_2$$

Factoring $v_2^2$ out of the outer parentheses set:

$$\frac{\rho v_2^2}{2}(1 - \left(\frac{A_2}{A_1}\right)^2) = P_1 - P_2$$

Solving for $v_2$, step by step:

$$\frac{\rho v_2^2}{2} = \left(\frac{1}{1 - \left(\frac{A_2}{A_1}\right)^2}\right)(P_1 - P_2)$$

$$\rho v_2^2 = 2\left(\frac{1}{1 - \left(\frac{A_2}{A_1}\right)^2}\right)(P_1 - P_2)$$

$$v_2^2 = 2\left(\frac{1}{1 - \left(\frac{A_2}{A_1}\right)^2}\right)\left(\frac{P_1 - P_2}{\rho}\right)$$

$$v_2 = \sqrt{2}\frac{1}{\sqrt{1-\left(\frac{A_2}{A_1}\right)^2}}\sqrt{\frac{P_1 - P_2}{\rho}}$$

The result shows how to solve for fluid velocity at the venturi throat ($v_2$) based on a difference of pressure measured between the mouth and the throat ($P_1 - P_2$). Only one step remains to achieve a volumetric flow equation here, and that is to convert velocity ($v$) into flow rate ($Q$). Velocity is expressed in units of length per time (feet or meters per second or minute), while volumetric flow is expressed in units of volume per time (cubic feet or cubic meters per second or minute). Simply multiplying throat velocity ($v_2$) by throat area ($A_2$) will give the desired result:

General flow/area/velocity relationship:

$$Q = Av$$

Equation for throat velocity:

$$v_2 = \sqrt{2}\frac{1}{\sqrt{1-\left(\frac{A_2}{A_1}\right)^2}}\sqrt{\frac{P_1 - P_2}{\rho}}$$

Multiplying both sides of the equation by throat area:

$$A_2 v_2 = \sqrt{2}A_2\frac{1}{\sqrt{1-\left(\frac{A_2}{A_1}\right)^2}}\sqrt{\frac{P_1 - P_2}{\rho}}$$

An equation solving for volumetric flow in terms of pressures and areas is the result:

$$Q = \sqrt{2}A_2\frac{1}{\sqrt{1-\left(\frac{A_2}{A_1}\right)^2}}\sqrt{\frac{P_1 - P_2}{\rho}}$$

Please note how many constants are in this equation. For any given venturi tube, the mouth and throat areas ($A_1$ and $A_2$) will be fixed. This means the majority of this rather long equation are constant for any particular venturi tube, and therefore do not change with pressure, density, or flow rate. Knowing this, the equation may be rewritten as a simple proportionality:

$$Q \propto \sqrt{\frac{P_1 - P_2}{\rho}}$$

To make this a more precise mathematical statement, a *constant of proportionality* ($k$) may be inserted:

$$Q = k\sqrt{\frac{P_1 - P_2}{\rho}}$$

The value of $k$ depends, of course, on the physical dimensions of the venturi mouth and throat. A practical advantage to using a constant of proportionality is that the value of $k$ may be adjusted as necessary to account for factors other than just venturi tube geometry. One very important factor to consider is units of measurement. If the value of $k$ is determined strictly by tube geometry, then the units used to express volumetric flow rate *must* correspond to the units used to express pressures and fluid density. For example, $Q$ will be in units of cubic feet per second only when pressure values $P_1$ and $P_2$ are in units of pounds per square *foot* and mass density in units of *slugs* per cubic feet! When using more convenient units of measurement such as inches of water column for pressure and specific gravity (unitless) for density, the volumetric flow value produced by the raw equation will not be in any useful unit.

However, if the differential pressure produced by any particular venturi tube with any particular fluid density at a specified flow rate is known, the $k$ value necessary to characterize that venturi tube for any other condition using those units may be calculated. For example, if a particular venturi tube develops 45 inches of water column differential pressure at a flow rate of 180 gallons per minute of water (specific gravity = 1), these values may be substituted and the equation solved for $k$:

$$Q = k\sqrt{\frac{P_1 - P_2}{\rho}}$$

$$180 = k\sqrt{\frac{45}{1}}$$

$$k = \frac{180}{\sqrt{\frac{45}{1}}} = 26.83$$

Now that it is known a value of 26.83 for $k$ will yield gallons per minute of liquid flow through this venturi tube given pressure in inches of water column and density as a specific gravity, the flow rate through this tube for any other pressure drop may be readily predicted:

$$\left[\frac{\text{gal}}{\text{m}}\right] = 26.83\sqrt{\left[\frac{\text{W.C.}}{\text{Specific gravity}}\right]}$$

60 inches of water column differential pressure generated by a flow of water (specific gravity = 1) in this particular venturi tube gives the following flow rate:

$$Q = 26.83\sqrt{\frac{60}{1}}$$

$$Q = 207.8 \text{ GPM}$$

110 inches of water column differential pressure generated by a flow of gasoline (specific gravity = 0.657) in this same venturi tube gives the following flow rate:

$$Q = 26.83\sqrt{\frac{110}{0.657}}$$

$$Q = 347 \text{ GPM}$$

To calculate *mass* flow instead of volumetric flow, the equation does not change much. The relationship between volume ($V$) and mass ($m$) for a sample of fluid is its mass density ($\rho$):

$$\rho = \frac{m}{V}$$

Similarly, the relationship between a volumetric *flow rate* ($Q$) and a mass *flow rate* ($W$) is also the fluid's mass density ($\rho$):

$$\rho = \frac{W}{Q}$$

Solving for $W$ in this equation leads to a product of volumetric flow rate and mass density:

$$W = \rho Q$$

A quick dimensional analysis check using common metric units confirms this fact. A mass flow rate in kilograms per second will be obtained by multiplying a mass density in kilograms per cubic meter by a volumetric flow rate in cubic meters per second:

$$\left[\frac{\text{kg}}{\text{s}}\right] = \left[\frac{\text{kg}}{\text{m}^3}\right]\left[\frac{\text{m}^3}{\text{s}}\right]$$

Therefore, to turn the general volumetric flow equation into a mass flow equation, multiply both sides by fluid density ($\rho$):

$$Q = k\sqrt{\frac{P_1 - P_2}{\rho}}$$

$$\rho Q = k\rho\sqrt{\frac{P_1 - P_2}{\rho}}$$

$$W = k\rho\sqrt{\frac{P_1 - P_2}{\rho}}$$

It is generally considered inelegant to show the same variable more than once in an equation if it is not necessary, so the two densities ($\rho$) are consolidated using algebra. First, $\rho$ is written as the product of two square-roots:

$$W = k\sqrt{\rho}\sqrt{\rho}\sqrt{\frac{P_1 - P_2}{\rho}}$$

Next, the last radical is broken up into a quotient of two separate square roots:

$$W = k\sqrt{\rho}\sqrt{\rho}\sqrt{\frac{P_1 - P_2}{\sqrt{\rho}}}$$

Now one of the square-rooted $\rho$ terms cancels out the one in the denominator of the fraction:

$$W = k\sqrt{\rho}\sqrt{P_1 - P_2}$$

Rewriting the two roots as one:

$$W = k\sqrt{\rho(P_1 - P_2)}$$

As with the volumetric flow equation, to arrive at a suitable $k$ value for any particular venturi tube, a set of values taken from a real venturi tube in service is needed, expressed in whatever units of measurement desired.

For example, if a venturi tube generates a differential pressure of 2.30 kilo-Pascals (kPa) at a mass flow rate of 500 kilograms per minute of naphtha (a petroleum product having a density of 0.665 kilograms per liter), the $k$ value of this venturi tube may be determined as such:

$$W = k\sqrt{\rho(P_1 - P_2)}$$

$$500 = k\sqrt{(0.665)(2.3)}$$

$$k = \frac{500}{\sqrt{(0.665)(2.3)}}$$

$$k = 404$$

Now that it is known that a value of 404 for $k$ will yield kilograms per minute of liquid flow through this venturi tube given pressure in kPa and density in kilograms per liter, the mass flow rate through this tube for any other pressure drop measured may be readily predicted:

$$\left[\frac{\text{kg}}{\text{m}}\right] = 404\sqrt{\left[\frac{\text{kg}}{\text{l}}\right]\,[\text{kPa}]}$$

6.1 kPa of differential pressure generated by a flow of sea water (density = 1.03 kilograms per liter) in this particular venturi tube gives the following mass flow rate:

$$W = 404\sqrt{(1.03)(6.1)}$$

$$W = 1012\,\frac{\text{kg}}{\text{m}}$$

It should be apparent by now that the relationship between flow rate (whether it be volumetric or mass) and differential pressure is non-linear: a doubling of flow rate will *not* result in a doubling of differential pressure. Rather, a doubling of flow rate will result in a *quadrupling* of differential pressure.

This quadratic relationship between flow and pressure drop due to fluid acceleration requires mathematical "conditioning" or "characterizing" of the pressure signal sensed by the differential pressure instrument in order to arrive at an expressed value for flow rate. The customary solution to this problem is to incorporate a "square root" function between the transmitter and the flow indicator, as shown in the following diagram:

Figure 13.8: Diagram of a venturi tube in operation with characterized transmitter output

In the days of pneumatic instrumentation, this square-root function was performed in a separate device called a *square root extractor*. The Foxboro Corporation Model 557 pneumatic square root extractor was a classic example of this technology[4]:

[4]Despite the impressive craftsmanship and engineering that went into the design of pneumatic square root extractors, their obsolescence is mourned by no one. These devices were notoriously difficult to set up and calibrate accurately, especially as they aged.

Figure 13.9: An industrial pneumatic square root extractor

The modern solution is to incorporate digital square-root computation and gain characterization either in the indicator or in the transmitter itself.

### 13.1.2 Orifice plates

Of all the pressure-based flow elements in existence, the most common is the *orifice plate*. This is simply a metal plate with a hole in the middle for fluid to flow through. Orifice plates are typically sandwiched between two flanges of a pipe joint, allowing for easy installation and removal:

Figure 13.10: Diagram of an installed orifice plate

The point where the fluid flow profile constricts to a minimum cross-sectional area after flowing through the orifice is called the *vena contracta*, and it is the area of minimum fluid pressure. The vena contracta corresponds to the narrow throat of a venturi tube. The precise location of the vena contracta for an orifice plate installation will vary with flow rate, and also with the *beta ratio* ($\beta$) of the orifice plate, defined as the ratio of bore diameter ($d$) to inside pipe diameter ($D$):

$$\beta = \frac{d}{D}$$

The simplest design of orifice plate is the *square-edged, concentric* orifice. This type of orifice plate is manufactured by machining a precise, straight hole in the middle of a thin metal plate. Looking at a side view of a square-edged concentric orifice plate reveals sharp edges ($90^o$ corners) at the hole:

Figure 13.11: Diagram of a square-edged concentric orifice plate

Square-edged orifice plates may be installed in either direction, since the orifice plate "appears" exactly the same from either direction of fluid approach. In fact, this allows square-edged orifice plates to be used for measuring bidirectional flow rates (where the fluid flow direction reverses itself from time to time). A text label printed on the "paddle" of any orifice plate customarily identifies the upstream side of that plate, but in the case of the square-edged orifice plate it does not matter.

The purpose of having a square edge on the hole in an orifice plate is to minimize contact with the fast-moving moving fluid stream going through the hole. Ideally, this edge will be knife-sharp. If the orifice plate is relatively thick (1/8 or an inch or more), it may be necessary to bevel the downstream side of the hole to further minimize contact with the fluid stream:

Figure 13.12: Diagram of a beveled concentric orifice plate

Looking at the side-view of this orifice plate, the intended direction of flow is left-to-right, with the sharp edge facing the incoming fluid stream and the bevel providing a non-contact outlet for the fluid. Beveled orifice plates are obviously uni-directional, and *must* be installed with the paddle text facing upstream.

Other square-edged orifice plates exist to address conditions where gas bubbles or solid particles may be present in liquid flows, or where liquid droplets or solid particles may be present in gas flows. The first of this type is called the *eccentric* orifice plate, where the hole is located off-center to allow the undesired portions of the fluid to pass through the orifice rather than build up on the upstream face:

Figure 13.13: Diagram of a square-edged eccentric orifice plate

For gas flows, the hole should be offset downward, so any liquid droplets or solid particles may easily pass through. For liquid flows, the hole should be offset upward to allow gas bubbles to pass through and offset downward to allow heavy solids to pass through.

The second off-center orifice plate type is called the *segmental orifice plate*, where the hole is not circular but rather just a segment of a concentric circle:

Figure 13.14: Diagram of a segmental orifice plate

As with the eccentric orifice plate design, the segmental hole should be offset downward in gas flow applications and either upward or downward in liquid flow applications depending on the type of undesired material(s) in the flowstream.

An alternative to offsetting or re-shaping the bore hole of an orifice plate is to simply drill a small hole near the edge of the plate, flush with the inside diameter of the pipe, allowing undesired substances to pass through the plate rather than collect on the upstream side. If such a hole is oriented upward to pass vapor bubbles, it is called a *vent hole*. If the hole is oriented downward to pass liquid droplets or solids, it is called a *drain hole*. Vent and drain holes are useful when the concentration of these undesirable substances is not significant enough to warrant either an eccentric or segmental orifice:

Figure 13.15: Diagram of an orifice plate showing vent and drain holes

The addition of a vent or drain hole should have negligible impact on the performance of an orifice plate due to its small size relative to the main bore (it's typically less than 10% of the bore hole). If the quantity of undesirable material in the flowstream (bubbles, droplets, or solids) is excessive, an eccentric or segmental orifice plate might be a better choice[5].

Some orifice plates employ non-square-edged holes for the purpose of improving performance at low Reynolds number values, where the effects of fluid viscosity are more apparent. These orifice plate types employ rounded- or conical-entrance holes in an effort to minimize the effects of fluid viscosity. Experiments have shown that decreased Reynolds number causes the flowstream to not contract as

[5] L.K. Spink, in his book *Principles and Practice of Flow Meter Engineering*, notes that drain holes intended to pass solid objects may be useless in small pipe sizes, where the hole is so small it will probably become plugged with solid debris and cease to provide benefit. In such installations he recommends reorienting the pipe vertically instead of horizontally. This allows solids to pass through the main bore of the orifice without "damming" on the upstream side of the orifice plate. Alternatively, consider a different primary element entirely, such as a venturi tube. The small size of the line will limit the cost of such an element, and the performance is likely to be far better than an orifice plate anyway.

much when traveling through an orifice, thus limiting fluid acceleration and decreasing the amount of differential pressure produced by the orifice plate. However, experiments have also shown that decreased Reynolds number in a venturi-type flow element causes an *increase* in differential pressure due to the effects of friction against the entrance cone walls. By manufacturing an orifice plate in such a way that the hole exhibits "venturi-like" properties (i.e. a dull edge where the fast-moving fluid stream has more contact with the plate), these two effects tend to cancel each other, resulting in an orifice plate that maintains consistent accuracy at lower flow rates and/or higher viscosities than the simple square-edged orifice.

Two common non-square-edge orifice plate designs are the *quadrant-edge* and *conic-entrance* orifices. The quadrant-edge is shown first:

Figure 13.16: Diagram of a quadrant-edged concentric orifice plate

The conical-entrance orifice plate looks like a beveled square-edge orifice plate installed backwards, with flow entering the conical side and exiting the square-edged side:

Figure 13.17: Diagram of a contical entrance concentric orifice plate

Here, is it vitally important to pay attention to the paddle's text label. This is the only sure indication of which direction an orifice plate needs to be installed. One can easily imagine an instrument technician mistaking a conical-entrance orifice plate for a square-edged, beveled orifice plate and installing it backward!

Several standards exist for pressure tap locations. Ideally, the upstream pressure tap will detect fluid pressure at a point of minimum velocity, and the downstream tap will detect pressure at the vena contracta (maximum velocity). In reality, this ideal is never perfectly achieved. An overview of the most popular tap locations for orifice plates is shown in the following illustration:

Figure 13.18: Diagram showing typical tap locations for differential pressure transmitters

*Flange taps* are the most popular tap location for orifice meter runs on large pipes in the United States. Flanges may be manufactured with tap holes pre-drilled and finished before the flange is even welded to the pipe, making this a very convenient pressure tap configuration. Most of the other tap configurations require drilling into the pipe after installation, which is not only labor-intensive, but may possibly weaken the pipe at the locations of the tap holes.

*Vena contracta taps* offer the greatest differential pressure for any given flow rate, but require precise calculations to properly locate the downstream tap position. *Radius taps* are an approximation of vena contracta taps for large pipe sizes (one-half pipe diameter downstream for

the low-pressure tap location). An unfortunate characteristic of both these taps is the requirement of drilling through the pipe wall. Not only does this weaken the pipe, but the practical necessity of drilling the tap holes in the installed location rather than in a controlled manufacturing environment means there is considerable room for installation error[6].

*Corner taps* must be used on small pipe diameters where the vena contracta is so close to the downstream face of the orifice plate that a downstream flange tap would sense pressure in the highly turbulent region (too far downstream). Corner taps often require special (i.e., expensive) flange fittings, which is why they tend to be used only when necessary. However, at least one manufacturer builds an orifice plate with the orifice and corner taps machined into a solid body that fits between standard flanges.

Care should be taken to avoid measuring downstream pressure in the highly turbulent region following the vena contracta. This is why the *pipe tap* (also known as *full-flow tap*) standard calls for a downstream tap location eight pipe diameters away from the orifice: to give the flow stream room to stabilize for more consistent pressure readings[7].

Wherever the taps are located, it is vitally important that the tap holes be completely flush with the inside wall of the pipe or flange. Even the smallest recess or burr left from drilling will cause measurement errors, which is why tap holes are best drilled in a controlled manufacturing environment rather that at the installation site where the task will likely be performed by non-experts.

For relatively low flow rates, an alternative arrangement is the *integral orifice plate*. This is where a small orifice plate directly attaches to the differential pressure-sensing element, eliminating the need for impulse lines. A photograph of an integral orifice plate and transmitter is shown here:

[6] One significant source of error for customer-drilled tap holes is the interior finish of the holes. Even a small "burr" of metal left where the hole penetrates the inner surface of the pipe wall will cause substantial flow measurement errors!

[7] What this means is that a "pipe tap" installation is actually measuring permanent pressure loss, which also happens to scale with the square of flow rate because the primary mechanism for energy loss in turbulent flow conditions is the translation of linear velocity to angular (swirling) velocity in the form of eddies. This kinetic energy is eventually dissipated in the form of heat as the eddies eventually succumb to viscosity.

Figure 13.19: An industrial orifice plate with integral transmitter

The task of properly sizing an orifice plate for any given application is complex enough to recommend the use of special orifice sizing computer software provided by orifice plate manufacturers. There are a number of factors to consider in orifice plate sizing, and these software packages account for all of them. Best of all, the software provided by manufacturers is often linked to data for that manufacturer's product line, helping to assure installed results in close agreement with predictions.

In the days before ubiquitous personal computers and the internet, some orifice plate manufacturers provided customers with paper "slide rule" calculators to help them select appropriate orifice plate sizes from known process parameters. The following photographs show the front and back sides of one such slide rule:

Figure 13.20: The front of an orifice flow slide rule calculator

Figure 13.21: The reverse of an orifice flow slide rule calculator

### 13.1.3 Other differential producers

Other pressure-based flow elements exist as alternatives to the orifice plate. The *Pitot tube*, for example, senses pressure as the fluid stagnates (comes to a complete stop) against the open end of a forward-facing tube. A shortcoming of the classic single-tube Pitot assembly is sensitivity to fluid velocity at just one point in the pipe, so a more common form of Pitot tube seen in industry is the *averaging* Pitot tube consisting of several stagnation holes sensing velocity at multiple points across the width of the flow:

Figure 13.22: Diagram of single point and averaging Pitot tubes

A variation on the latter theme is the *Annubar* flow element, a trade name of the Emerson Process Management Rosemount division. An "Annubar" is an averaging pitot tube consolidating high and low pressure-sensing ports in a single probe assembly:

Figure 13.23: Diagram of an Annubar

Figure 13.24: An Annubar flow element

What appears at first glance to be a single, square-shaped tube inserted into the pipe is actually a double-ported tube with holes on both the upstream and downstream edges:

Figure 13.25: Closeup of an Annubar flow element

A section of Annubar tube clearly shows the porting and dual chambers, designed to bring upstream (stagnation) and downstream pressures out of the pipe to a differential pressure-sensing instrument:

Figure 13.26: Cross section of an Annubar flow element

A less sophisticated realization of the stagnation principle is the *target* flow sensor, consisting of a blunt "paddle" (or "drag disk") inserted into the flowstream. The force exerted on this paddle by the moving fluid is sensed by a special transmitter mechanism, which then outputs a signal corresponding to flow rate (proportional to the square of fluid velocity, just like an orifice plate):

Figure 13.27: A target flow element

The classic venturi tube pioneered by Clemens Herschel in 1887 has been adapted in a variety of forms broadly classified as *flow tubes*. All flow tubes work on the same principle: developing a differential pressure by channeling fluid flow from a wide tube to a narrow tube. The differ from the classic venturi only in construction details, the most significant detail being a significantly shorter length than the classic venturi tube. Examples of flow tube designs include the *Dall* tube, *Lo-Loss* flow tube, *Gentile* or *Bethlehem* flow tube, and the *B.I.F. Universal Venturi*.

Another variation on the venturi theme is called a *flow nozzle*, designed to be clamped between the faces of two pipe flanges in a manner similar to an orifice plate. The goal here is to achieve simplicity of installation approximating that of an orifice plate while improving performance (less permanent pressure loss) over orifice plates:

Figure 13.28: A flow nozzle with transmitter

Two more variations on the venturi theme are the *V-cone* and *Segmental wedge* flow elements. The V-cone (or "venturi cone," a trade name of the McCrometer division of the Danaher corporation) may be thought of as a venturi tube in reverse: instead of narrowing the tube's diameter to cause fluid acceleration, fluid must flow around a cone-shaped obstruction placed in the middle of the tube. The tube's effective area will be reduced by the presence of this cone, causing fluid to accelerate through the restriction just as it would through the throat of a classic venturi tube:

Figure 13.29: A V-cone flow element with transmitter

This cone is hollow, with a pressure-sensing port on the downstream side allowing for easy detection of fluid pressure near the vena contracta. Upstream pressure is sensed by another port in the pipe wall upstream of the cone. The following photograph shows a V-cone flow tube, cut away for demonstration purposes:

Figure 13.30: Cutaway view of a V-cone flow element

Segmental wedge elements are special pipe sections with wedge-shaped restrictions built in. These devices, albeit crude, are useful for measuring the flow rates of *slurries*, especially when pressure is sensed by the transmitter through remote-seal diaphragms (to eliminate the possibility of impulse tube plugging):

Figure 13.31: A segmental wedge flow element with transmitter

Finally, the lowly pipe elbow may be pressed into service as a flow-measuring element, since fluid turning a corner in the elbow experiences radial acceleration and therefore generates a differential pressure along the axis of acceleration:

Figure 13.32: A pipe elbow used as a flow element with transmitter

Pipe elbows should be considered for flow measurement only as a last resort. Their inaccuracies tend to be extreme, owing to the non-precise construction of most pipe elbows and the relatively weak differential pressures generated[8].

[8]The fact that a pipe elbow generates small differential pressure is an accuracy concern because other sources of pressure become larger by comparison. Noise generated by fluid turbulence in the elbow, for example, becomes a significant portion of the pressure sensed by the transmitter when the differential pressure is so low (i.e. the signal-to-noise ratio becomes smaller). Errors caused by differences in elbow tap elevation and different impulse line fill fluids, for example, become more significant as well.

A final point should be mentioned on the subject of differential-producing elements, and that is their energy dissipation. Orifice plates are simple and relatively inexpensive to install, but their permanent pressure loss is high compared with other primary elements such as venturi tubes. Permanent pressure loss is permanent energy loss from the flowstream, which usually represents a loss in energy invested into the process by pumps, compressors, and/or blowers. Fluid energy dissipated by an orifice plate thus (usually) translates into a requirement of greater energy input to that process[9].

With the financial and ecological costs of energy being non-trivial in the modern world, it is important to consider energy loss as a significant factor in choosing the appropriate primary element for a pressure-based flowmeter. It might very well be that an "expensive" venturi tube saves more money in the long term than a "cheap" orifice plate, while delivering greater measurement accuracy as an added benefit.

### 13.1.4 Proper installation

Perhaps the most common way in which the flow measurement accuracy of any flowmeter becomes compromised is incorrect installation, and pressure-based flowmeters are no exception to this rule. The following list shows some of the details one must consider in installing a pressure-based flowmeter element:

- Necessary upstream and downstream straight-pipe lengths
- Beta ratio (ratio of orifice bore diameter to pipe diameter: $\beta = \frac{d}{D}$)
- Impulse tube tap locations
- Tap finish
- Transmitter location in relation to the pipe

Sharp turns in piping networks introduce large-scale turbulence[10] into the flowstream. Elbows, tees, valves, fans, and pumps are some of the most common causes of large-scale turbulence in piping systems. Successive pipe elbows in different planes are some of the worst offenders in this regard. When the natural flow path of a fluid is disturbed by such piping arrangements, the velocity profile of that fluid will become asymmetrical; e.g., the velocity gradient from one wall boundary of the pipe to the other will not be orderly. Large eddies in the flowstream (called *swirl*) will be present. This may cause problems for pressure-based flow elements which rely on linear acceleration (change in velocity in one dimension) to measure fluid flow rate. If the flow profile is distorted enough, the acceleration detected at the element may be too great or too little, and therefore not properly represent the full fluid flowstream[11].

[9] This is not always the case, as primary elements are often found on throttled process lines. In such cases where a control valve normally throttles the flow rate, any energy dissipated by the orifice plate is simply less energy that the valve would otherwise be required to dissipate. Therefore, the presence or absence of an orifice plate has no net impact on energy dissipation when used on a process flow throttled by a control valve.

[10] This is not to be confused with micro-turbulence in the fluid, which cannot be eliminated at high Reynolds number values. In fact, "fully-developed turbulent flow" is desirable for head-based meter elements such as orifice plates because it means the flow profile will be relatively flat (even velocities across the pipe's diameter) and frictional forces (viscosity) will be negligible. Avoid *large-scale* turbulent effects such as eddies, swirl, and asymmetrical flow profiles, which compromise the ability of most flowmeters to accurate measure flow rate.

[11] L.K. Spink mentions in his book *Principles and Practice of Flow Meter Engineering* that certain tests have shown flow measurement errors induced from severe disturbances as far as 60 to 100 pipe diameters upstream of the primary flow element!

Figure 13.33: Illustration of disturbances in the flow streamlines due to a bend

Even disturbances located *downstream* of the flow element impact measurement accuracy (albeit not as much as upstream disturbances). Unfortunately, both upstream and downstream flow disturbances are unavoidable on all but the simplest fluid systems. This means ways to stabilize a flowstream's velocity profile near the flow element must be devised in order to achieve accurate measurements of flow rate. A very simple and effective way to stabilize a flow profile is to provide adequate lengths of straight pipe ahead of (and behind) the flow element. Given enough time, even the most chaotic flowstream will "settle down" to a symmetrical profile all on its own. The following illustration shows the effect of a pipe elbow on a flowstream, and how the velocity profile returns to a normal (symmetrical) form after traveling through a sufficient length of straight pipe:

Figure 13.34: Disturbances in the flow streamlines dissipate along straight pipe

Recommendations for minimum upstream and downstream straight-pipe lengths vary significantly with the nature of the turbulent disturbance, piping geometry, and flow element. As a general rule, elements having a smaller beta ratio (ratio of throat diameter $d$ to pipe diameter $D$) are more tolerant of disturbances, with profiled flow devices (e.g., venturi tubes, flow tubes, V-cones) having the greatest tolerance[12]. Ultimately, consult the flow element manufacturer's documentation for a more detailed recommendation appropriate to any specific application.

In applications where sufficient straight-run pipe lengths are impractical, another option exists for "taming" turbulence generated by piping disturbances. Devices called *flow conditioners* may be installed upstream of the flow element to help the flow profile achieve symmetry in a far shorter distance than simple straight pipe could do alone. Flow conditioners take the form of a series of tubes or vanes installed inside the pipe, parallel to the direction of flow. These tubes or vanes force

[12] However, there are disadvantages to using low-beta elements, one of them being increased permanent pressure loss which usually translates to increased operating costs due to energy loss.

the fluid molecules to travel in straighter paths, thus stabilizing the flowstream prior to entering a flow element:

Figure 13.35: Use of a flow conditioner to straighten flow streamlines

Another common source of trouble for pressure-based flowmeters is improper transmitter location. Here, the type of process fluid flow being measured dictates how the pressure-sensing instrument should be located in relation to the pipe. For gas and vapor flows, it is important that no stray liquid droplets collect in the impulse lines leading to the transmitter, lest a vertical liquid column begin to collect in those lines and generate an error-producing pressure. For liquid flows, it is important that no gas bubbles collect in the impulse lines, or else those bubbles may displace liquid from the lines and thereby cause unequal vertical liquid columns, which would (again) generate an error-producing differential pressure.

In order to let gravity do the work of preventing these problems, locate the transmitter *above* the pipe for gas flow applications and *below* the pipe for liquid flow applications:

Figure 13.36: Mounting positions for dp-style flow transmitters

Figure 13.37: Use of a flow conditioner to straighten flow streamlines

The impulse lines are typically connected to the sides of the pipe, not the top or bottom as shown in the figure. This gives the best opportunity to avoid collecting gas, condensation, or sediment in the lines. Also, the impulse lines should slope at a minimum of 1/4" per foot of horizontal run toward the pipe for gas flow transmitters to allow condensation to drain toward the pipe, and away from the pipe for liquid flow transmitters to keep the impulse lines filled.

Condensible vapor applications (such as steam flow measurement) should be treated the same as liquid measurement applications. Here, condensed liquid will collect in the transmitter's impulse lines so long as the impulse lines are cooler than the vapor flowing through the pipe (which is typically the case). Placing the transmitter below the pipe allows vapors to condense and fill the impulse lines with liquid (condensate), which then acts as a natural seal protecting the transmitter from exposure to hot process vapors. In such applications it is important to pre-fill both impulse lines with condensed liquid prior to placing the flowmeter into service. Tee fittings with removable plugs or fill valves are provided to do this. Failure to pre-fill the impulse lines will likely result in measurement errors during initial operation, as condensed vapors will inevitably fill the impulse lines at slightly different rates and cause a difference in vertical liquid column heights within those lines.

If tap holes must be drilled into the pipe (or flanges) at the process site, great care must be taken to properly drill and de-burr the holes. A pressure-sensing tap hole should be flush with the inner pipe wall, with no rough edges or burrs to create turbulence. Also, there should be no reliefs or countersinking near the hole on the inside of the pipe. Even small irregularities at the tap holes may generate surprisingly large flow-measurement errors.

### 13.1.5 High-accuracy flow measurement

Many assumptions were made in formulating flow equations from physical conservation laws. Suffice it to say, the flow formulae so far in this chapter are only approximations of reality. Orifice plates are some of the worst offenders in this regard, since the fluid encounters such abrupt changes in geometry passing through the orifice. Venturi tubes are nearly ideal, since the machined contours of the tube ensure gradual changes in fluid pressure and minimize turbulence.

However, in the real world imperfect technologies are often used. Orifice plates, despite being less than perfect as flow-sensing elements, are convenient and economical to install in flanged pipes. Orifice plates are also the easiest type of flow element to replace in the event of damage or routine servicing. In applications such as custody transfer, where the flow of fluid represents product being bought and sold, flow measurement accuracy is paramount. It is therefore important to figure out how to coax the most accuracy from the common orifice plate in order to measure fluid flows both accurately and economically.

If the true flow rate through a pressure-generating primary sensing element is compared to the theoretical flow rate predicted by an idealized equation, a substantial discrepancy is noted[13]. Causes of this discrepancy include, but are not limited to:

- Energy losses due to turbulence and viscosity
- Energy losses due to friction against the pipe and element surfaces
- Unstable location of *vena contracta* with changes in flow
- Uneven velocity profiles caused by irregularities in the pipe
- Fluid compressibility
- Thermal expansion (or contraction) of the element and piping
- Non-ideal pressure tap location(s)
- Excessive turbulence caused by rough internal pipe surfaces

The ratio between true flow rate and theoretical flow rate for any measured amount of differential pressure is known as the *discharge coefficient* of the flow-sensing element, symbolized by the variable $C$. Since a value of 1 represents a theoretical ideal, the actual value of $C$ for any real pressure-generating flow element will be less than 1:

$$C = \frac{\text{True flow}}{\text{Theoretical flow}}$$

For gas and vapor flows, true flow rate deviates even more from the theoretical (ideal) flow value than liquids do, for reasons that have to do with the compressible nature of gases and vapors. A *gas expansion factor* ($Y$) may be calculated for any flow element by comparing its discharge coefficient for gases against its discharge coefficient for liquids. As with the discharge coefficient, values of $Y$ for any real pressure-generating element will be less than 1:

$$Y = \frac{C_{gas}}{C_{liquid}}$$

[13] Richard W. Miller, in his outstanding book *Flow Measurement Engineering Handbook*, states that venturi tubes may come within 1 to 3 percent of ideal, while a square-edged orifice plate may perform as poorly as only 60 percent of theoretical!

$$Y = \frac{\left(\frac{\text{True gas flow}}{\text{Theoretical gas flow}}\right)}{\left(\frac{\text{True liquid flow}}{\text{Theoretical liquid flow}}\right)}$$

Incorporate these factors into the ideal volumetric flow equation developed on page 491 to arrive at the following formulation:

$$Q = \sqrt{2}\frac{CYA_2}{\sqrt{1 - \left(\frac{A_2}{A_1}\right)^2}}\frac{\sqrt{P_1 - P_2}}{\rho}$$

If desired, another factor could be added to account for any necessary unit conversions ($N$), getting rid of the constant $\sqrt{2}$ in the process:

$$Q = N\frac{CYA_2}{\sqrt{1 - \left(\frac{A_2}{A_1}\right)^2}}\sqrt{\frac{P_1 - P_2}{\rho}}$$

Sadly, neither the discharge coefficient ($C$) nor the gas expansion factor ($Y$) will remain constant across the entire measurement range of any given flow element. These variables are subject to some change with flow rate, which further complicates the task of accurately inferring flow rate from differential pressure measurement. However, if the values of $C$ and $Y$ for typical flow conditions are known, good accuracy may be achieved most of the time.

Likewise, the fact that $C$ and $Y$ change with flow places limits on the accuracy obtainable with the "proportionality constant" formulae seen earlier. Whether measuring volumetric or mass flow rate, the $k$ factor calculated at one particular flow condition will not hold constant for *all* flow conditions:

$$Q = k\sqrt{\frac{P_1 - P_2}{\rho}}$$

$$W = k\sqrt{\rho(P_1 - P_2)}$$

This means after calculating a value for $k$ based on a particular flow condition, the results of the equation can only be trusted for flow conditions not too different from the one used to calculate $k$.

As can be seen in both flow equations, the density of the fluid ($\rho$) is an important factor. If fluid density is relatively stable, $\rho$ may be treated as a constant, incorporating its value into the proportionality factor ($k$) to make the two formulae even simpler:

$$Q = k_Q\sqrt{P_1 - P_2}$$

$$W = k_W\sqrt{P_1 - P_2}$$

However, if fluid density is subject to change over time, some means to continually calculate $\rho$ will be needed so the inferred flow measurement will remain accurate. Variable fluid density is a typical state of affairs in gas flow measurement, since all gases are compressible by definition. A

simple change in static gas pressure within the pipe is all that is needed to make $\rho$ change, which in turn affects the relationship between flow rate and differential pressure drop.

The American Gas Association (AGA) provides a formula for calculating volumetric flow of any gas using orifice plates in their #3 Report, compensating for changes in gas pressure and temperature. A variation of that formula is shown here (consistent with previous forms in this section):

$$Q = N \frac{CYA_2}{\sqrt{1 - \left(\frac{A_2}{A_1}\right)^2}} \sqrt{\frac{Z_s P_1 (P_1 - P_2)}{G_f Z_{f1} T}}$$

Where,

$Q$ = Volumetric flow rate (e.g., gallons per minute, standard cubic feet per second)

$N$ = Unit conversion factor

$C$ = Discharge coefficient (accounts for energy losses, Reynolds number corrections, pressure tap locations, etc.)

$A_1$ = Cross-sectional area of mouth

$A_2$ = Cross-sectional area of throat

$Z_s$ = Compressibility factor of gas under standard conditions

$Z_f 1$ = Compressibility factor of gas under flowing conditions, upstream

$G_f$ = Specific gravity of gas

$T$ = Absolute temperature of gas

$P_1$ = Upstream pressure (absolute)

$P_2$ = Downstream pressure (absolute)

This equation implies the continuous measurement of gas pressure ($P_1$) and temperature ($T$) inside the pipe, in addition to the differential pressure produced by the orifice plate ($P_1 - P_2$). These measurements may be taken by three separate devices, their signals routed to a gas flow computer:

Figure 13.38: Use of a flow computer with an orifice plate for high-accuracy measurement

Note the location of the RTD (thermowell), positioned downstream of the orifice plate so the turbulence it generates will have negligible impact on the fluid dynamics at the orifice plate. The American Gas Association (AGA) allows for upstream placement of the thermowell, but only if located at least three feet upstream of a flow conditioner[14].

In order to best control all the physical parameters necessary for good orifice metering accuracy, it is standard practice for custody transfer flowmeter installations to use *honed meter runs* rather than standard pipe and pipe fittings. A "honed run" is a complete piping assembly consisting of a manufactured fitting to hold the orifice plate and sufficient straight lengths of pipe upstream and downstream, the interior surfaces of that pipe machined ("honed") to have a glass-smooth surface with precise and symmetrical dimensions. Such piping "runs" are quite expensive, but necessary if flow measurement accuracy worthy of custody transfer is to be achieved.

Figure 13.39 shows a set of AGA-3-compliant orifice meter runs measuring the flow of natural gas:

[14]Specified in Part 2 of the AGA Report #3, section 2.6.5, page 22.

Figure 13.39: Natural gas flow instrument meter runs and manifolds

Note the special transmitter manifolds, built to accept both the differential pressure and absolute pressure (Rosemount model 3051) transmitters. Also note the quick-change fittings (the ribbed cast-iron housings) holding the orifice plates, to facilitate convenient change-out of the orifice plates which is periodically necessary due to wear. It is not unheard of to replace orifice plates on a daily basis in some industries to ensure the sharp orifice edges necessary for accurate measurement[15].

Although not visible in this photograph, these meter runs are connected together by a network of shut-off valves directing the flow of natural gas through as few meter runs as desired. When the total gas flow rate is great, all meter runs are placed into service and their respective flow rates summed to yield a total flow measurement. When the total flow rate decreases, individual meter runs are shut off, resulting in increased flow rates through the remaining meter runs. This "staging" of meter runs expands the effective *rangeability* of the orifice plate as a flow-sensing element, resulting in much more accurate flow measurement over a wide range of flow rates than if a single (large) orifice meter run were used.

An alternative to multiple instruments (differential pressure, absolute pressure, and temperature) installed on each meter run is to use a single *multi-variable* transmitter capable of measuring gas temperature as well as both static and differential pressures. This approach enjoys the advantage of simpler installation over the multi-instrument approach:

[15]This is especially true in the gas exploration industry, where natural gas coming out of the earth is laden with a substantial amount of sand, rocks, and grit.

Figure 13.40: A multivariable transmitter may be used for compensated flow measurement

The Rosemount model 3095MV and Yokogawa model EJX910 are examples of multi-variable transmitters designed to perform compensated gas flow measurement, equipped with multiple pressure sensors, a connection port for an RTD temperature sensor, and sufficient digital computing power to continuously calculate flow rate based on the AGA equation. Such multi-variable transmitters may provide an analog output for computed flow rate, or a digital output where all three primary variables *and* the computed flow rate may be transmitted to a host system (as shown in the previous illustration). The Yokogawa EJX910A provides an interesting signal output option: a digital *pulse* signal, where each pulse represents a specific quantity (either volume or mass) of fluid. The frequency of this pulse train represents flow rate, while the total number of pulses counted over a period of time represents the total amount of fluid that has passed through the orifice plate over that amount of time.

Liquid flow measurement applications may also benefit from compensation, because liquid density changes with temperature. Static pressure is not a concern here, because liquids are considered incompressible for all practical purposes[16]. Thus, the formula for compensated liquid flow measurement does not include any terms for static pressure, just differential pressure and temperature:

$$Q = N \frac{CYA_2}{\sqrt{1 - \left(\frac{A_2}{A_1}\right)^2}} \sqrt{(P_1 - P_2)[1 + k_T(T - T_{ref})]}$$

The constant $k_T$ shown in the above equation is the proportionality factor for liquid expansion with increasing temperature. The difference in temperature between the measured condition ($T$) and the reference condition ($T_{ref}$) multiplied by this factor determines how much less dense the

[16]Liquids can and do compress, the measurement of their "compressibility" being what is called the *bulk modulus*. However, this compressibility is too slight to be of any consequence in most flow measurement applications.

liquid is compared to its density at the reference temperature. It should be noted that some liquids – notably hydrocarbons – have thermal expansion factors significantly greater than water. This makes temperature compensation for hydrocarbon liquid flow measurement very important if the measurement principle is volumetric rather than mass-based.

### 13.1.6 Equation summary

Volumetric flow rate ($Q$):

$$Q = N \frac{CYA_2}{\sqrt{1 - \left(\frac{A_2}{A_1}\right)^2}} \sqrt{\frac{P_1 - P_2}{\rho_f}}$$

Mass flow rate ($W$):

$$W = N \frac{CYA_2}{\sqrt{1 - \left(\frac{A_2}{A_1}\right)^2}} \sqrt{\rho_f (P_1 - P_2)}$$

Where,

$Q$ = Volumetric flow rate (e.g., gallons per minute, flowing cubic feet per second)

$W$ = Mass flow rate (e.g., kilograms per second, slugs per minute)

$N$ = Unit conversion factor

$C$ = Discharge coefficient (accounts for energy losses, Reynolds number corrections, pressure tap locations, etc.)

$Y$ = Gas expansion factor ($Y = 1$ for liquids)

$A_1$ = Cross-sectional area of mouth

$A_2$ = Cross-sectional area of throat

$\rho_f$ = Fluid density at flowing conditions (actual temperature and pressure at the element)

The beta ratio ($\beta$) of a differential-producing element is the ratio of throat diameter to mouth diameter ($\beta = \frac{d}{D}$). This is the primary factor determining acceleration as the fluid increases velocity entering the constricted throat of a flow element (venturi tube, orifice plate, wedge, etc.). The following expression is often called the *velocity of approach factor* (commonly symbolized as $E_v$), because it relates the velocity of the fluid through the constriction to the velocity of the fluid as it approaches the flow element:

$$E_v = \frac{1}{\sqrt{1 - \beta^4}} = \text{ Velocity of approach factor}$$

This same velocity approach factor may be expressed in terms of mouth and throat areas ($A_1$ and $A_2$, respectively):

$$E_v = 1\sqrt{1 - \left(\frac{A_2}{A_1}\right)^2} = \text{ Velocity of approach factor}$$

When computing the volumetric flow of a gas in *standard* volume units (e.g., scfm), the equation becomes much more complex than the simple (flowing) volumetric rate equation. Any equation computing flow in standard units must predict the effective expansion of the gas if it were to transition

from flowing conditions (the actual pressure and temperature it experiences flowing through the pipe) to standard conditions (one atmosphere pressure at 60 degrees Fahrenheit). The compensated gas flow measurement equation published by the American Gas Association (AGA Report #3) in 1992 for orifice plates with flange taps calculates this expansion to standard conditions with a series of factors accounting for flowing and standard ("base") conditions, in addition to the more common factors such as velocity of approach and gas expansion. Most of these factors are represented in the AGA3 equation by different variables beginning with the letter $F$:

$$Q = F_n(F_c + F_{sl})YF_{pb}F_{tb}F_{tf}F_{gr}F_{pv}\sqrt{h_W P_{f1}}$$

Where,

$Q$ = Volumetric flow rate (standard cubic feet per hour – SCFH)

$F_n$ = Numeric conversion factor (accounts for certain numeric constants, unit-conversion coefficients, and the velocity of approach factor $E_v$)

$F_c$ = Orifice calculation factor (a polynomial function of the orifice plate's $\beta$ ratio and Reynolds number), appropriate for flange taps

$F_{sl}$ = Slope factor (another polynomial function of the orifice plate's $\beta$ ratio and Reynolds number), appropriate for flange taps

$F_c + F_{sl} = C_d$ = Discharge coefficient, appropriate for flange taps

$Y$ = Gas expansion factor (a function of $\beta$, differential pressure, static pressure, and specific heats)

$F_{pb}$ = Base pressure factor = $14.73\frac{\text{psia}}{P_b}$, with pressure in psia (absolute)

$F_{tb}$ = Base temperature factor = $\frac{T_b}{519.67}$, with temperature in degrees Rankine

$F_{tf}$ = Flowing temperature factor = $\sqrt{\frac{519.67}{T_f}}$, with temperature in degrees Rankine

$F_{gr}$ = Real gas relative density factor = $\sqrt{\frac{1}{G_r}}$

$F_{pv}$ = Supercompressibility factor = $\sqrt{\frac{Z_b}{Z_{f1}}}$

$h_W$ = Differential pressure produced by orifice plate (inches water column)

$P_{f1}$ = Flowing pressure of gas at the upstream tap (psia)

## 13.2 Laminar flowmeters

A unique form of differential pressure-based flow measurement deserves its own section in this flow measurement chapter, and that is the *laminar* flowmeter.

Laminar flow is a condition of fluid motion where viscous (internal fluid friction) forces greatly overshadow inertial (kinetic) forces. A flowstream in a state of laminar flow exhibits no turbulence, with each fluid molecule traveling in its own path, with limited mixing and collisions with adjacent molecules. The dominant mechanism for resistance to fluid motion in a laminar flow regime is

friction with the pipe or tube walls. Laminar flow is qualitatively predicted by low values of Reynolds number.

This pressure drop created by fluid friction in a laminar flowstream is quantifiable, and is expressed in the Hagen-Poiseuille equation:

$$Q = k\left(\frac{\Delta P D^4}{\mu L}\right)$$

Where,

$Q$ = Flow rate
$\Delta P$ = Pressure dropped across a length of pipe
$D$ = Pipe diameter
$\mu$ = Fluid viscosity
$L$ = Pipe length
$k$ = Coefficient accounting for units of measurement

Laminar flowmeter elements generally consist of one or more tubes whose length greatly exceeds the inside diameter, arranged in such a way as to produce a slow-moving flow velocity. An example is shown here:

Figure 13.41: A laminar flow element with transmitter

The expanded diameter of the flow element ensures a lower fluid velocity than in the pipes entering and exiting the element. This decreases the Reynolds number to the point where the flow regime exhibits laminar behavior. The large number of small-diameter tubes packed in the wide area of the element provide adequate wall surface area for the fluid's viscosity to act upon, creating an overall pressure drop from inlet to outlet which is measured by the differential pressure transmitter. This pressure drop is permanent (no recovery of pressure downstream) because the mechanism of pressure drop is friction: total dissipation (loss) of energy in the form of heat.

Another common form of laminar flow element is simply a coiled *capillary tube*: a long tube with a very small inside diameter. The small inside diameter of such a tube makes wall-boundary

effects dominant, such that the flow regime will remain laminar over a wide range of flow rates. The extremely restrictive nature of a capillary tube, of course, limits the use of such flow elements to very low flow rates such as those encountered in the sampling networks of certain analytical instruments.

A unique advantage of the laminar flowmeter is its linear relationship between flow rate and developed pressure drop. It is the only pressure-based flow measurement device for filled pipes that exhibits a linear pressure/flow relationship. This means no "square-root" characterization is necessary to obtain linear flow measurements with a laminar flowmeter. The big disadvantage of this meter type is its dependence on fluid viscosity, which in turn is strongly influenced by fluid temperature. Thus, all laminar flowmeters require temperature compensation in order to derive accurate measurements, and some even use temperature *control* systems to force the fluid's temperature to be constant as it moves through the element[17].

Laminar flow elements find their widest application inside pneumatic instruments, where a linear pressure/flow relationship is highly advantageous (behaving like a "resistor" for instrument air flow) and the viscosity of the fluid (instrument air) is relatively constant. Pneumatic controllers, for instance, use laminar restrictors as part of the derivative and integral calculation modules, the combination of "resistance" from the restrictor and "capacitance" from volume chambers forming a sort of pneumatic time-constant ($\tau$) network.

## 13.3 Variable-area flow gauges

An *Variable-area* flow gauge is one where the fluid must pass through a restriction whose area increases with flow rate.

### 13.3.1 Rotameters

The simplest example of a variable-area flow gauge is the *rotameter*, which uses a solid object (called a *plummet* or *float*) as a flow indicator, suspended in the midst of a tapered tube:

[17] This includes elaborate oil-bath systems where the laminar flow element is submerged in a temperature-controlled oil bath, the purpose of which is to hold temperature inside the laminar element constant despite sudden changes in the measured fluid's temperature.

Figure 13.42: Diagram of a variable area flow gauge

As fluid flows upward through the tube, a pressure differential develops across the plummet. This pressure differential, acting on the effective area of the plummet body, develops an upward force ($F = \frac{P}{A}$). If this force exceeds the weight of the plummet, the plummet moves up. As the plummet moves further up in the tapered tube, the area between the plummet and the tube walls (through which the fluid must travel) grows larger. This increased flowing area allows the fluid to make it past the plummet without having to accelerate as much, thereby developing less pressure drop across the plummet's body. At some point, the flowing area reaches a point where the pressure-induced force on the plummet body exactly matches the weight of the plummet. This is the point in the tube where the plummet stops moving, indicating flow rate by it position relative to a scale mounted (or etched) on the outside of the tube.

The following rotameter uses a spherical plummet, suspended in a flow tube machined from a solid block of clear plastic. An adjustable valve at the bottom of the rotameter provides a means for adjusting gas flow:

Figure 13.43: A rotameter flow gauge

The same basic flow equation used for pressure-based flow elements holds true for rotameters as well:

$$Q = k\sqrt{\frac{P_1 - P_2}{\rho}}$$

However, the difference in this application is that the value of the radicand is constant, since the pressure difference will remain constant[18] and the fluid density will likely remain constant as well. Thus, $k$ will change in proportion to $Q$. The only variable within $k$ relevant to plummet position is the flowing area between the plummet and the tube walls.

Most rotameters are indicating devices only. They may be equipped to transmit flow information electronically by adding sensors to detect the plummet's position in the tube, but this is not common

[18] If the plummet's weight will remain constant, its area will remain constant, and that the force generated by the pressure drop will always be in equilibrium with the plummet's weight for any steady flow rate, then the relationship $F = \frac{P}{A}$ dictates a constant pressure. Thus, the rotameter may be classified as a *constant-pressure*, *variable-area* flowmeter. This stands in contrast to devices such as orifice plates, which are *variable-pressure*, *constant-area*.

practice.

Rotameters are very commonly used as purge flow indicators for pressure and level measurement systems requiring a constant flow of purge fluid (see pages 343 and 417 for examples). Such rotameters are usually equipped with hand-adjustable needle valves for manual regulation of purge fluid flow rate.

### 13.3.2 Open-channel flowmeters

A very different style of variable-area flowmeter is used extensively to measure flow rate through open channels, such as irrigation ditches. If an obstruction is placed within a channel, any liquid flowing through the channel must rise on the upstream side of the obstruction. By measuring this liquid level rise, it is possible to infer the rate of liquid flow past the obstruction.

#### Weirs

The first form of open-channel flowmeter is the *weir*, which is nothing more than a dam obstructing passage of liquid through the channel. Three styles of weir are shown in the following illustration; the *rectangular*, *Cippoletti*, and *V-notch*:

Figure 13.44: Different shapes used as weir flow elements

A rectangular weir has a notch of simple rectangular shape, as the name implies. A Cippoletti weir is much like a rectangular weir, except that the vertical sides of the notch have a 4:1 slope (rise of 4, run of 1; approximately a 14 degree angle from vertical). A V-notch weir has a triangular notch, customarily measuring either 60 or 90 degrees.

The following photograph shows water flowing through a Cippoletti weir:

Figure 13.45: A Cippoletti weir flow element

At a condition of zero flow through the channel, the liquid level will be at or below the crest (lowest point on the opening) of the weir. As liquid begins to flow through the channel, it must spill over the crest of the weir in order to get past the weir and continue downstream in the channel. In order for this to happen, the level of the liquid upstream of the weir must rise above the weir's crest height. This height of liquid upstream of the weir represents a hydrostatic pressure, much the same as liquid heights in piezometer tubes represent pressures in a liquid flowstream through an enclosed pipe. The height of liquid above the crest of a weir is analogous to the pressure differential generated by an orifice plate. As liquid flow is increased even more, a greater pressure (head) will be generated upstream of the weir, forcing the liquid level to rise. This effectively increases the cross-sectional area of the weir's "throat" as a taller stream of liquid exits the notch of the weir[19].

Figure 13.46: The function of a weir flow element is illustrated

[19]Orifice plates are *variable-pressure*, *constant-area* flowmeters. Rotameters are *constant-pressure*, *variable-area* flowmeters. Weirs are *variable-pressure*, *variable-area* flowmeters. As one might expect, the mathematical functions describing each of these flowmeter types is unique!

Figure 13.47: Dependence of notch area on flow rate for a weir flow element

This dependence of notch area on flow rate creates a very different relationship between flow rate and liquid height (measured above the crest) than the relationship between flow rate and differential pressure in an orifice plate:

$$Q = 3.33(L - 0.2H)H^{1.5} \qquad \text{Rectangular weir}$$

$$Q = 3.367LH^{1.5} \qquad \text{Cippoletti weir}$$

$$Q = 2.48\left(\tan\frac{\theta}{2}\right)H^{2.5} \qquad \text{V-notch weir}$$

Where,
$Q$ = Volumetric flow rate (cubic feet per second – CFS)
$L$ = Width of crest (feet)
$\theta$ = V-notch angle (degrees)
$H$ = Head (feet)

As can be seen from a comparison of characteristic flow equations between these three types of weirs, the shape of the weir's notch has a dramatic effect on the mathematical relationship between flow rate and head (liquid level upstream of the weir, measured above the crest height). This implies that it is possible to create almost any characteristic equation by carefully shaping the weir's notch in some custom form. A good example of this is the so-called *proportional* or *Sutro* weir:

Figure 13.48: Cross-section of a proportional weir flow element

This weir design is not used very often, due to its inherently weak structure and tendency to clog with debris.

**Flumes**

A variation on the theme of a weir is another open-channel device called a *flume*. If weirs may be thought of as open-channel orifice plates, then flumes may be thought of as open-channel venturi tubes:

Figure 13.49: Diagram of a flume flow element

Like weirs, flumes generate upstream liquid level height changes indicative of flow rate. One of the most common flume design is the *Parshall flume*, named after its inventor R.L. Parshall when it was developed in the year 1920.

The following formulae relate head (upstream liquid height) to flow rate for free-flowing Parshall flumes[20]:

[20] It is also possible to operate a Parshall flume in fully *submerged* mode, where liquid level must be measured at both the upstream and throat sections of the flume. Correction factors must be applied to these equations if the flume is submerged.

$$Q = 0.992H^{1.547} \qquad \text{3-inch wide throat Parshall flume}$$

$$Q = 2.06H^{1.58} \qquad \text{6-inch wide throat Parshall flume}$$

$$Q = 3.07H^{1.53} \qquad \text{9-inch wide throat Parshall flume}$$

$$Q = 4LH^{1.53} \qquad \text{1-foot to 8-foot wide throat Parshall flume}$$

$$Q = (3.6875L + 2.5)H^{1.53} \qquad \text{10-foot to 50-foot wide throat Parshall flume}$$

Where,
$Q$ = Volumetric flow rate (cubic feet per second – CFS)
$L$ = Width of flume throat (feet)
$H$ = Head (feet)

Flumes are generally less accurate than weirs, but they do enjoy the advantage of being inherently self-cleaning. If the liquid stream being measured is drainage- or waste-water, a substantial amount of solid debris may be present in the flow that could cause repeated clogging problems for weirs. In such applications, flumes are often the more practical flow element for the task (and more accurate over the long term as well, since even the finest weir will not register accuracy once it becomes fouled by debris).

Once a weir or flume has been installed in an open channel to measure the flow of liquid, some method must be employed to sense upstream liquid level and translate this level measurement into a flow measurement. Perhaps the most common technology for weir/flume level sensing is *ultrasonic* (see section 12.5.1, page number 444, for more information on how this technology works). Ultrasonic level sensors are completely non-contact, which means they cannot become fouled by the process liquid (or debris in the process liquid). However, they may be "fooled" by foam or debris floating on top of the liquid, as well as waves on the liquid surface.

The following photograph shows a Parshall flume measuring effluent flow from a municipal sewage treatment plant, with an ultrasonic transducer mounted above the middle of the flume to detect water level flowing through:

Figure 13.50: An installed Parshall flume

Once the liquid level is successfully measured, a computing device is used to translate that level measurement into a suitable flow measurement (and in some cases even integrate that flow measurement with respect to time to arrive at a value for total liquid volume passed through the element, in accordance with the calculus relationship $V = \int Q\,dt + C$).

A technique for providing a clean and "quiet" (still) liquid surface to measure the level of is called a *stilling well*. This is an open-top chamber connected to the weir/flume channel by a pipe, so the liquid level in the stilling well matches the liquid level in the channel. The following illustration shows a stilling well connected to a weir/flume channel, with the direction of liquid flow in the channel being perpendicular to the page (i.e. either into or out of the page):

Figure 13.51: Diagram showing the use of a stilling well for open channel flow measurement

To discourage plugging of the passageway connecting the stilling well to the channel, a small flow rate of clean water may be introduced into the well. This forms a constant *purge flow* into the channel, flushing out debris that might otherwise find its way into the connecting passageway to plug it up. Note how the purge water enters the stilling well through a submerged tube, so it does not cause splashing on the water's surface inside the well which could cause measurement problems for the ultrasonic sensor:

Figure 13.52: Diagram showing the use of a purged stilling well for open channel flow measurement

A significant advantage that weirs and flumes have over other forms of flow measurement is exceptionally high *rangeability*: the ability to measure very wide ranges of flow with a modest pressure (height) span. Another way to state this is to say that the accuracy of a weir or flume is quite high even at low flow rates.

A three-image representation of liquid flow through a rectangular weir was shown in Figure 13.47. As fluid flow rate increased, so did the height (head) of the liquid upstream of the weir. The height of liquid upstream of the weir depends on the flow rate (volumetric $Q$ or mass $W$) as well as the effective area of the notch through which the fluid must pass. Unlike an orifice plate, this area changes with flow rate in both weirs and flumes. One way to envision this by comparison is to imagine a weir as acting like an elastic orifice plate, whose bore area increases with flow rate. This flow-dependent notch area exhibited by both weirs and flumes means that these devices become *more sensitive* to changes in flow as the flow rate becomes smaller.

A comparison of transfer function graphs for closed-pipe head elements such as orifice plates and venturi tubes versus weirs and flumes shows this striking difference in characteristics:

Figure 13.53: Comparison of head vs. flow for dp and open-channel flow elements

Looking at the orifice plate / venturi tube graph near the lower-left corner, it is easy to see that small changes in flow result in extremely small changes in head (differential pressure) because the function has a very low slope (small $\frac{dH}{dQ}$) at that end. By comparison, a weir or flume produces relatively large changes in head (liquid elevation) for small changes in flow near the bottom end of the range, because the function has a very steep slope (large $\frac{dH}{dQ}$) at that end.

The practical advantage this gives weirs and flumes is the ability to maintain high accuracy of flow measurement at very low flow rates – something a fixed-orifice element simply cannot do. It is commonly understood in industry that traditional orifice plate flowmeters cannot maintain good measurement accuracy much below a third of their full-range flow (a rangeability of 3:1), whereas weirs (especially the V-notch design) can achieve far greater rangeability (up to 500:1 according to some sources[21]).

## 13.4 Velocity-based flowmeters

The Law of Continuity for fluids states that the product of mass density ($\rho$), cross-sectional pipe area ($A$) and average velocity ($\overline{v}$) must remain constant through any continuous length of pipe:

[21] These figures are reported in Béla Lipták's excellent reference book *Instrument Engineers' Handbook – Process Measurement and Analysis Volume I* (Fourth Edition). To be fair to closed-pipe elements such as orifice plates and venturi tubes, much improvement in the classic 3:1 rangeability limitation has been achieved through the use of microprocessor-based differential pressure sensors. Lipták reports rangeabilities for orifice plates as great as 10:1 through the use of such modern differential pressure instruments. However, even this pales in comparison to the rangeability of a typical weir or flume, which Lipták reports to be 75:1 for "most devices" in this category.

Figure 13.54: Illustration of the Law of Continuity

If the density of the fluid is not subject to change as it travels through the pipe (a very good assumption for liquids), the Law of Continuity may be simplified by eliminating the density terms from the equation:

$$A_1\overline{v_1} = A_2\overline{v_2}$$

The product of cross-sectional pipe area and average fluid velocity is the volumetric flow rate of the fluid through the pipe ($Q = A\overline{v}$). This says that fluid velocity will be directly proportional to volumetric flow rate given a known cross-sectional area and a constant density for the fluid flowstream. Any device able to directly measure fluid velocity is therefore capable of inferring volumetric flow rate of fluid in a pipe. This is the basis for *velocity-based* flowmeter designs.

### 13.4.1 Turbine flowmeters

*Turbine* flowmeters use a free-spinning turbine wheel to measure fluid velocity, much like a miniature windmill installed in the flow stream. The fundamental design goal of a turbine flowmeter is to make the turbine element as free-spinning as possible, so no torque will be required to sustain the turbine's rotation. If this goal is achieved, the turbine blades will achieve a rotating (tip) speed that equalizes with the linear velocity of the fluid:

Figure 13.55: Diagram showing the profile of a turbine flow element

A cut-away demonstration model of a turbine flowmeter is shown in the following photograph. The blade sensor may be seen protruding from the top of the flowtube, just above the turbine wheel:

Figure 13.56: Cutaway showing the interior of a turbine flow element

Note the sets of "flow conditioner" vanes immediately before and after the turbine wheel in the photograph. As one might expect, turbine flowmeters are very sensitive to *swirl* in the process fluid flowstream. In order to achieve high accuracy, the flow profile must not be swirling in the vicinity of the turbine, lest the turbine wheel spin faster or slower than it should to represent the velocity of a straight-flowing fluid.

Each blade on the turbine acts as an inclined plane for the fluid molecules as they pass by. The angle of the blades determines the ratio of tip speed to fluid velocity[22].

Turbine speed may be transmitted to an indicator mechanically by means of cables and/or gears, electronically by means of magnetic sensor using a "pickup" coil to generating voltage pulses as the turbine blades rotate underneath, or even optically in some applications by reflecting light off the specific locations on the turbine wheel (the light pulses conveyed to and from the wheel via fiber-optic cables). Pickup coils are preferred over mechanical cables or gears for the simple reason of less resistance to rotation. Cables and gears always present some degree of friction to the turbine's rotation, causing the flowmeter to register less flow than there actually is. Magnetic pickup sensors, however, are frictionless.

The rotational speed of the turbine wheel directly relates to fluid velocity, which is proportional to volumetric flow rate. If a magnetic pickup is used to detect the turbine blades as they pass by the sensor, the *frequency* of the AC voltage signal output by the sensor relates directly to fluid velocity

[22]For instance, a blade angle of 45 degrees would make blade tip speed equal to fluid velocity. A blade angle of only 30 degrees (from the turbine shaft centerline) would result in a blade tip speed of about one-half fluid velocity.

(and volumetric flow).

Since volumetric flow and pickup coil output frequency are directly proportional to each other, this relationship may be expressed in the form of an equation:

$$f = kQ$$

Where,

$f$ = Frequency of output signal (Hz)

$Q$ = Volumetric flow rate (e.g., gallons per second)

$k$ = "K" factor of the turbine element (e.g., pulses per gallon)

Dimensional analysis confirms the validity of this equation. Using units of gps (gallons per second) and pulses per gallon, the product of these two quantities is indeed pulses per second (equivalent to cycles per second, or Hz):

$$\left[\frac{\text{Pulses}}{\text{s}}\right] = \left[\frac{\text{Pulses}}{\text{gal}}\right]\left[\frac{\text{gal}}{\text{s}}\right]$$

Using algebra to solve for flow ($Q$), the quotient of frequency and $k$ factor that yields a volumetric flow rate for a turbine flowmeter:

$$Q = \frac{f}{k}$$

If pickup signal frequency directly represents volumetric flow rate, then the total number of pulses accumulated in any given time span will represent the amount of fluid volume passed through the turbine meter over that same time span. This may be expressed algebraically as the product of average flow rate ($\overline{Q}$), average frequency ($\overline{f}$), $k$ factor, and time:

$$V = \overline{Q}t = \frac{\overline{f}t}{k}$$

A more sophisticated way of calculating total volume passed through a turbine meter requires calculus, representing total volume as the time-integral of instantaneous signal frequency and $k$ factor over a period of time from $t = 0$ to $t = T$:

$$V = \int_0^T Q\,dt \qquad \text{or} \qquad V = \int_0^T \frac{f}{k}\,dt$$

Approximately the same result may be achieved simply by using a digital counter circuit to totalize pulses output by the pickup coil and a microprocessor to calculate volume in whatever unit of measurement deemed appropriate.

As with the orifice plate flow element, standards have been drafted for the use of turbine flowmeters as precision measuring instruments in gas flow applications, particularly the custody transfer[23] of natural gas. The American Gas Association has published a standard called the Report #7 specifying the installation of turbine flowmeters for high-accuracy gas flow measurement, along

[23] "Custody transfer" refers to measurement applications where a product is exchanging ownership. In other words, someone is selling, and someone else is buying, quantities of fluid as part of a business transaction. It is not difficult to understand why accuracy is important in such applications, as both parties have a vested interest in a fair exchange. Government institutions also have a stake in accurate metering, as taxes are typically levied on the sale of commodity fluids such as natural gas.

with the associated mathematics for precisely calculating flow rate based on turbine speed, gas pressure, and gas temperature.

The following photograph shows three AGA-7-compliant installations of turbine flowmeters for measuring the flow rate of natural gas:

Figure 13.57: Installed turbine flow elements and associated transmitters

Note the pressure-sensing and temperature-sensing instrumentation installed in the pipe, reporting gas pressure and gas temperature to a flow-calculating computer (along with turbine pulse frequency) for the calculation of natural gas flow rate.

Less-critical gas flow measurement applications may use a "compensated" turbine flowmeter that mechanically performs the same pressure- and temperature-compensation functions on turbine speed to achieve true gas flow measurement, as shown in the following photograph:

Figure 13.58: A compensated turbine flow element with indicator

The particular flowmeter shown in the above photograph uses a filled-bulb temperature sensor (note the coiled, armored capillary tube connecting the flowmeter to the bulb) and shows total gas flow by a series of pointers, rather than gas flow *rate*.

### 13.4.2 Vortex flowmeters

When a fluid moves with high Reynolds number past a stationary object (a "bluff body"), there is a tendency for the fluid to form *vortices* on either side of the object. Each vortex will form, then detach from the object and continue to move with the flowing gas or liquid, one side at a time in alternating fashion. This phenomenon is known as *vortex shedding*, and the pattern of moving vortices carried downstream of the stationary object is known as a *vortex street*.

It is commonplace to see the effects of vortex shedding on a windy day by observing the motion of flagpoles, light poles, and tall smokestacks. Each of these objects has a tendency to oscillate perpendicular to the direction of the wind, owing to the pressure variations caused by the vortices as they alternately form and break away from the object:

Figure 13.59: Illustration of flagpole motion due to vortex shedding

This alternating series of vortices was studied by Vincenc Strouhal in the late nineteenth century and later by Tódor von Kármán in the early twentieth century. It was determined that the distance between successive vortices downstream of the stationary object is relatively constant, and directly proportional to the width of the object, for a wide range of Reynolds number values[24]. If these vortices are viewed as crests of a continuous wave, the distance between vortices may be represented by the symbol customarily reserved for wavelength: the Greek letter "lambda" ($\lambda$).

Figure 13.60: A bluff body sheds vortices at a constant distance downstream independent of flow rate

The proportionality between object width ($d$) and vortex street wavelength ($\lambda$) is called the *Strouhal number* ($S$), approximately equal to 0.17:

$$\lambda S = d \qquad\qquad \lambda \approx \frac{d}{0.17}$$

If a differential pressure sensor is installed immediately downstream of the stationary object in such an orientation that it detects the passing vortices as pressure variations, an alternating signal will be detected:

[24] It is important to note that the vortex-shedding phenomenon ceases altogether if the Reynolds number is too low. Laminar flow produces no vortices, but rather stream-line flow around any object placed in its way.

Figure 13.61: The pressure sensor detects vortices at a constant distance downstream proportional to flow rate

The *frequency* of this alternating pressure signal is directly proportional to fluid velocity past the object, since the wavelength is constant. This follows the classic frequency-velocity-wavelength formula common to all traveling waves ($\lambda f = v$). Since the wavelength is known to be equal to the bluff body's width divided by the Strouhal number (approximately 0.17), this may be substituted into the frequency-velocity-wavelength formula to solve for fluid velocity ($v$) in terms of signal frequency ($f$) and bluff body width ($d$).

$$v = \lambda f$$

$$v = \frac{d}{0.17} f$$

$$v = \frac{df}{0.17}$$

Thus, a stationary object and pressure sensor installed in the middle of a pipe section constitute a form of flow element called a *vortex flow element*. Like a turbine flow element with an electronic "pickup" sensor to detect the passage of rotating turbine blades, the output frequency of a vortex flow element is linearly proportional to volumetric flow rate.

The pressure sensors used in vortex flow elements are not standard differential pressure transmitters, since the vortex frequency is too high to be successfully detected by such bulky instruments. Instead, the sensors are typically piezoelectric crystals. These pressure sensors need not be calibrated, since the amplitude of the pressure waves detected is irrelevant. Only the frequency

of the waves matter for measuring flow rate, and so nearly any pressure sensor with a fast enough response time will suffice.

Like turbine meters, the relationship between sensor frequency ($f$) and volumetric flow rate ($Q$) may be expressed as a proportionality, with the letter $k$ used to represent the constant of proportionality for any particular flowmeter:

$$f = kQ$$

Where,

$f$ = Frequency of output signal (Hz)

$Q$ = Volumetric flow rate (e.g., gallons per second)[25]

$k$ = "K" factor of the vortex shedding flowtube (e.g., pulses per gallon)

This means vortex flowmeters, like electronic turbine meters, each have a particular "$k$ factor" relating the number of pulses generated per unit volume passed through the meter[26]. Counting the total number of pulses over a certain time span yields total fluid volume passed through the meter over that same time span, making the vortex flowmeter readily adaptable for "totalizing" fluid volume just like turbine meters.

Since vortex flowmeters have no moving parts, they do not suffer the problems of wear and lubrication facing turbine meters. There is no moving element to "coast" as in a turbine flowmeter if fluid flow suddenly stops, which means vortex flowmeters are better suited to measuring erratic flows.

A significant disadvantage of vortex meters is a behavior known as *low flow cutoff*, where the flowmeter simply stops working below a certain flow rate. The reason for this is the cessation of vortices when the fluid's Reynolds number drops below a critical value and the flow regime passes from turbulent to laminar. When the flow is laminar, fluid viscosity is sufficient to prevent vortices from forming, causing the vortex flowmeter to register zero flow even when there may be some (laminar) flow through the pipe.

Figure 13.62 shows a Rosemount model 8800C vortex flow transmitter:

[25] Note that if flow rate is to be expressed in units of gallons per *minute* as is customary, the equation must contain a factor for minutes-to-seconds conversion: $f = \frac{kQ}{60}$

[26] This $k$ factor is empirically determined for each flowmeter by the manufacturer using water as the test fluid (a factory "wet-calibration"), to ensure optimum accuracy.

Figure 13.62: An industrial vortex flow transmitter

Figures 13.63a and 13.63b show close-up views of the flowtube assembly, front (left) and rear (right):

(a) Front view

(b) Rear view

Figure 13.63: Close-up views of vortex flowtube assembly

### 13.4.3 Magnetic flowmeters

When an electrical conductor moves perpendicular to a magnetic field, a voltage is induced in that conductor perpendicular to both the magnetic flux lines and the direction of motion. This phenomenon is known as *electromagnetic induction*, and it is the basic principle upon which all electro-mechanical generators operate.

In a generator mechanism, the conductor in question is typically a coil (or set of coils) made of copper wire. However, there is no reason the conductor must be made of copper wire. *Any* electrically conductive substance in motion is sufficient to electromagnetically induce a voltage, even if that substance is a liquid (or a gas[27]).

Consider water flowing through a pipe as shown in Figure 13.64, with a magnetic field passing perpendicularly through the pipe:

Figure 13.64: Illustration of the principles of magnetic flow measurement

The direction of liquid flow cuts perpendicularly through the lines of magnetic flux, generating a voltage along an axis perpendicular to both. Metal electrodes opposite each other in the pipe wall intercept this voltage, making it readable to an electronic circuit.

A voltage induced by the linear motion of a conductor through a magnetic field is called *motional EMF*, the magnitude of which is predicted by the following formula (assuming perfect perpendicularity between the direction of velocity, the orientation of the magnetic flux lines, and the axis of voltage measurement):

$$\mathcal{E} = Blv$$

Where,

[27] Technically, a gas must be super-heated into a *plasma* state before it is able to conduct electricity.

$\mathcal{E}$ = Motional EMF (volts)
$B$ = Magnetic flux density (Tesla)
$l$ = Length of conductor passing through the magnetic field (meters)
$v$ = Velocity of conductor (meters per second)

Assuming a fixed magnetic field strength (constant $B$) and an electrode spacing equal to the fixed diameter of the pipe (constant $l = d$), the only variable capable of influencing the magnitude of induced voltage is velocity ($v$). In this example, $v$ is not the velocity of a wire segment, but rather the average velocity of the liquid flowstream ($\overline{v}$). Since this voltage will be proportional to average fluid velocity, it must also be proportional to volumetric flow rate, since volumetric flow rate is also proportional to average fluid velocity[28]. Thus, this is a type of flowmeter based on electromagnetic induction. These flowmeters are commonly known as *magnetic flowmeters* or simply *mag-flow meters*.

The relationship between volumetric flow rate ($Q$) and motional EMF ($\mathcal{E}$) may be stated more precisely by algebraic substitution. First, write the formula relating volumetric flow to average velocity, and then manipulate it to solve for average velocity:

$$Q = A\overline{v}$$

$$\frac{Q}{A} = \overline{v}$$

Next, restate the motional EMF equation, and then substitute $\frac{Q}{A}$ for $\overline{v}$ to arrive at an equation relating motional EMF to volumetric flow rate ($Q$), magnetic flux density ($B$), pipe diameter ($d$), and pipe area ($A$):

$$\mathcal{E} = Bd\overline{v}$$

$$\mathcal{E} = Bd\frac{Q}{A}$$

$$\mathcal{E} = \frac{BdQ}{A}$$

Since this is a circular pipe, the area and diameter are directly related to each other by the formula $A = \frac{\pi d^2}{4}$. Thus, this definition for area may be substituted into the last equation to arrive at a formula with one less variable (only $d$, instead of both $d$ and $A$):

$$\mathcal{E} = \frac{BdQ}{\frac{\pi d^2}{4}}$$

$$\mathcal{E} = \frac{BdQ}{1}\frac{4}{\pi d^2}$$

$$\mathcal{E} = \frac{4BQ}{\pi d}$$

[28]This is an application of the transitive property in mathematics: if two quantities are both equal to a common third quantity, they must also be equal to each other. This property applies to proportionalities as well as equalities: if two quantities are proportional to a common third quantity, they must also be proportional to each other.

To have a formula defining flow rate $Q$ in terms of motional EMF ($\mathcal{E}$), simply manipulate the last equation to solve for $Q$:

$$Q = \pi d \frac{\mathcal{E}}{4B}$$

This formula will successfully predict flow rate only for absolutely perfect circumstances. In order to compensate for inevitable imperfections, a "proportionality constant" ($k$) is usually included in the formula[29]:

$$Q = k\pi d \frac{\mathcal{E}}{4B}$$

Note the linearity of this equation. Nowhere is there a power, root, or other non-linear mathematical function in the equation for a magnetic flowmeter. This means no special characterization is required to calculate volumetric flow rate.

A few conditions must be met for this formula to successfully infer volumetric flow rate from induced voltage:

- The liquid must be a reasonably good conductor of electricity
- Both electrodes must contact the liquid
- The pipe must be completely filled with liquid
- The flowtube must be properly grounded to avoid errors caused by stray electric currents in the liquid

The first condition is met by careful consideration of the process liquid prior to installation. Magnetic flowmeter manufacturers will specify the minimum conductivity value of the liquid to be measured. The second and third conditions are met by correct installation of the magnetic flowtube in the pipe. The installation must be done in such a way as to guarantee full flooding of the flowtube (no gas pockets). The flowtube is usually installed with electrodes across from each other horizontally (never vertically!) so even a momentary gas bubble will not break electrical contact between an electrode tip and the liquid flowstream.

Electrical conductivity of the process liquid must meet a certain minimum value, but that is all. It is surprising to some technicians that changes in liquid conductivity have little to no effect on flow measurement accuracy. It is not as though a doubling of liquid conductivity will result in a doubling of induced voltage! Motional EMF is strictly a function of physical dimensions, magnetic field strength, and fluid velocity. Liquids with poor conductivity simply present a greater electrical resistance in the voltage-measuring circuit, but this is of little consequence because the input impedance of the detection circuitry is phenomenally high. Common fluid types that will *not* work with magnetic flowmeters include deionized water (e.g., steam boiler feedwater, ultrapure water for pharmaceutical and semiconductor manufacturing) and oils.

Proper grounding of the flowtube is very important for magnetic flowmeters. The motional EMF generated by most liquid flowstreams is very weak (1 millivolt or less!), and therefore may be easily overshadowed by noise voltage present as a result of stray electric currents in the piping and/or liquid. To combat this problem, magnetic flowmeters are usually equipped to shunt stray electric currents around the flowtube so the only voltage intercepted by the electrodes will be the motional EMF produced by liquid flow. Figure 13.65 shows a Rosemount model 8700 magnetic flowtube, with braided-wire grounding straps clearly visible:

[29] The colloquial term in the United States for this sort of thing is *fudge factor*.

Figure 13.65: An industrial magnetic flowmeter

Note how both grounding straps attach to a common junction point on the flowtube housing. This common junction point should also be bonded to a functional earth ground when the flowtube is installed in the process line. This particular flowtube has a stainless steel *grounding ring* on the face of the near flange, connected to one of the braided grounding straps. An identical grounding ring lays on the other flange, but it is not clearly visible in this photograph. These rings provide points of electrical contact with the liquid in installations where the pipe is made of plastic, or where the pipe is metal but lined with a plastic material for corrosion resistance.

Magnetic flowmeters are fairly tolerant of swirl and other large-scale turbulent fluid behavior. They do not require the long straight-runs of pipe upstream and downstream that orifice plates do, which is a great advantage in many piping systems.

Some magnetic flowmeters have their signal conditioning electronics located integral to the flowtube assembly. A couple of examples are shown in Figures 13.66a and 13.66b (a pair of small flowmeters on the left and a large flowmeter on the right):

(a) Endress+Hauser magnetic flowmeters (b) A Toshiba flowmeter

Figure 13.66: Magnetic flowmeters with integral electronics

Other magnetic flowmeters have separate electronics and flowtube assemblies, connected together by shielded cable. In these installations, the electronics assembly is referred to as the flow transmitter (FT) and the flowtube as the flow element (FE), as shown in Figure 13.67:

Figure 13.67: An industrial magnetic flow transmitter and its remote flow element

While in theory a permanent magnet should be able to provide the necessary magnetic flux for a magnetic flowmeter to function, this is almost never done in practice. The reason for this has to do with a phenomenon called *polarization* which occurs when a DC voltage is impressed across a

liquid containing ions (electrically charged molecules). Ionic polarization would soon interfere with detection of the motional EMF if a magnetic flowmeter were to use a constant magnetic flux such as that produced by a permanent magnet. A simple solution to this problem is to alternate the polarity of the magnetic field, so the motional EMF polarity also alternates and never gives the fluid ions enough time to polarize.

This is why magnetic flowmeter tubes almost always use electromagnet coils to generate the magnetic flux necessary for induction to occur. In Figure 13.68 a Foxboro magnetic flowtube with one of the protective covers removed shows these wire coils clearly (in blue):

Figure 13.68: Cutaway view of the interior of an industrial magnetic flowmeter

Perhaps the simplest form of coil excitation is when the coil is energized by 60 Hz AC power taken from the line power source. Since motional EMF is proportional to fluid velocity and to the flux density of the magnetic field, the induced voltage for such a coil will be a sine wave whose amplitude varies with volumetric flow rate.

Unfortunately, if there is any stray electric current traveling through the liquid to produce erroneous voltage drops between the electrodes, chances are it will be 60 Hz AC as well. With the coil energized by 60 Hz AC, any such noise voltage may be falsely interpreted as fluid flow because the sensor electronics has no way to distinguish between 60 Hz noise in the fluid and a 60 Hz motional EMF caused by fluid flow.

A more sophisticated solution to this problem uses a *pulsed* excitation power source for the flowtube coils. This is called *DC* excitation by magnetic flowmeter manufacturers, which is a bit

misleading because these "DC" excitation signals often reverse polarity, appearing more like an AC square wave on an oscilloscope display. The motional EMF for one of these flowmeters will bear the same waveshape, with amplitude once again being the indicator of volumetric flow rate. The sensor electronics can more easily reject any AC noise voltage because the frequency and waveshape of the noise (60 Hz, sinusoidal) will not match that of the flow-induced motional EMF signal.

### 13.4.4 Ultrasonic flowmeters

*Ultrasonic* flowmeters measure fluid velocity by passing high-frequency sound waves along the fluid flow path. Fluid motion influences the propagation of these sound waves, which may then be measured to infer fluid velocity. Two major sub-types of ultrasonic flowmeters exist: *Doppler* and *transit-time*. Both types of ultrasonic flowmeter work by transmitting a high-frequency sound wave into the fluid stream (the *incident* pulse) and analyzing the received pulse.

Doppler flowmeters exploit the *Doppler effect*, which is the shifting of frequency resulting from waves emitted by or reflected by a moving object. Doppler flowmeters bounce sound waves off of bubbles or particulate material in the flow stream, measure the frequency shift, and infer fluid velocity from the magnitude of that shift. Figure 13.69 shows the operating principle of Doppler flowmeters.

Figure 13.69: Illustration of the principles of Doppler ultrasonic flow measurement

If the reflected wave returns from a bubble advancing toward the ultrasonic transducer[30], the reflected frequency will be greater than the incident frequency. If the flow reverses direction and the reflected wave returns from a bubble traveling away from the transducer, the reflected frequency will be less than the incident frequency.

Note that the Doppler effect yields a direct measurement of fluid *velocity* from each echo received by the transducer. This stands in marked contrast to measurements of *distance* based on time-of-flight (time domain reflectometry – where the amount of *time* between the incident pulse and the returned echo is proportional to distance between the transducer and the reflecting surface). In a Doppler flowmeter, the time delay between the incident and reflected pulses is irrelevant. Only the *frequency shift* between the incident and reflected signals matters.

[30] In the industrial instrumentation world, the word "transducer" usually has a very specific meaning: a device used to process or convert standardized instrumentation signals, such as 4-20 mA converted into 3-15 psig, etc. In the general scientific world, however, the word "transducer" describes any device converting one form of energy into another. It is this latter definition of the word used to describe an ultrasonic "transducer" – a device used to convert electrical energy into ultrasonic sound waves, and vice versa.

Doppler-effect ultrasonic flowmeters obviously require flowstream containing bubbles or particulate matter. In many applications this is a normal state of affairs (municipal wastewater, for example). However, some process fluids are simply too clean and too homogeneous to reflect sound waves. In such applications, a different sort of ultrasonic velocity detection technique must be applied.

Transit-time flowmeters, sometimes called *counterpropagation* flowmeters, use a pair of opposed sensors to measure the time difference between a sound pulse traveling with the fluid flow versus a sound pulse traveling against the fluid flow. Since the motion of fluid tends to carry a sound wave along, the sound pulse transmitted downstream will make the journey faster than a sound pulse transmitted upstream. This principle is shown in Figure 13.70.

Figure 13.70: Illustration of the principles of transit-time ultrasonic flow measurement

In this flowmeter design, a clean fluid with no solid impurities is essential for good signal transmission.

One potential problem with the transit-time flowmeter is being able to measure the true average fluid velocity when the flow profile changes with Reynolds number. If just one ultrasonic "beam" is used to probe the fluid velocity, the path this beam takes will likely see a different velocity profile as the flow rate changes (and the Reynolds number changes along with it). Recall the difference in fluid velocity profiles between low Reynolds number flows (left) and high Reynolds number flows (right) shown in Figure 13.71:

(a) Low Reynolds number (b) High Reynolds number

Figure 13.71: Other I/P converters

A popular way to mitigate this problem is to use multiple sensor pairs, sending acoustic signals along multiple paths through the fluid (i.e. a *multipath* ultrasonic flowmeter), and to average the resulting velocity measurements. Dual-beam flowmeters have been in use for well over a decade, and

one manufacturer even has a *five beam* ultrasonic flowmeter model which they claim maintains an accuracy of +/- 0.15% through the laminar-to-turbulent flow regime transition[31].

Some modern ultrasonic flowmeters have the ability to switch back and forth between Doppler and transit-time (counterpropagation) modes, automatically adapting to the fluid being sensed. This capability enhances the suitability of ultrasonic flowmeters to a wider range of process applications.

Ultrasonic flowmeters are adversely affected by swirl and other large-scale fluid disturbances, and as such may require substantial lengths of straight pipe upstream and downstream of the measurement flowtube to stabilize the flow profile.

Advances in ultrasonic flow measurement technology have reached a point where it is now feasible to consider ultrasonic flowmeters for custody transfer measurement of natural gas. The American Gas Association has released a report specifying the use of multipath ultrasonic flowmeters in this capacity (Report #9).

A unique advantage to ultrasonic flow measurement is the ability to measure flow through the use of temporary *clamp-on* sensors rather than a specialized flowtube with built-in ultrasonic transducers. While clamp-on sensors are not without their share of problems[32], they constitute an excellent solution for certain flow measurement applications.

## 13.5 Inertia-based (true mass) flowmeters

Flowmeters based on true mass measurement ignore fluid density, outputting a signal directly (and linearly) proportional to mass flow rate. These are quite useful in the chemical industries, where stoichiometric ratios must be accurately maintained.

### 13.5.1 Coriolis flowmeters

In physics, certain types of forces are classified as *fictitious* or *pseudoforces* because they only appear to exist when viewed from an accelerating perspective (called a *non-inertial reference frame*). The feeling of acceleration either up or down in an elevator or when riding a roller-coaster at an amusement park feels like a force acting against your body, when it is really nothing more than the reaction of your body's inertia to being accelerated by the vehicle. The real force is the force of the vehicle against your body, causing it to accelerate. What you perceive is merely a reaction to that force, and not the primary cause of your discomfort as it might appear to be.

*Centrifugal force* is another example of a "pseudoforce" because although it may appear to be a real force acting on any rotating object, it is in fact nothing more than an inertial reaction. Centrifugal force is a common experience to any child who has ever played on a "merry-go-round:" that perception of a force drawing you away from the center of rotation, toward the rim. The real force acting on any rotating object is toward the center of rotation (a *centripetal* force) which is necessary to make the object radially accelerate toward a center point rather than travel in a straight line as it normally would without any forces acting upon it. When viewed from the perspective of

[31] See page 10 of Friedrich Hofmann's *Fundamentals of Ultrasonic Flow Measurement for industrial applications* paper.

[32] Most notably, the problem of achieving good acoustic coupling with the pipe wall so signal transmission to the fluid and signal reception back to the sensor may be optimized. Also, there is the potential for sound waves to "ring around the pipe" instead of travel through the fluid with clamp-on ultrasonic flowmeters because the sound waves must travel through the full thickness of the pipe walls in order to enter and exit the fluid stream.

the spinning object, however, it would seem there is a force drawing the object away from the center (a *centrifugal* force).

Yet another example of a "pseudoforce" is the *Coriolis force*, more complicated than centrifugal force, arising from motion perpendicular to the axis of rotation in a non-inertial reference frame. The example of a merry-go-round works to illustrate Coriolis force as well: imagine sitting at the center of a spinning merry-go-round, holding a ball. If you gently toss the ball away from you and watch the trajectory of the ball, you will notice it curve rather than travel away in a straight line. In reality, the ball *is* traveling in a straight line (as viewed from an observer standing on the ground), but from your perspective on the merry-go-round, it appears to be deflected by an invisible force which is called the Coriolis force.

In order to generate a Coriolis force, a mass must be moving at a velocity perpendicular to an axis of rotation, as shown in Figure 13.72:

Figure 13.72: Illustration of the principle of Coriolis force in a non-intertial reference frame

The magnitude of this force is predicted by the following vector equation[33]:

$$\vec{F}_c = -2\vec{\omega} \times \vec{v}' m$$

Where,

$\vec{F}_c$ = Coriolis force vector

$\vec{\omega}$ = Angular velocity (rotation) vector

$\vec{v}'$ = Velocity vector as viewed from the rotating reference frame

$m$ = Mass of the object

If the ball is replaced with a fluid moving through a tube, and a rotation vector is created by tilting that tube around a stationary axis (a fulcrum), a Coriolis force develops on the tube in such a way as to oppose the direction of rotation just like the Coriolis force opposed the direction of rotation of the rotating platform in Figure 13.72. This is shown in Figure 13.73:

[33]This is an example of a vector *cross-product* where all three vectors are perpendicular to each other, and the directions follow the right-hand rule.

Figure 13.73: Illustration of Coriolis force in a fluid-filled tube

To phrase this in anthropomorphic terms, the fluid "fights" against this rotation because it "wants" to keep traveling in a straight line. For any given rotational velocity, the amount of "fight" will be directly proportional to the product of fluid velocity and fluid mass. In other words, the magnitude of the Coriolis force will be in direct proportion to the fluid's mass flow rate. This is the basis of a *Coriolis mass flowmeter*.

It can be difficult to engineer a tubing system capable of spinning in circles while carrying a flowstream of pressurized fluid. To bypass the practical difficulties of building a spinning tube system, Coriolis flowmeters are instead built on the principle of a flexible tube that *oscillates* back and forth, producing the same effect in an intermittent fashion rather than continuously. The effect is not unlike shaking a hose side to side as it carries a stream of water, as illustrated in Figure 13.74:

Figure 13.74: Illustration of Coriolis force in a garden hose

A Coriolis flowmeter exactly like the water hose illustration shown above cannot be built unless the process fluid can exit the tubing, so a common Coriolis flowmeter design uses a U-shaped tube that redirects the fluid flow back to the center of rotation. The curved end of the flexible U-tube is forced to shake back and forth while the tube ends anchor to a stationary manifold. This is shown in Figure 13.75:

Figure 13.75: Illustration of a U-tube Coriolis flowmeter element

The two parallel tubes will experience opposite Coriolis forces as the U-tube assembly shakes up and down, causing the U-bend to twist. As mass flow rate through the tube increases, so does the degree of twisting. By monitoring the phase difference of the inlet and outlet tubes during this twisting motion, the mass flow rate of the fluid passing through the tube may be inferred, as indicated in Figure 13.76:

Figure 13.76: Illustration of Coriolis force in a U-tube producing twisting

In order to reduce the amount of vibration generated by a Coriolis flowmeter, and more

importantly to reduce the effect any external vibrations may have on the flowmeter, two identical U-tubes are built next to each other and shaken in complementary fashion (always moving in opposite directions)[34]. Tube twist is measured as *relative* motion from one tube to the next, not as motion between the tube and the stationary housing of the flowmeter. This (ideally) eliminates the effect of any common-mode vibrations on the inferred flow measurement, as shown in Figure 13.77:

Figure 13.77: Illustration of the internals of a U-tube Coriolis flowmeter

Great care is taken by the manufacturer to ensure the two tubes are as close to identical as possible: not only are their physical characteristics precisely matched, but the fluid flow is split very evenly between the tubes[35] so their respective Coriolis forces should be identical in magnitude.

In Figure 13.78, a Rosemount (Micro-Motion) U-tube Coriolis flowmeter demonstration unit shows the U-shaped tubes (one tube is directly above the other in this picture, it is impossible to tell but there are actually two U-tubes):

[34]For those readers with an automotive bent, this is the same principle applied in opposed-cylinder engines (e.g., Porsche "boxer" air-cooled 6-cylinder engine, Volkswagen air-cooled 4-cylinder engine, BMW air-cooled motorcycle twin engine, Citroen 2CV 2-cylinder engine, Subaru 4- and 6-cylinder opposed engines, etc.). Opposite piston pairs are *always* $180^o$ out of phase for the purpose of maintaining mechanical balance: both moving away from the crankshaft or both moving toward the crankshaft, at any given time.

[35]An alternative to splitting the flow is to plumb the tubes in series so they *must* share the exact same flow rate, like series-connected resistors sharing the exact same amount of electrical current.

Figure 13.78: Cutaway view of the internals of a U-tube Coriolis flowmeter

A closer inspection of this flowmeter shows that there are actually two U-tubes, one positioned directly above the other, shaken in complementary directions by a common electromagnetic force coil:

Figure 13.79: Closeup view of the internals of a U-tube Coriolis flowmeter

Two magnetic displacement sensors monitor the relative motions of the tubes and transmit signals to an electronics module for digital processing. One of those sensor coils may be seen in the previous photograph. Both the force coil and the sensor coil are nothing more than permanent magnets surrounded by movable copper wire coils. The main difference between the force coil and the sensor coil is that the force coil is powered by an AC signal to impart a vibratory force to the tubes, whereas the sensor coils are both unpowered so they can detect tube motion by generating AC voltages to be sensed by the electronics module. The force coil is shown in in Figure 13.80a, while one of the two sensor coils appears in 13.80b.

(a) Force coil (b) Sensor coil

Figure 13.80: Coriolis flow sensor elements

Advances in sensor technology and signal processing have allowed the construction of Coriolis flowmeters employing straighter tubes than the U-tube unit previously illustrated and photographed. Straighter tubes are advantageous for reasons of reduced plugging potential and the ability to easily drain all liquids out of the flowmeter when needed.

The tubes of a Coriolis flowmeter are not just conduits for fluid flow, they are also precision spring elements. As such, it is important to precisely know the spring constant value of these tubes so the Coriolis force may be inferred from tube displacement (i.e. how far the tubes twist) and so the fluid density may be inferred from the tubes' resonant frequency. Every Coriolis flow element is factory-tested to determine the flow tubes' mechanical properties, then the electronic transmitter is programmed with the various constant values describing those properties. Figure 13.81 shows a close-up view of the nameplate on a Rosemount (Micro-Motion) Coriolis mass flowmeter, showing the physical constant values determined for that specific flowtube assembly at the time of manufacture.

Figure 13.81: Nameplate of a U-tube Coriolis flowmeter

This means every Coriolis flowmeter element (the tube and sensor assembly) and transmitter (the electronics package outputting the process variable signals) are a matched pair. Elements and transmitters cannot be interchanged without reprogramming the transmitters with the new elements' physical constant values.

Coriolis flowmeters are equipped with RTD temperature sensors to continuously monitor the process fluid temperature. Fluid temperature is important to know because it affects certain properties of the tubes (e.g., spring constant, diameter, and length). The temperature indication is

usually accessible as an auxiliary output, which means a Coriolis flowmeter may double as a (very expensive!) temperature transmitter.

Another variable is measured and (potentially) transmitted by a Coriolis flowmeter, and this variable is fluid *density*. The tubes within a Coriolis flowmeter are shaken at their mechanical resonant frequency to maximize their shaking motion with the least amount of applied power to the force coil possible. The electronics module continuously varies the force coil's AC excitation frequency to maintain mechanical resonance. This resonant frequency happens to change with process fluid density, since the effective mass of the fluid-filled tubes changes with process fluid density[36], and mass is one of the variables influencing the resonant frequency of any physical object. Note the "mass" term in the following formula, describing the resonant frequency of a tensed string:

$$f = \frac{1}{2L}\sqrt{\frac{F_T}{\mu}}$$

Where,

$f$ = Fundamental resonant frequency of string (Hertz)

$L$ = String length (meters)

$F_T$ = String tension (newtons)

$\mu$ = Unit mass of string (kilograms per meter)

This means fluid density, along with fluid temperature, is another variable measured by a Coriolis flowmeter. The ability to simultaneously measure these three variables (mass flow rate, temperature, and density) makes the Coriolis flowmeter a very versatile instrument indeed. This is especially true when the flowmeter in question communicates digitally using a "fieldbus" standard rather than an analog 4-20 mA signal. Fieldbus communication allows multiple variables to be transmitted by the device to the host system (and/or to other devices on the same fieldbus network), allowing the Coriolis flowmeter to do the job of three instruments.

An example of a Coriolis mass flowmeter being used as a multi-variable transmitter appears in Figure 13.82. Note the instrument tag labels in Figure 13.82b (FT, TT, and DT[37]), documenting its use as a flow transmitter, temperature transmitter, and density transmitter, respectively.

[36] Cconsider each tube as a container with a fixed volume capacity, therefore a change in fluid density (e.g., pounds per cubic foot) must result in a change of weight for each tube.

[37] The letter "D" is a user's choice letter when used to indicate as a measured/initiating variable per ANSI/ISA 5.1. Here it has been chosen to indicate density.

(a) A Coriolis transmitter

(b) Use as a multivariable transmitter

Figure 13.82: An installed Coriolis transmitter

Even though a Coriolis flowmeter inherently measures *mass* flow rate, the continuous measurement of fluid density allows the meter to calculate *volumetric flow rate* if this is the preferred means of expressing fluid flow. The relationship between mass flow ($W$), volumetric flow ($Q$), and mass density ($\rho$) is quite simple:

$$W = \rho Q \qquad\qquad Q = \frac{W}{\rho}$$

All the flowmeter's computer must do to output a volumetric flow measurement is take the mass flow measurement value and divide that by the fluid's measured density. A simple exercise in dimensional analysis (performed with metric units of measurement) validates this concept for both forms of the equation shown above:

$$\left[\frac{\text{kg}}{\text{s}}\right] = \left[\frac{\text{kg}}{\text{m}^3}\right]\left[\frac{\text{m}^3}{\text{s}}\right] \qquad\qquad \left[\frac{\text{m}^3}{\text{s}}\right] = \frac{\left[\frac{\text{kg}}{\text{s}}\right]}{\left[\frac{\text{kg}}{\text{m}^3}\right]}$$

Coriolis mass flowmeters are very accurate and dependable. They are also completely immune to swirl and other fluid disturbances, which means they may be located nearly anywhere in a piping system with no need at all for straight-run pipe lengths upstream or downstream of the meter. Their natural ability to measure true mass flow, along with their characteristic linearity and accuracy, makes them ideally suited for custody transfer applications (where the flow of fluid represents product being bought and sold).

Perhaps the greatest disadvantages of Coriolis flowmeters are their high initial cost and large physical size, especially for large pipe sizes. Coriolis flowmeters are also more limited in operating temperature than other types of flowmeters and may have difficulty measuring low-density fluids (gases) and mixed-phase (liquid/vapor) flows. The bent tubes used to sense process flow may also trap process fluid inside to the point where it becomes unacceptable for hygienic (e.g., food processing, pharmaceuticals) applications. Straight-tube Coriolis flowmeter designs, and designs where the angle of the tubes is slight, fare better in this regard than the traditional U-tube Coriolis flowmeter design.

## 13.6 Thermal-based (mass) flowmeters

*Wind chill* is a phenomenon common to nearly everyone who has ever lived in a cold environment. When the ambient air temperature is substantially colder than the temperature of the body, heat will transfer from the body to the surrounding air. If there is no breeze to move air past the body, the air molecules immediately surrounding the body will begin to warm up as they absorb heat from the body, which will then decrease the rate of heat loss. However, if there is even a slight breeze of air moving past the body, the body will come into contact with more cool (unheated) air molecules than it would otherwise, causing a greater rate of heat loss. Thus, perception of the surrounding temperature will be cooler than if there were no breeze.

This principle may be exploited to measure mass flow rate by placing a heated object in the midst of a fluid flowstream and measuring how much heat the flowing fluid convects away from the heated object. The "wind chill" experienced by that heated object is a function of true mass flow rate (and not just volumetric flow rate) because the mechanism of heat loss is the rate at which fluid molecules contact the heated object, with each of those molecules having a definite mass.

The simplest form of thermal mass flowmeter is the *hot-wire anemometer*, used to measure air speed. This flowmeter consists of a metal wire through which an electric current is passed to heat it up. An electric circuit monitors the resistance of this wire (which is directly proportional to wire temperature because most metals have a definite temperature coefficient of resistance). If air speed past the wire increases, more heat will be drawn away from the wire and cause its temperature to drop. The circuit senses this temperature change and compensates by increasing current through the wire to bring its temperature back up to setpoint. The amount of current sent through the wire becomes a representation of mass air flow rate past the wire.

Most mass air flow sensors used in automotive engine control applications employ this principle. It is important for engine control computers to measure *mass* air flow and not just volumetric air flow because it is important to maintain proper air/fuel ratio even if the air density changes due to changes in altitude. In other words, the computer needs to know how many air molecules are entering the engine per second in order to properly meter the correct amount of fuel into the engine for complete and efficient combustion. The "hot wire" mass air flow sensor is simple and inexpensive to produce in quantity, which is why it finds common use in automotive applications.

Industrial thermal mass flowmeters usually consist of a specially designed "flowtube" with two temperature sensors inside: one that is heated and one that is unheated. The heated sensor acts as the mass flow sensor (cooling down as flow rate increases) while the unheated sensor serves to compensate for the "ambient" temperature of the process fluid.

A typical thermal mass flowtube appears in Figure 13.83a (note the swirl vanes in Figure 13.83b designed to introduce large-scale turbulence into the flowstream to maximize the convective cooling effect of the fluid against the heated sensor element).

(a) Flow element

(b) Swirl vanes

Figure 13.83: A thermal mass flowtube

The simple construction of thermal mass flowmeters allows them to be manufactured in very small sizes. The following photograph shows a small device that is not only a mass flow meter, but also a mass flow *controller* with its own built-in throttling valve mechanism and control electronics. To give a sense of scale, the tube fittings seen on the left- and right-hand sides of this device are 1/4 inch, making this photograph nearly full-size:

Figure 13.84: An industrial thermal mass flow controller

An important factor in the calibration of a thermal mass flowmeter is the *specific heat* of the process fluid. "Specific heat" is a measure of the amount of heat energy needed to change the temperature of a standard quantity of substance by some specified amount[38]. Some substances have much greater specific heat values than others, meaning those substances have the ability to absorb (or release) a lot of heat energy without experiencing a great temperature change. Fluids with high specific heat values make good *coolants*, because they are able to remove much heat energy from hot objects without experiencing great increases in temperature themselves. Since thermal mass flowmeters work on the principle of convective cooling, this means a fluid having a high specific heat value will elicit a greater response from a thermal mass flowmeter than the exact same mass flow rate of a fluid having a lesser specific heat value (i.e. a fluid that is not as good of a coolant).

This means the specific heat value of whatever fluid measured with a thermal mass flowmeter

[38]For example, the specific heat of water is 1.00 kcal / kg $\cdot$ C$^o$, meaning that the addition of 1000 calories of heat energy is required to raise the temperature of 1 kilogram of water by 1 degree Celsius, or that 1000 calories of heat energy must be removed to cool that same quantity of water by 1 degree Celsius. Ethyl alcohol, by contrast, has a specific heat value of only 0.58 kcal / kg $\cdot$ C$^o$, meaning it is almost twice as easy to warm up or cool down as water (little more than half the energy required to heat or cool water needs to be transferred to heat or cool the same mass quantity of ethyl alcohol by the same amount of temperature).

must be known, and that its specific heat value will remain constant. For this reason, thermal mass flowmeters are not suitable for measuring the flow rates of fluid streams whose chemical composition is likely to change over time. This limitation is analogous to that of a pressure sensor used to hydrostatically measure the level of liquid in a vessel: in order for this level-measurement technique to be accurate, the density of the liquid must be known and that density must be constant over time.

## 13.7 Positive displacement flowmeters

A *positive displacement* flowmeter is a cyclic mechanism built to pass a fixed volume of fluid through with every cycle. Many positive displacement flowmeters are rotary in nature, meaning each shaft revolution represents a certain volume of fluid has passed through the meter.

Positive displacement flowmeters have been the traditional choice for residential and commercial natural gas flow and water flow measurement in the United States (a simple application of *custody transfer* flow measurement, where the fluid being measured is a commodity bought and sold). The cyclic nature of a positive displacement meter lends itself well to total gas quantity measurement (and not just flow *rate*), as the mechanism may be coupled to a mechanical counter which is read by utility personnel on a monthly basis. A rotary gas flowmeter is shown in the following photograph. Note the odometer-style numerical display on the left-hand end of the meter, totalizing gas usage over time:

Figure 13.85: A residential positive displacement flowmeter

Positive displacement flowmeters rely on moving parts to shuttle quantities of fluid through them, and these moving parts must effectively seal against each other to prevent leakage past the mechanism (which will result in the instrument indicating less fluid passing through than there actually is). The finely-machined construction of a positive displacement flowmeter will suffer damage from grit or other abrasive materials present in the fluid, which means these flowmeters are applicable only to clean fluid flowstreams. Even with clean fluid flowing through, the mechanisms are subject to wear and accumulating inaccuracies over time. However, there is really nothing more definitive for measuring volumetric flow rate than an instrument built to measure individual volumes of fluid with each mechanical cycle. As one might guess, these instruments are completely immune to swirl and other large-scale fluid turbulence, and may be installed nearly anywhere in a piping system (no need for long sections of straight-length pipe upstream or downstream). Positive displacement flowmeters are also very linear, since mechanism cycles are directly proportional to fluid volume.

Cut away views of a large positive displacement flowmeter used to measure the flow of liquid (registering total accumulated volume in units of gallons) are shown in Figure 13.86.

(a) Gear and indicator (b) Rotors

Figure 13.86: A positive displacement flowmeter

Figure 13.86a shows the gear mechanism used to convert rotor motion into a visible total readout. Figure 13.86b shows a close-up of the interlocking rotors (one with three lobes, the other with four slots which those lobes mesh with). Both the lobes and slots are spiral-shaped, such that fluid passing along the spiral pathways must "push" the lobes out of the slots and cause the rotors to

rotate. So long as there is no leakage between rotor lobes and slots, rotor turns will have a precise relationship to fluid volume passed through the flowmeter.

## 13.8 Weighfeeders

A completely different kind of flowmeter is the *weighfeeder*, used to measure the flow of solid material such as powders and grains. One of the most common weighfeeder designs consists of a conveyor belt with a section supported by rollers coupled to one or more load cells, such that a fixed length of the belt is continuously weighed:

Figure 13.87: Diagram illustrating use of a weighfeeder

The load cell measures the weight of a fixed-length belt section, yielding a figure of material weight per linear distance on the belt. A tachometer (speed sensor) measures the speed of the belt. The product of these two variables is the mass flow rate of solid material "through" the weighfeeder:

$$W = \frac{FS}{d}$$

Where,
$W$ = Mass flow rate (e.g., pounds per second)
$F$ = Force of gravity acting on the weighed belt section (e.g., pounds)
$S$ = Belt speed (e.g., feet per second)
$d$ = Length of weighed belt section (e.g., feet)

## 13.9 Change-of-quantity flow measurement

*Flow*, by definition, is the passage of material from one location to another over time. So far this chapter has explored technologies for measuring flow rate en route from source to destination. However, a completely different method exists for measuring flow rates: measuring how much material has either departed or arrived at the terminal locations over time.

Mathematically, flow may be expressed as a ratio of quantity to time. Whether volumetric flow or mass flow, the concept is the same: quantity of material moved per quantity of time. Average flow rates may be expressed as ratios of changes:

$$\overline{W} = \frac{\Delta m}{\Delta t} \qquad \overline{Q} = \frac{\Delta V}{\Delta t}$$

Where,

$\overline{W}$ = Average mass flow rate
$\overline{Q}$ = Average volumetric flow rate
$\Delta m$ = Change in mass
$\Delta V$ = Change in volume
$\Delta t$ = Change in time

Suppose a water storage vessel is equipped with load cells to precisely measure weight (which is directly proportional to mass with constant gravity) as shown in Figure 13.88. Assuming only one pipe entering or exiting the vessel, any flow of water through that pipe will result in the vessel's total weight changing over time:

Figure 13.88: Diagram illustrating use of change-of-weight flow measurement

If the measured mass of this vessel decreased from 74,688 kilograms to 70,100 kilograms between 4:05 AM and 4:07 AM, the average mass flow rate of water leaving the vessel is 2,294 kilograms per minute over that time span.

$$\overline{W} = \frac{\Delta m}{\Delta t} = \frac{70100\text{kg} - 74688\text{ kg}}{4{:}07 - 4{:}05} = \frac{-4588\text{kg}}{2\text{ min}} = 2294\frac{\text{kg}}{\text{min}}$$

Note that this average flow measurement may be determined without any flowmeter of any kind installed in the pipe to intercept the water flow. All the concerns of flowmeters studied thus far (turbulence, Reynolds number, fluid properties, etc.) are completely irrelevant. Practically any flow rate desired may be measured simply by measuring stored weight (or volume) over time. A computer may do this calculation automatically, on practically any time scale desired.

Now suppose the practice of determining average flow rates every two minutes was considered too infrequent. Imagine that operations personnel require flow data calculated and displayed more often than just 30 times an hour. To achieve better time resolution, take weight (mass) measurements more often. Of course, each mass-change interval will be expected to be less with more frequent measurements, but the amount of time divided by in each calculation will be proportionally smaller as well. If the flow rate happens to be absolutely steady, mass may be sampled as frequently as desired and the calculation will still arrive at the same flow rate value as before (sampling mass just once every two minutes). If, however, the flow rate is not steady, sampling more often will allow the immediate "ups" and "downs" of flow behavior to be seen.

Imagine now that a hypothetical "flow computer" takes weight (mass) measurements at an infinitely fast pace: an infinite number of samples per second. Now, this no longer *averages* flow rates over finite periods of time; instead this calculates *instantaneous* flow rate at any given *point* in time.

Rewriting the average flow rate equations as time derivatives:

$$W = \frac{dm}{dt} \qquad\qquad Q = \frac{dV}{dt}$$

Where,

$W$ = Instantaneous mass flow rate
$Q$ = Instantaneous volumetric flow rate
$dm$ = Infinitesimal (infinitely small) change in mass
$dV$ = Infinitesimal (infinitely small) change in volume
$dt$ = Infinitesimal (infinitely small) change in time

Hypothetical computers capable of infinite calculations per second are not needed in order to derive a flow measurement from a mass (or volume) measurement. Analog electronic circuitry exploits the natural properties of resistors and capacitors to essentially do this very thing in real time[39]. An example is shown in Figure 13.89:

Figure 13.89: Schematic of an analog differentiator circuit

Digital computers are used to calculate average flow rates rather than analog electronic circuits calculating instantaneous flow rates in the vast majority of applications. The broad capabilities of digital computers virtually ensures they will be used somewhere in the measurement/control system, so the rationale is to use the existing digital computer to calculate flow rates (albeit imperfectly) rather than complicate the system design with additional (analog) circuitry. As fast as modern digital computers are able to process simple calculations such as these anyway, there is little practical reason to prefer analog signal differentiation except in specialized applications where high speed performance is paramount.

Perhaps the single greatest disadvantage to inferring flow rate by differentiating mass or volume measurements over time is the requirement that the storage vessel have but one flow path in and out. If the vessel has multiple paths for liquid to move in and out (simultaneously), any flow rate calculated on change-in-quantity will be a *net* flow rate only. It is impossible to use this flow measurement technique to measure one flow out of multiple flows common to one liquid storage vessel.

A simple "thought experiment" confirms this fact. Imagine a water storage vessel receiving a flow rate in at 200 gallons per minute. Next, imagine that same vessel emptying water out of a second pipe at the exact same flow rate: 200 gallons per minute. With the exact same flow rate both entering and exiting the vessel, the water level in the vessel will remain constant. Any change-of-quantity flow measurement system would register zero change in mass or volume over time, consequently calculating a flow rate of absolutely zero. Truly, the *net* flow rate for this vessel is zero, but this says nothing about the flow in each pipe, except that those flow rates are equal in magnitude and opposite in direction.

[39] To be precise, the equation describing the function of this analog differentiator circuit is: $V_{out} = -RC\frac{dV_{in}}{dt}$. The negative sign is an artifact of the circuit design – being essentially an inverting amplifier with negative gain – and not an essential element of the math.

## 13.10 Insertion flowmeters

This section does not describe a particular type of flowmeter, but rather a design that may be implemented for several different kinds of flow measurement technologies. When the pipe carrying process fluid is large in size, it may be impractical or cost-prohibitive to install a full-diameter flowmeter to measure fluid flow rate. A practical alternative for many applications is the installation of an *insertion* flowmeter: a probe that may be inserted into or extracted from a pipe, to measure fluid velocity in one region of the pipe's cross-sectional area (usually the center).

A classic example of an insertion flowmeter element is the *Annubar*, a form of averaging pitot tube pioneered by the Dieterich Standard corporation. Figure 13.90 shows the Annubar flow element inserted into a pipe carrying fluid where it generates a differential pressure for a pressure sensor to measure:

Figure 13.90: Diagram of an Annubar insertion flow element with fitting and transmitter

The Annubar element may be extracted from the pipe by loosening a "gland nut" and pulling the assembly out until the end passes through a hand ball valve. Once the element has been extracted this far, the ball valve may be shut and the Annubar completely removed from the pipe. This is shown in Figure 13.91:

Figure 13.91: Diagram illustrating extraction of an Annubar insertion flow element

For safety reasons, a "stop" is usually built into the assembly to prevent someone from accidently pulling the element all the way out with the valve still open.

Other flowmeter technologies manufactured in insertion form include vortex, turbine, and thermal mass. An insertion-type turbine flowmeter appears in the photographs in Figure 13.92.

(a) The element and transmitter

(b) Rotor closeup

Figure 13.92: An insertion turbine flowmeter

If the flow-detection element is compact rather than distributed (as is certainly the case with the turbine flowmeter shown above), care must be taken to ensure correct positioning within the pipe. Since flow profiles are never completely flat, any insertion meter element will register a greater flow rate at the center of the pipe than near the walls. Wherever the insertion element is placed in the pipe diameter, that placement must remain consistent through repeated extractions and re-insertions or else the effective calibration of the insertion flowmeter will change every time it is removed and re-inserted into the pipe. Care must also be taken to insert the flowmeter so the flow element points directly upstream, and not at an angle.

A unique advantage of insertion instruments is that they may be installed in an operating pipe by using specialized *hot-tapping* equipment. A "hot tap" is a procedure whereby a safe penetration is made into a pipe while the pipe is carrying fluid under pressure. The first step in a hot-tapping operation is to weld a "saddle tee" fitting on the side of the pipe, as illustrated in Figure 13.93:

Figure 13.93: Diagram showing a flanged saddle tee fitting welded to a pipe

Next, a ball valve is bolted onto the saddle tee flange as shown in Figure 13.94. This ball valve will be used to isolate the insertion instrument from the fluid pressure inside the pipe:

Figure 13.94: Diagram showing a flanged ball valve attached to the saddle tee fitting

Figure 13.95 shows a special hot-tapping drill bolted to the open end of the ball valve. This drill uses a high-pressure seal to contain fluid pressure inside the drill chamber as a motor spins the drill bit. The ball valve is opened, then the drill bit is advanced toward the pipe wall where it cuts a hole into the pipe. Fluid pressure rushes into the empty chamber of the ball valve and hot-tapping drill as soon as the pipe wall is breached:

Figure 13.95: Diagram showing the hot-tapping process

Once the hole has been completely drilled, the bit is extracted and the ball valve shut to allow removal of the hot-tapping drill, as seen in Figure 13.96:

Figure 13.96: Diagram showing the removal of the hot-tapping drill

Now there is a flanged and isolated connection into the "hot" pipe, through which an insertion flowmeter (or other instrument/device) may be installed.

Hot-tapping is a technical skill, with many safety concerns specific to different process fluids, pipe types, and process applications. This brief introduction to the technique is not intended to be instructional, but merely informational.

## 13.11 Flow switches

A *flow switch* is one detecting the flow of some fluid through a pipe. Flow switches often use "paddles" as the flow-sensing element, the motion of which actuates one or more switch contacts.

Recall from section 9.1 that the "normal" status of a switch is the resting condition of *no stimulation*. A flow switch will be in its "normal" status when it senses minimum flow (i.e., no fluid moving through the pipe). For a flow switch, "normal" status is any fluid flow rate *below* the trip threshold of the switch. Figure 13.97 shows schematic symbols for flow switches.

Figure 13.97: Schematic symbols for flow switches

A simple paddle placed in the midst of a fluid stream generates a mechanical force which may be used to actuate a switch mechanism, as shown in Figure 13.98:

Figure 13.98: An industrial flow switch

## 13.12 Process/instrument suitability

Every flow-measuring instrument exploits a physical principle to measure the flow rate of fluid stream. Understanding each of these principles as they apply to different flow-measurement technologies is the first and most important step in properly applying a suitable technology to the measurement of a particular process stream flow rate. The following table lists the specific operating principles exploited by different flow measurement technologies:

| **Flow measurement technology** | **Operating principle** | **Linearity** | **2-way flow** |
|---|---|---|---|
| Differential pressure | Fluid mass self-acceleration, potential-kinetic energy exchange | $\sqrt{\Delta P}$ | (some) |
| Laminar | Viscous fluid friction | linear | yes |
| Weirs & flumes | Fluid mass self-acceleration, potential-kinetic energy exchange | $H^n$ | no |
| Turbine (velocity) | Fluid velocity spinning a vaned wheel | linear | yes |
| Vortex | von Kármán effect | linear | no |
| Magnetic | Electromagnetic induction | linear | yes |
| Ultrasonic | Sound wave time-of-flight | linear | yes |
| Coriolis | Fluid inertia, Coriolis effect | linear | yes |
| Turbine (mass) | Fluid inertia | linear | (some) |
| Thermal | Convective cooling, specific heat of fluid | linear | no |
| Positive displacement | Movement of fixed volumes | linear | (some) |

A potentially important factor in choosing an appropriate flow measurement technology is the allowable pressure drop. Piping system designers usually specify the allowable pressure drop. Energy loss is caused by pressure drop, and if the loss in the flow element is large, the piping designer may have to specify larger system components such as pipes and pumps or compressors. Some flowmeter designs, such as the common orifice plate, are inexpensive to install but carry a high price in terms of the energy lost in *permanent pressure drop* (the total, non-recoverable loss in pressure from the inlet of the device to the outlet, not the temporary pressure difference between inlet and vena contracta). Energy costs money, and so industrial facilities would be wise to consider the long-term cost of a flowmeter before settling on the one that is cheapest to install. It could very well be, for example, that an expensive venturi tube will cost less after years of operation than a cheap orifice plate[40].

In this regard, certain flowmeters stand above the rest: those with obstructionless flowtubes. Magnetic and ultrasonic flowmeters have no obstructions whatsoever in the path of the flow. This translates to (nearly) zero permanent pressure loss along the length of the tube, and therefore. Thermal mass and straight-tube Coriolis flowmeters are nearly obstructionless, while vortex and turbine meters are only slightly worse.

[40]This is not always the case, as primary elements are often found on throttled process lines. In such cases where a control valve normally throttles the flow rate, any energy dissipated by the orifice plate is simply less energy that the valve would otherwise be required to dissipate. Therefore, the presence or absence of an orifice plate has no net impact on energy dissipation when used on a process flow throttled by a control valve, and therefore does not affect cost over time due to energy loss.

# References


[1] B. Lipták, Ed., Instrument Engineers Handbook: Process Measurement and Analysis, 4th ed. Boca Raton, FL: CRC Press, 2003, vol. 1, ch. 2.

[2] AGA Report No. 3 - Orifice metering of natural gas and other related hydrocarbon fluids, Part 1 (General Equations and Uncertainty Guidelines), 3rd ed., AGA XQ9210, American Gas Association and American Petroleum Institute, Washington, D.C., 1990.

[3] AGA Report No. 3 - Orifice metering of natural gas and other related hydrocarbon fluids, Part 2 (Specification and Installation Requirements), 4th ed., AGA XQ0002, American Gas Association and American Petroleum Institute, Washington, D.C., 2000.

[4] AGA Report No. 3 - Orifice metering of natural gas and other related hydrocarbon fluids, Part 3 (Natural Gas Applications), 3rd ed., AGA XQ9210, American Gas Association and American Petroleum Institute, Washington, D.C., 1992.

[5] AGA Report No. 3 - Orifice metering of natural gas and other related hydrocarbon fluids, Part 4 (Background, Development, Implementation Procedure, and Subroutine Documentation for Empirical Flange-Tapped Discharge Coefficient Equation), 3rd ed., AGA XQ9211, American Gas Association and American Petroleum Institute, Washington, D.C., 1992.

[6] Instrumentation Symbols and Identification, International Society of Automation Std. 5.1, 2009.

[7] V. T. Chow, Open-Channel Hydraulics, 4th ed. New York, NY,: McGraw-Hill Book Company, Inc., 1959.

[8] Flow Measurement User Manual, Form Number A6043, Emerson Process Management, Marshalltown, IA, 2005.

[9] A. E. Fribance, Industrial Instrumentation Fundamentals. New York, NY: McGraw-Hill, 1962.

[10] EJX910A Multivariable Transmitter, Document GS 01C25R01-01E 5th Ed., Yokogawa Electric Corporation, Tokyo, Japan, 2005.

[11] D. C. Giancoli, Physics for Scientists & Engineers, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2000.

[12] H. P. Kallen, Handbook of Instrumentation and Controls. New York, NY: McGraw-Hill, 1961.

[13] R. W. Miller, Flow Measurement Engineering Handbook, 2nd ed. New York, NY: McGraw-Hill, 1989.

[14] J. K. Vennard, Elementary Fluid Mechanics, 3rd ed. New York, NY: John Wiley & Sons, Inc., 1954.

[15] F. K. Hofmann, "Fundamental principles of electromagnetic flow measurement," Krohne Messtechnik GmbH & Co. KG, Duisburg, Germany, 2003.

[16] ——, "Fundamentals of ultrasonic flow measurement for industrial applications," Krohne Messtechnik GmbH & Co. KG, Duisburg, Germany, 2001.

[17] J. F. Price, "A coriolis tutorial," Woods Hole Oceanographic Institution, Woods Hole, MA, 2006.

[18] H. J. Keisler, Elementary Calculus – An Infinitesimal Approach, 2nd ed. Madison, WI: University of Wisconsin, 2000.

[19] L. K. Spink, Principles and Practice of Flow Meter Engineering, 9th ed. Foxboro, MA: The Foxboro Company, 1967.

# Chapter 14

# Electrical Measurements

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Draw standard symbols used for measurement of electrical parameters in power systems
2. Explain how instrument transformers function, including CTs, VTs, and CCVTs
3. Properly determine and apply instrument transformer polarity
4. Discuss the effects of improper burden on instrument transformers
5. Perform burden calculations for instrument transformers
6. Discuss the use of instrument transformer test switches

Electrical power is a commodity in the modern world, bought and sold on the open market like any other. Thus, it is important to be able to measure and control electricity, not only for reasons of efficiency but also for sale, taxation, safety, and reliability of service.

As with any other quantity to be measured and controlled, the systems designed for these purposes may be divided into three general categories: *sensors* to measure, *final control elements* to exert influence, and *controllers* to make the necessary control decisions automatically. This chapter presents measurement instruments for electrical power systems. The devices presented are permanently installed, not handheld. The voltages and currents encountered in electrical power systems at the transmission and distribution levels are too high to measure directly, so some method of attenuation is required. This attenuation may be accomplished using instrument transformers (ITs) or by creating dividers that provide only a fraction of the measurand to an instrument transformer or other sensing element. ITs are designed to linearly reproduce a range of input into a range of output when installed in accordance with the manufacturer's directions.

The two "process variables" relied on most heavily in the field of electrical measurement and control are *voltage* and *current*. From these primary variables, impedance, reactance, resistance, as well as the reciprocals of those quantities (admittance, susceptance, and conductance) may be determined.

Two common types of electrical sensors used in the power industry are *current transformers* (CTs) and *voltage transformers* (VTs). These are precision-ratio electromagnetic transformers used to step high currents and high voltages down to more reasonable levels for the benefit of panel-mounted instruments to receive, display, and/or process.

## 14.1 Current

Standard ammeters cannot measure the hundreds or even thousands of amps flowing through power lines. Thus, some method of reducing the current seen by an ammeter or other receiving device must be used. The most common method of attenuating current is to use a current transformer.

### 14.1.1 Current transformers (CT)

*Current transformers* are used to step down a higher current to a lower current without any phase shift for the purpose of measurement. They also provide electrical isolation between the higher voltages typically seen on the primary so that the equipment attached to the secondary can be rated for lower voltages. The standard symbol for a CT is shown in Figure 14.1.

Figure 14.1: Schematic symbol for a current transformer (CT)

Since CTs are used to provide an output to a wide variety of devices (e.g., meters, protective relays), the output voltage current have been standardized as have the input current ratings of the receiving devices. The standard output current range of the secondary is 0 - 5 A, though a few

CTs deliver 0 - 1 A, and a few others are designed with electronics in them to produce 4-20 mA outputs. The transformation ratings are usually written in the ratio of primary to secondary full range current, e.g., 400:5 or 1000:5.

For the same reasons necessitating the use of voltage (potential) transformers, *current transformers* are used to reduce high current values and isolate high voltage values between the electrical power system conductors and panel-mounted instruments.

Figure 14.2 is a simple diagram illustrating how the line current of a three-phase AC power system may be sensed by a low-current ammeter through the use of a current transformer.

Figure 14.2: Application of a CT

When driving an ammeter – which is essentially a short-circuit (very low resistance) – the CT behaves as a current source to the receiving instrument, sending a current signal to that instrument proportionately representing the power system's line current.

In typical practice a CT does not have a primary "winding" in the conventional sense of the word, but rather uses the line conductor itself as the primary winding, with multiple turns of wire wrapped around a toroidal magnetic core as the secondary winding, as shown in Figure 14.3.

Figure 14.3: Primary and secondary windings of a current transformer

The view of a current transformer's construction in Figure 14.4 shows the wrapping of the secondary turns around the toroidal magnetic core in such a way that the secondary conductor remains parallel to the primary (power) conductor for good magnetic coupling.

Figure 14.4: Toroidal CT construction

With the power conductor serving as a single-turn[1] winding, the multiple turns of secondary wire

[1]This raises an interesting possibility: if the power conductor were to be wrapped around the toroidal core of the CT so that it passes through the center *twice* instead of *once*, the current step-down ratio will be cut in half. For example, a 100:5 CT with the power conductor wrapped around so it passes through the center twice will exhibit an actual current ratio of only 50:5. If wrapped so that it passed through the CT's center three times, the ratio would be reduced to 33.33:5. This useful "trick" may be used in applications where a lesser CT ratio cannot be found, and one must make do with whatever CT happens to be available. However, beware that the current-measuring capacity of the CT will be correspondingly reduced. Each extra turn of the power conductor adds to the magnetic flux experienced by the CT's core for any given amount of line current, making it possible to magnetically saturate the core if the line current exceeds the reduced value (e.g., 50 amps for the home-made 50:5 CT where the line passes twice through the

around the toroidal core of a CT makes it function as a step-up transformer from the perspective of voltage, and as a *step-down* transformer from the perspective of current. The turns ratio of a CT is typically specified as a ratio of full line conductor current to 5 amps, which is a standard output current for power CTs. Therefore, a 100:5 ratio CT outputs 5 amps when the power conductor carries 100 amps.

The turns ratio of a current transformer suggests a danger worthy of note: if the secondary winding of an energized CT is ever open-circuited, it may develop an extremely high voltage as it attempts to force current through the air gap of that open circuit. An energized CT secondary winding acts like a current source, and like all current sources it will develop as great a voltage (potential) as it can when presented with an open circuit. Given the high voltage capability of the power system being monitored by the CT, and the CT turns ratio with more turns in the secondary than in the primary, the ability for a CT to function as a *voltage step-up* transformer poses a significant hazard.

Like any other current source, there is no harm in short-circuiting the output of a CT. Only an open circuit poses risk of damage. For this reason, CT circuits are often equipped with *shorting bars* and/or *shorting switches* to allow technicians to place a short-circuit across the CT secondary winding before disconnecting any other wires in the circuit. Later subsections will elaborate on this topic in greater detail.

Current transformers are manufactured in a wide range of sizes, to accommodate different applications. Figure 14.5 is a photograph of a current transformer showing the "nameplate" label with all relevant specifications. This nameplate specifies the current ratio as "100/5" which means this CT will output 5 amps of current when there is 100 amps flowing through a power conductor passed through the center of the toroid.

center of a 100:5 CT).

Figure 14.5: Photo of a CT showing nameplate

The black and white wire pair exiting this CT carries the 0 to 5 amp AC current signal to any monitoring instrument scaled to that range. That instrument will see $\frac{1}{20}$ ($\frac{5}{100}$) of the current flowing through the power conductor.

Figure 14.6 is a photograph of some much larger CTs intended for installation inside the "bushings[2]" of a large circuit breaker, stored on a wooden pallet.

[2]High-voltage devices situate their connection terminals at the ends of long insulators, to provide a large air gap between the conductors and the grounded metal chassis of the device. The point at which the long insulator (with a conductor inside of it) penetrates the housing of the device is called the *bushing*.

Figure 14.6: Bushing CTs

The installed CTs appear as cylindrical bulges at the base of each insulator on the high-voltage circuit breaker. Figure 14.7 shows flexible conduit running to each bushing CT, carrying the low-current CT secondary signals to a terminal strip inside a panel on the right-hand end of the breaker:

Figure 14.7: Installed high-voltage circuit breaker bushing CTs

Signals from the bushing CTs on a circuit breaker may be connected to *protective relay* devices to trip the breaker in the event of any abnormal condition. If unused, a CT's secondary terminals must be short-circuited at the panel.

Figure 14.8 shows a set of three very large CTs intended for installation at the bushings of a high-voltage power transformer. Each one has a current step-down ratio of 600:5.

Figure 14.8: Bushing CTs ready for installation

Figure 14.9 shows a tiny CT designed for low current measurements, clipped over a wire carrying only a few amps of current. This particular current transformer is constructed in such a way that it may be clipped around an existing wire for temporary test purposes, rather than being a solid toroid where the conductor must be threaded through it for a more permanent installation.

Figure 14.9: A clamp-on CT

This CT ratio of 3000:1 would step down a 5 amp AC signal to 1.667 milliamps AC.

Figure 14.10 shows a current transformer used to measure line current in a 500 kV substation

switchyard. The actual CT coil is located in the red-colored housing at the top of the insulator, where the power conductor passes through. The tall insulator stack provides necessary separation between the conductor and the earth below to prevent high voltage from "jumping" to ground through the air.

Figure 14.10: A high-voltage CT in a substation application

### 14.1.2 CT design

CTs are designed to be as close as possible to an ideal transformer to minimize the effects of resistance and reactance inside the transformer on the reading delivered to its receiving device. In an ideal transformer, the ampere-turns rating, known as magnetomotive force (mmf), is the same for both sides of the transformer ($N_1 i_1 = N_2 i_2$). However, that is not the case in a real transformer. Since some of the primary mmf is used to magnetize the core, there is less mmf to create current on the secondary. By minimizing the primary mmf required to magnetize the core, more mmf is available to deliver flux from the primary to the secondary, thus producing a secondary current that is closer in both magnitude and phase to the primary current multiplied by the turns ratio. However, some error will always exist.

### 14.1.3 Ratings

*Burden* refers to resistive load the secondary can drive. It can be expressed as a resistance in ohms, or as the apparent power (VA) rating of the secondary at full rated secondary current at a particular power factor (often 0.8). Standard ANSI burden ratings are B-0.1, B-0.2, B-0.5, B-1.0, B-2.0, and B-4.0. These correspond to burdens of 2.5 VA, 5 VA, 12.5 VA, 25 VA, 50 VA, and 100 VA[3]. The ANSI standard burden ratings state the load resistance in ohms that the CT can drive current into without affecting the accuracy of the measurement. For example, a B-0.1 rating means the total connected load resistance should not exceed 0.1 Ω or additional error will result. The load resistance includes the resistance of any receiving devices and the resistance of any interconnecting wiring. The interconnecting wiring resistance can be appreciable. For example, 12 AWG has a resistance of about 1.6 Ω per 1000 feet, so a receiver placed 50 feet away will have 160 mΩ of connecting wiring (due to the two interconnecting wires running there and back - a total of 100 feet). Thus, if the receiver resistance exceeds 38 mΩ, the accuracy of a B-0.2 CT would be affected. Thus, either a CT with a higher burden class would be needed, or larger conductors would need to be used as the interconnecting wiring to reduce the lead resistance.

## 14.2 Voltage

### 14.2.1 Voltage Transformers (VT)

*Voltage transformers* are used to step down a higher voltage to a lower voltage without any phase shift for the purpose of measurement. The primary of the transformer is connected between two lines or between a line and neutral. The symbol for a VT is shown in Figure 14.11.

[3] The VA rating can be calculated by multiplying the ANSI resistance rating by the rated secondary current squared ($VA = I^2Z$), which for a X:5 CT is 25. To get the ANSI rating of a CT or receiving device given its VA rating, divide the VA rating by 25.

Figure 14.11: Schematic symbol for a voltage transformer (VT)

Electrical power systems typically operate at dangerously high voltage. It would be both impractical and unsafe to connect panel-mounted instruments directly to the conductors of a power system if the voltage of that power system exceeds several hundred volts. For this reason, a special type of step-down transformer referred to as a *voltage transformer* must be used to reduce and isolate the high line voltage of a power system to levels safe for panel-mounted instruments to input.

Figure 14.12 shows a simple diagram illustrating how the high phase and line voltages of a three-phase AC power system may be sensed by low-voltage voltmeters through the use of step-down voltage transformers.

Figure 14.12: Application of voltage transformers (VTs)

Voltage transformers are commonly referred to as "VT" units in the electrical power industry. It should be noted that use of the term "potential transformer" and its associated abbreviation PT preceded the use of the term "voltage transformer" and its associated abbreviation VT, and may be still be in use by engineers and in documentation, but is no longer the preferred terminology.

When driving a voltmeter – which is essentially an open-circuit (very high resistance) – the VT behaves as a voltage source to the receiving instrument, sending a voltage signal to that instrument proportionately representing the power system's voltage.

Figure 14.13 shows a voltage transformer sensing the phase-to-ground voltage on a three-phase power distribution system. The normal phase voltage in this system is 7.2 kV (12.5 kV three-phase line voltage), and the VT's normal secondary voltage is 120 volts, necessitating a ratio of 60:1 (as shown on the transformer's side).

Figure 14.13: A VT installed on a medium voltage line

Any voltage output by this VT will be $\frac{1}{60}$ of the actual phase voltage, allowing panel-mounted instruments to read a precisely scaled proportion of the 7.2 kV (typical) phase voltage safely and effectively. A panel-mounted voltmeter, for example, would have a scale registering 7200 volts when its actual input terminal voltage was only 120 volts. This is analogous to a 4-20 mA indicating meter bearing a scale labeled in units of "psi" or "Degrees Celsius" because the 4-20 mA analog signal merely represents some other physical variable sensed by a process transmitter. Here, the physical variable being sensed by the voltage transformer is still voltage, just at a 60:1 ratio greater than what the panel-mounted instrument receives. Like the 4-20 mA DC analog signal standard so common in the process industries, 120 volts is the standard voltage transformer output voltage used in the electrical industry to represent normal power system voltage.

Figure 14.14 shows three large VTs used to precisely step the phase-to-ground voltages for each phase of a 230 kV system (230 kV line voltage, 133 kV phase voltage) all the way down to 120 volts for the panel-mounted instruments to monitor.

Figure 14.14: VTs installed on a high voltage line

A loose-hanging wire joins one side of the primary winding of each VT to the respective phase conductor of the 230 kV bus. The other terminal of each VT's primary winding connects to a common neutral point, forming a Wye-connected VT transformer array. The secondary terminals of these VTs connect to two-wire shielded cables conveying the 120 volt signals back to the control room where they terminate at various instruments. These shielded cables run through underground conduit for protection from weather.

Just as with the previous VT, the standard output voltage of these large VTs is 120 volts, equating to a transformer turns ratio of about 1100:1. This standardized output voltage of 120 volts allows VTs of any manufacture to be used with receiving instruments of any manufacture, just as the 4-20 mA standard for analog industrial instruments allows "interoperability" between different manufacturers' brands and models.

### 14.2.2 Capacitively-Coupled Voltage Transformers (CCVTs)

A special form of instrument transformer used on very high-voltage systems is the *capacitively-coupled voltage transformer*, or CCVT. These sensing devices employ a series-connected set of capacitors dividing the power line voltage down to a lesser quantity before it gets stepped down further by an electromagnetic transformer. A simplified diagram of a CCVT appears in Figure 14.15.

Figure 14.15: Diagram of a capacitively-coupled VT

Since VTs are used to provide an output to a wide variety of devices (e.g., meters, protective relays), the output voltage ratings have been standardized as have the input voltage ratings of the receiving devices. The standard output across the secondary is 0 - 120 V. VTs are designed to be as close as possible to an ideal transformer to minimize the effects of resistance and reactance inside the transformer on the reading delivered to its receiving device. Special attention to construction is needed to make sure the receiving device doesn't see voltage and phase angle shifts caused by the VT. CCVTs use at least two capacitors to create a voltage divider from a line with high voltage. Proper selection of the capacitors used controls the ratio of the voltages across each capacitor. A standard VT is attached across the lower potential created by the capacitive divider so that it is not exposed to higher voltages. Damping resistors are often placed in the tap line to prevent resonance from occuring between the capacitance of the divider and the inductance of the primary coil of the VT.

## 14.3 Transformer polarity

An important characteristic to identify for transformers in power systems – both power transformers and instrument transformers – is *polarity*. At first it may seem inappropriate to speak of "polarity"

when dealing with *alternating* voltages and currents, but what is really meant by this word is *phasing*. When multiple power transformers are interconnected in order to share load, or to form a three-phase transformer array from three single-phase transformer units, it is critical that the phase relationships between the transformer windings be known and clearly marked. Also, the phase relationship between the primary and secondary windings (coils) of an instrument transformer must be known in order to properly connect it to a receiving instrument such as a protective relay. For some instruments such as simple indicating meters, polarity (phasing) is unimportant. For other instruments comparing the phase relationships of two or more received signals from instrument transformers, proper polarity (phasing) is critical.

Polarity markings for any transformer may be symbolized several different ways, as shown in Figure 14.16.

Figure 14.16: Transformer polarity markings and notation

The marks should be interpreted in terms of *voltage polarity*, not current. Figure 14.17 illustrates a method to determine polarity using a "test circuit[4]" feeding a momentary pulse of DC to a transformer.

Figure 14.17: Determining polarity markings via DC test

Note how the secondary winding of the transformer develops the same polarity of voltage drop as is impressed across the primary winding by the DC pulse: for both the primary and secondary windings, the sides with the dots share the same positive potential.

[4]The battery-and-switch test circuit shown here is not just hypothetical, but may actually be used to test the polarity of an unmarked transformer. Simply connect a DC voltmeter to the secondary winding while pressing and releasing the pushbutton switch: the voltmeter's polarity indicated while the button is pressed will indicate the relative phasing of the two windings. Note that the voltmeter's polarity will reverse when the pushbutton switch is released and the magnetic field collapses in the transformer coil, so be sure to pay attention to the voltmeter's indication *only* during the time of switch closure!

If the battery were reversed and the test performed again, the side of each transformer winding with the dot would be negative, as shown in Figure 14.18.

Figure 14.18: Determining polarity markings via DC test - reversed source

If the secondary winding's connection to the resistor is reversed and all voltages and currents redrawn as in Figure 14.19, it can be seen that the polarity dot always represents common voltage potential, regardless of source polarity.

Figure 14.19: Determining polarity markings via DC test - reversed winding

To emphasize this important point again: *transformer polarity dots refer to voltage, not current.* The polarity of voltage across a transformer winding will always match the polarity of every other winding on that same transformer in relation to the dots. The direction of current through a transformer winding, however, depends on whether the winding in question is functioning as a *source* or a *load*. This is why currents are seen to be in opposite directions (into the dot, out of the dot) from primary to secondary in all the previous examples shown while the voltage polarities all match the dots. A transformer's primary winding functions as a *load* (conventional-flow current drawn flowing into the positive terminal) while its secondary winding functions as a *source* (conventional-flow current flowing out of the positive terminal).

When three single-phase transformers are interconnected to form a three-phase transformer bank, the winding polarities *must* be properly oriented. Windings in a delta network must be connected such that the polarity marks of no two windings are common to each other. Arrows are drawn next to each winding in Figure 14.20 for emphasis.

Figure 14.20: Delta-connected high-side transformer windings

Windings in a wye network must be connected such that the polarity marks all face the same direction with respect to the center of the wye (typically, the polarity marks are all facing away from the center), as indicated in Figure 14.21.

Figure 14.21: Wye-connected high-side transformer windings

Failure to heed these phase relationships in a power transformer bank may result in tripped circuit breakers, blown fuses, or catastrophic failure as soon as the transformers are energized!

Figure 14.22 shows the diagram for a large utility power transformer[5] equipped with a number of current transformers permanently installed in the bushings (the points at which power conductors penetrate the steel casing of the power transformer unit). Note that each CT has a solid black square marking one side of its secondary winding (the side also labeled "X1").

[5]This particular transformer happens to be a *tap-changing* unit, designed to provide a number of ratio increments useful for adjusting voltages in a power distribution system. Its typical primary voltage is 115 kV and its typical secondary voltage is 12.5 kV. If the secondary voltage happens to sag due to a heavy-load conditions, the transformer's tap setting may be manually adjusted to output a slightly greater secondary voltage (i.e., a lesser step-down ratio). This is how electric power distribution utilities manage to keep voltages to customers relatively stable despite ongoing changes in load conditions.

Figure 14.22: Tap-changing transformer with polarity dots marked

An example of the importance of polarity marks to the connection of instrument transformers may be seen in Figure 14.23, where a pair of current transformers are connected in parallel to drive a common instrument which is supposed to measure the *difference* in current entering and exiting a load.

Figure 14.23: Differential CT connection

Properly connected as shown above, the meter in the center of the circuit registers only the *difference* in current output by the two current transformers. If current into the load is precisely equal to current out of the load (which it should be), and the two CTs are precisely matched in their turns ratio, the meter will receive zero net current. If, however, a ground fault develops within the load causing more current to enter than to exit it, the imbalance in CT currents will be registered by the meter and thus indicate a fault condition in the load.

Suppose, though, that a technician mistakenly connected one of these CT units backwards. Examining the resulting circuit in Figure 14.24 shows that the meter now senses the *sum* of the line currents rather than the *difference* as it should.

Figure 14.24: Improper differential CT connection

This will cause the meter to falsely indicate a current imbalance through the load circuit when none exists.

## 14.4 Instrument transformer burdens and safety

Voltage transformers (PTs or VTs) tend to behave as voltage sources to the instruments they drive: the signal output by a VT is supposed to be a proportional representation of the power system's voltage. Conversely, current transformers (CTs) tend to behave as current sources to the instruments they drive: the signal output by a CT is supposed to be a proportional representation of the power system's current.

In order for an instrument transformer to function as an accurate sensing device, it must not be burdened with delivering any substantial amount of power to a load. In order to not deliver power, a VT should drive zero current to any instrument(s) connected to it, and a CT should drive zero voltage to any instrument(s) connected to it. Ideally, instruments connecting to VTs function as voltmeters, which draw very little current. Ideally, instruments connecting to CTs function as ammeters, behaving very nearly as a short-circuit and therefore dropping little voltage.

The goal of delivering zero power to any instrument is difficult to achieve in real life. Every voltmeter does indeed draw some current, however slight. Every ammeter does drop some voltage, however slight. The amount of apparent power drawn from any instrument transformer is appropriately called *burden*, and like all expressions of apparent power is measured in units of volt-amps. The greater this burden, the more the instrument transformer's signal will "sag" (decrease from loading). Therefore, minimizing burden is a matter of sensing accuracy for power system instrumentation.

Burden for an instrument transformer may be calculated using the familiar apparent power formulae $S = \frac{V^2}{Z}$ and $S = I^2 Z$:

$$\text{VT burden} = \frac{V_{signal}^2}{Z_{instrument}}$$

$$\text{CT burden} = \frac{(I_{signal}^2)}{(Z_{instrument})}$$

Burden for any device or circuit connected to an instrument transformer may be expressed as an impedance value ($Z$) in ohms per the ANSI standard, or as an apparent power value ($S$) in volt-amps at maximum rated current per the IEC standard. Similarly, instrument transformers themselves are usually rated for the amount of burden they may source and still perform within a certain accuracy tolerance (e.g., ± 1% at a burden of 2 VA).

Following the same reasoning of VTs behaving as voltage sources and CTs behaving as current sources, *a VT's secondary winding should never be short-circuited and a CT's secondary winding should never be open-circuited!* Short-circuiting a VT's secondary winding may result in a dangerous amount of current developing in the circuit. Open-circuiting a CT's secondary winding may result in a dangerous amount of voltage[6] developing between the secondary terminals.

In either case – short-circuiting a VT or open-circuiting a CT – such an event places a near-infinite burden on the transformer, and it will "do all it can" to try to maintain its output signal value.

[6]The hazards of an open-circuited CT can be spectacular. Power electricians have personally witnessed huge arcs develop across the opened terminals in a CT circuit! This safety tip is not one to be lightly regarded.

## 14.5 Instrument transformer test switches

Connections made between instrument transformers and receiving instruments such as panel-mounted meters and relays must be occasionally broken in order to perform tests and other maintenance functions. An accessory often seen in power instrument panels is a *test switch bank*, consisting of a series of knife switches. A photograph of a test switch bank manufactured by ABB is seen in Figure 14.25.

Figure 14.25: Installed instrument transformer test switches

Some of these knife switches serve to disconnect voltage transformers (VTs) from receiving instruments mounted on this relay panel, while other knife switches in the same bank serve to disconnect current transformers (CTs) from receiving instruments mounted on the same panel.

For added security, covers may be installed on the switch bank to prevent accidental operation or electrical contact. Some test switch covers are even lockable by padlock for an added measure of access prevention.

Test switches used to disconnect voltage transformers (VTs) from voltage-sensing instruments are nothing more than simple single-pole, single-throw (SPST) knife switches, as shown in Figure 14.26.

Figure 14.26: Application of a VT test switch

There is no danger in open-circuiting a voltage transformer circuit, and so nothing special is needed to disconnect a VT from a receiving instrument.

A series of photographs showing the operation of one of these knife switches appears in Figure 14.27, from closed (in-service) on the left to open (disconnected) on the right.

(a) Closed (b) Halfway open (c) Open

Figure 14.27: VT test switch operation

Test switches used to disconnect current transformers (CTs) from current-sensing instruments, however, must be specially designed to avoid opening the CT circuit when disconnecting, due to the high-voltage danger posed by open-circuited CT secondary windings. Thus, CT test switches are designed to place a *short-circuit* across the CT's output before opening the connection to the current-measuring device. This is done through the use of a special *make-before-break* knife switch shown in Figure 14.28.

Figure 14.28: Application of a make-before-break CT test switch

A series of photographs showing the operation of a make-before break knife switch appears in Figure 14.29, from closed (in-service) on the left to shorted (disconnected) on the right:

(a) In-service (b) Halfway open (c) Shorted

Figure 14.29: CT test switch operation

The shorting action takes place at a spring-steel leaf contacting the moving knife blade at a cam cut near the hinge. Note how the leaf is contacting the cam of the knife in the right-hand and middle photographs, but not in the left-hand photograph. This metal leaf joins with the base of the knife switch adjacent to the right (the other pole of the CT circuit), forming the short-circuit between CT terminals necessary to prevent arcing when the knife switch opens the circuit to the receiving instrument.

A step-by-step sequence of illustrations in Figure 14.30 shows how this shorting spring works to prevent the CT circuit from opening when the first switch is opened.

Figure 14.30: CT test switch make-before-break operation

It is typical that the non-shorting switch in a CT test switch pair be equipped with a "test jack" allowing the insertion of an additional ammeter in the circuit for measurement of the CT's signal.

This test jack consists of a pair of spring-steel leafs contacting each other in the middle of the knife switch's span. When that knife switch is in the open position, the metal leafs continue to provide continuity past the open knife switch. However, when a special ammeter adapter plug is forced between the leafs, spreading them apart, the circuit breaks and the current must flow through the two prongs of the test plug (and to the test ammeter connected to that plug).

A step-by-step sequence of illustrations shows how a test jack maintains continuity across an opened knife switch, and then allows the insertion of a test probe and ammeter, without ever breaking the CT circuit:

Figure 14.31: CT test switch make-before-break operation with a test jack

This close-up photograph shows a closed CT test switch equipped with a test jack, the jack's spring leafs visible as a pair of "hoop" shaped structures flanking the blade of the middle knife switch:

Figure 14.32: Photo of a CT test switch with a test jack

# Chapter 15

# Signal Characterization

## Objectives

Upon completion of this chapter, the student shall be able to:

Mathematics is full of complementary principles and symmetry. Perhaps nowhere is this more evident than with *inverse functions*: functions that "un-do" one another when put together. A few examples of inverse functions are shown in the following table:

| $f(x)$ | $f^{-1}(x)$ |
|---|---|
| Addition | Subtraction |
| Multiplication | Division |
| Power | Root |
| Exponential | Logarithm |
| Derivative | Integral |

Inverse functions are vital to master to be able to manipulate algebraic (literal) expressions. For example, to solve for time ($t$) in this exponential formula, the fact that the natural logarithm function is the inverse function for the exponential $e^x$ must be known. This is the only way to get $t$ isolated by itself on one side of the equals sign:

$$V = 12e^{-t}$$

*Divide both sides by 12*

$$\frac{V}{12} = e^{-t}$$

*Take the natural logarithm of both sides*

$$\ln\left(\frac{V}{12}\right) = \ln\left(e^{-t}\right)$$

*The natural logarithm "cancels out" the exponential*

$$\ln\left(\frac{V}{12}\right) = -t$$

*Multiply both sides by negative one*

$$-\ln\left(\frac{V}{12}\right) = t$$

In industry there exist a great many practical problems where inverse functions play a similar role. Just as inverse functions are useful for manipulating literal expressions in algebra, they are also useful in inferring measurements of things that cannot be directly measured. Many continuous industrial measurements are *inferential* in nature, meaning that some other variable is actually measured in order to quantify the variable of interest. More often than not, the relationship between the primary variable and the inferred variable is nonlinear, necessitating some form of mathematical processing to complete the inferential measurement.

Take for instance the problem of measuring fluid flow through a pipe. To the layperson, this may seem to be a trivial problem. However there is no practical way to *directly* and continuously measure the flow rate of a fluid, especially when the fluid in question cannot be allowed to become exposed to the atmosphere (e.g., when the liquid or gas in question is toxic, flammable, under high pressure, or any combination thereof).

One standard way to measure the flow rate of a fluid through a pipe is to intentionally place a restriction in the path of the fluid, and measure the pressure drop across that restriction. The most common form of intentional restriction used for this purpose is a thin plate of metal with a hole precisely machined in the center, called an *orifice plate*.

A side view of the orifice plate assembly and pressure-measuring instrument looks like this:

Figure 15.1: An orifice plate with d/p transmitter used for measuring flow

This approach should make intuitive sense: the faster the flow rate of the fluid, the greater the pressure difference developed across the orifice. The actual physics of this process has to do with energy exchanging between potential and kinetic forms, but that is incidental to this discussion. The mathematically interesting characteristic of this flow measurement technique is its nonlinearity. Pressure does not rise linearly with flow rate; rather, it increases with the *square* of the flow rate:

Figure 15.2: Differential pressure across a restriction as a function of flow through it

To write this as a proportionality, relate flow rate ($Q$) to pressure ($P$) as follows (the constant $k$ accounts for unit conversions and the geometries of the orifice plate and pipe):

$$P = kQ^2$$

This is a practical problem for us because the intent is to use pressure measurement ($P$) as an indirect (inferred) indication of flow rate ($Q$). If the two variables are not *directly* related to one another, one cannot be regarded as being directly representative of the other. To make this problem more clear to see, imagine a pressure gauge connected across the restriction, with the face of the gauge labeled in percent:

Figure 15.3: A flow gauge with a linear characteristic

Consider a pressure gauge such as the one shown above, registering 20 percent on a linear scale at some amount of flow through the pipe. What will happen if the flow rate through that pipe suddenly doubles? An operator or technician looking at the gauge *ought* to see a new reading of 40 percent, if indeed the gauge is supposed to indicate flow rate. However, this will not happen. Since the pressure dropped across the orifice in the pipe increases with the square of flow rate, a doubling of flow rate will actually cause the pressure gauge reading to *quadruple!* In other words, it will go from reading 20% to reading 80%, which is definitely not a linear function of the flow increase.

A couple of simple solutions exist for addressing this problem. One is to re-label the pressure gauge with a "square root" scale. Examine this photograph of a 3-15 psig receiver gauge:

Figure 15.4: A flow gauge with a square root characteristic

Now, a doubling of fluid flow rate still results in a quadrupling of needle motion, but due to the nonlinear (inner) scale on this gauge this needle motion translates into a simple doubling of indicated flow, which is precisely what is needed for this to function as an accurate flow indicator.

If the differential pressure instrument outputs a 4-20 mA analog electronic signal instead of a 3-15 psig pneumatic signal, the same "nonlinear scale" treatment may be applied to any current meter to achieve the same result:

Figure 15.5: A flow indicator with a square root characteristic

Another simple solution is to use a *nonlinear manometer*, with a curved viewing tube[1]:

Figure 15.6: A manometer with a nonlinear characteristic

The scale positioned alongside the curved viewing tube will be linear, with equal spacings between division marks along its entire length. The vertical height of the liquid column translates pressure into varying degrees of movement along the axis of the tube by the tube's curvature. Literally, any

[1] This solution works best for measuring the flow rate of gases, not liquids, since the manometer obviously must use a liquid of its own to indicate pressure, and mixing or other interference between the process liquid and the manometer liquid could be problematic.

inverse function desired may be "encoded" into this manometer by fashioning the viewing tube into the desired (custom) shape without any need to print a nonlinear scale.

Shown here is a photograph of an actual curved-tube manometer. This particular specimen does not have a scale reading in units of flow, but it certainly could if it had the correct curve for a square-root characterization:

Figure 15.7: An industrial manometer with a nonlinear characteristic

A more sophisticated solution to the "square root problem" is to use a computer to manipulate the signal coming from the differential pressure instrument so the characterized signal becomes a direct, linear representation of flow. In other words, the computer *square-roots* the pressure sensor's signal in order that the final signal becomes a direct representation of fluid flow rate:

Figure 15.8: Diagram showing use of an electronic characterizer

Both solutions achieve their goal by mathematically "un-doing" the nonlinear (square) function intrinsic to the physics of the orifice plate with a complementary (inverse) function. This intentional compounding of inverse functions is sometimes called *linearization*, because it has the overall effect of making the output of the instrument system a direct proportion of the input:

$$\text{Output} = k(\text{Input})$$

Fluid flow rate measurement in pipes is not the only application where nonlinearities complicate the task of measurement. Several other applications exhibit similar challenges:

- Liquid flow measurement in open channels (over weirs)
- Liquid level measurement in non-cylindrical vessels
- Temperature measurement by radiated energy
- Chemical composition measurement

The following sections will describe the mathematics behind each of these measurement applications.

## 15.1 Flow measurement in open channels

Measuring the flow rate of liquid through an open channel is not unlike measuring the flow rate of a liquid through a closed pipe: one of the more common methods for doing so is to place a restriction

in the path of the liquid flow and then measure the "pressure" dropped across that restriction. The easiest way to do this is to install a low "dam" in the middle of the channel, then measure the height of the liquid upstream of the dam as a way to infer flow rate. This dam is technically referred to as a *weir*, and three styles of weir are commonly used:

Figure 15.9: Types of weirs

Another type of open-channel restriction used to measure liquid flow is called a *flume*. An illustration of a *Parshall flume* is shown here:

Figure 15.10: A Parshall flume

*Weirs* and *flumes* may be thought of being somewhat like "orifice plates" and "venturi tubes," respectively, for open-channel liquid flow. Like an orifice plate, a weir or a flume generates a differential pressure that varies with the flow rate through it. However, this is where the similarities end. Exposing the fluid stream to atmospheric pressure means the differential pressure caused by the flow rate manifests itself as a difference in liquid height at different points in the channel. Thus, weirs and flumes allow the indirect measurement of liquid flow by sensing liquid height. An interesting feature of weirs and flumes is that although they are nonlinear primary sensing elements, their nonlinearity is quite different from that of an orifice.

Note the following transfer functions for different weirs and flumes, relating the rate of liquid flow through the device ($Q$) to the level of liquid rise upstream of the device (called "head", or $H$):

$$Q = 2.48\left(\tan\frac{\theta}{2}\right)H^{\frac{5}{2}} \qquad \text{V-notch weir}$$

$$Q = 3.367LH^{\frac{3}{2}} \qquad \text{Cippoletti weir}$$

$$Q = 0.992H^{1.547} \qquad \text{3-inch wide throat Parshall flume}$$

$$Q = 3.07H^{1.53} \qquad \text{9-inch wide throat Parshall flume}$$

Where,
$Q$ = Volumetric flow rate (cubic feet per second – CFS)
$L$ = Width of notch crest or throat width (feet)
$\theta$ = V-notch angle (degrees)
$H$ = Head (feet)

It is important to note these functions provide answers for flow rate ($Q$) with head ($H$) being the independent variable. In other words, they will indicate how much liquid is flowing given a certain head. In the course of calibrating the head-measuring instruments that infer flow rate, however, it is important to know the inverse operation: how much head there will be for any given value of flow. For example, here is the solution for $H$ in the function for a Cippoletti weir:

$$Q = 3.367LH^{\frac{3}{2}}$$

*Dividing both sides of the equation by 3.367 and L:*

$$\frac{Q}{3.367L} = H^{\frac{3}{2}}$$

*Taking the 3/2 root of both sides:*

$$\sqrt[3/2]{\frac{Q}{3.367L}} = H$$

This in itself may be problematic, as some hand calculators do not have an $\sqrt[x]{y}$ function. In cases such as this, it is helpful to remember that a root is nothing more than an inverse power. Therefore, the final form of the equation could be rewritten using a 2/3 power instead of a 3/2 root:

$$\left(\frac{Q}{3.367L}\right)^{\frac{2}{3}} = H$$

## 15.2 Liquid volume measurement

When businesses use large vessels to store liquids, it is useful to know how much liquid is stored in each vessel. A variety of technologies exist to measure stored liquid. Hydrostatic pressure, radar, ultrasonic, and tape-and-float are just a few of the more common technologies:

Figure 15.11: Tank level measurement methods

These liquid measuring technologies share a common trait: they measure the quantity of liquid in the vessel by measuring liquid *height*. If the vessel in question has a constant cross-sectional area throughout its working height (e.g., a vertical cylinder), then liquid height will directly correspond to liquid volume. However, if the vessel in question does not have a constant cross-sectional area throughout its height, the relationship between liquid height and liquid volume will not be linear.

For example, there is a world of difference between the height/volume functions for a vertical cylinder versus a horizontal cylinder:

Figure 15.12: Volumes illustrated for vertical and horizontal cylindrical tanks

The volume function for a vertical cylinder is a simple matter of geometry – height ($h$) multiplied by the cylinder's cross-sectional area ($\pi r^2$):

$$V = \pi r^2 h$$

Calculating the volume of a horizontal cylinder as a function of liquid height ($h$) is a far more complicated matter, because the cross-sectional area is also a function of height. For this, apply calculus.

First, begin with the mathematical definition of a circle, then graphically represent a partial area of that circle as a series of differential elements:

Figure 15.13: Using differential slices to determine filled cylindrical tank volume

In this sketch, the circle is shown "filling" from left to right rather than from bottom to top. This is strictly out of mathematical convention, where the $x$ (horizontal) axis is the independent variable. No matter how the circle gets filled, the relationship of area ($A$) to fill distance ($h$) will be the same.

If $x^2 + y^2 = r^2$ (the mathematical definition of a circle), then the area of each rectangular "slice" comprising the accumulated area between $-r$ and $h - r$ is equal to $2y\,dx$. In other words, the total accumulated area between $-r$ and $h - r$ is:

$$A = \int_{-r}^{h-r} 2y\,dx$$

Now, writing $y$ in terms of $r$ and $x$ ($y = \sqrt{r^2 - x^2}$) and moving the constant "2" outside the integrand:

$$A = 2\int_{-r}^{h-r} \sqrt{r^2 - x^2}\,dx$$

Consulting a table of integrals, a solution is found for the general form:

$$\int \sqrt{a^2 - u^2}\,du = \frac{u}{2}\sqrt{a^2 - u^2} + \frac{a^2}{2}\sin^{-1}\left(\frac{u}{a}\right) + C$$

Applying this solution to the particular integral at hand:

$$A = 2\left[\frac{x}{2}\sqrt{r^2 - x^2} + \frac{r^2}{2}\sin^{-1}\left(\frac{x}{r}\right)\right]_{-r}^{h-r}$$

$$A = 2\left[\left(\frac{(h-r)}{2}\sqrt{r^2 - (h-r)^2} + \frac{r^2}{2}\sin^{-1}\frac{(h-r)}{r}\right) - \left(\frac{-r}{2}\sqrt{r^2 - (-r)^2} + \frac{r^2}{2}\sin^{-1}\frac{-r}{r}\right)\right]$$

$$A = 2\left[\left(\frac{(h-r)}{2}\sqrt{r^2 - (h^2 - 2hr + r^2)} + \frac{r^2}{2}\sin^{-1}\frac{(h-r)}{r}\right) - \left(\frac{-r}{2}\sqrt{0} + \frac{r^2}{2}\frac{-\pi}{2}\right)\right]$$

$$A = 2\left[\left(\frac{(h-r)}{2}\sqrt{2hr - h^2} + \frac{r^2}{2}\sin^{-1}\frac{(h-r)}{r}\right) - \left(\frac{-\pi r^2}{4}\right)\right]$$

$$A = \left[(h-r)\sqrt{2hr - h^2} + r^2\sin^{-1}\frac{(h-r)}{r} + \frac{\pi r^2}{2}\right]$$

Knowing that the stored liquid volume in the horizontal tank will be this area multiplied by the constant length ($L$) of the tank, the formula for volume is as follows:

$$V = L\left[(h-r)\sqrt{2hr - h^2} + r^2\sin^{-1}\frac{(h-r)}{r} + \frac{\pi r^2}{2}\right]$$

As shown, the result is far from simple. Any instrumentation system tasked with the inference of stored liquid volume by measurement of liquid height in a horizontal cylinder must somehow apply this formula on a continuous basis. This is a prime example of how digital computer technology is essential to certain continuous measurement applications!

Spherical vessels, such as those used to store liquefied natural gas (LNG) and butane, present a similar challenge. The height/volume function is nonlinear because the cross-sectional area of the vessel changes with height.

Calculus provides a way to derive an equation solving for stored volume ($V$) with height ($h$) as the independent variable. Begin in a similar manner to the last problem with the mathematical definition of a circle, except now consider the filling of a sphere with a series of thin, circular disks:

Figure 15.14: Using differential disks to determine filled spherical tank volume

If $x^2 + y^2 = r^2$ (the mathematical definition of a circle), then the volume of each circular disk comprising the accumulated volume between $-r$ and $h - r$ is equal to $\pi y^2\, dx$. In other words, the total accumulated area between $-r$ and $h - r$ is:

$$V = \int_{-r}^{h-r} \pi y^2\, dx$$

Now, writing $y$ in terms of $r$ and $x$ ($y = \sqrt{r^2 - x^2}$) and moving the constant $\pi$ outside the integrand:

$$V = \pi \int_{-r}^{h-r} \left(\sqrt{r^2 - x^2}\right)^2\, dx$$

The square and the square-root cancel one another, leaving a fairly simple integrand:

$$V = \pi \int_{-r}^{h-r} r^2 - x^2\, dx$$

This may be written as the difference of two integrals:

$$V = \left(\pi \int_{-r}^{h-r} r^2\, dx\right) - \left(\pi \int_{-r}^{h-r} x^2\, dx\right)$$

Since $r$ is a constant, the left-hand integral is simply $\pi r^2 x$. The right-hand integral is solvable by the power rule:

$$V = \pi r^2 \left[x\right]_{-r}^{h-r} - \left[\pi \frac{x^3}{3}\right]_{-r}^{h-r}$$

$$V = \pi r^2 \left[(h-r) - (-r)\right] - \left[\pi \frac{(h-r)^3}{3} - \pi \frac{(-r)^3}{3}\right]$$

$$V = \pi r^2 \left[h - r + r\right] - \frac{\pi}{3}\left[(h-r)^3 - (-r)^3\right]$$

$$V = \pi h r^2 - \frac{\pi}{3}\left[(h^3 - 2h^2 r + hr^2 - h^2 r + 2hr^2 - r^3) + r^3\right]$$

$$V = \pi h r^2 - \frac{\pi}{3}\left[h^3 - 3h^2 r + 3hr^2\right]$$

$$V = \pi h r^2 - \frac{\pi h^3}{3} + \pi h^2 r - \pi h r^2$$

$$V = -\frac{\pi h^3}{3} + \pi h^2 r$$

$$V = \pi h^2 r - \frac{\pi h^3}{3}$$

$$V = \pi h^2 (r - \frac{h}{3})$$

This function characterizes the inherent height/volume nonlinearity of a spherical vessel, allowing a height measurement to translate directly into a volume measurement. A characterizing function such as this is typically executed in a either a digital computer or controller connected to the level sensor or in a computer chip within the sensor device itself.

An interesting alternative to a formal equation for linearizing the level measurement signal is to use something called a *multi-segment characterizer* function, also implemented in a digital computer or controller. This is an example of what mathematicians call a *piecewise function*: a function made up of line segments. Multi-segment characterizer functions may be programmed to emulate virtually any continuous function, with reasonable accuracy:

Figure 15.15: Using piecewise linear characterizing functions to approximate a continuous function

The computer correlates the input signal (height measurement, $h$) to a point on this piecewise function, linearly interpolating between the nearest pair of programmed coordinate points. The number of points available for multi-point characterizers varies between ten and one hundred[2] depending on the desired accuracy and the available computing power.

Although true fans of math might blanch at the idea of approximating an inverse function for level measurement using a piecewise approach rather than simply implementing the correct continuous function, the multi-point characterizer technique does have certain practical advantages. For one, it is readily adaptable to any shape of vessel, no matter how strange. Take for instance this vessel, made of separate cylindrical sections welded together:

Figure 15.16: An example requiring use of piecewise linear characterization

[2]There is no theoretical limit to the number of points in a digital computer's characterizer function given sufficient processing power and memory. There is, however, a limit to the patience of the human programmer who must encode all the necessary $x, y$ data points defining this function. Most of the piecewise characterizing functions available in digital instrumentation systems provide 10 to 20 $(x, y)$ coordinate points to define the function. Fewer than 10 coordinate points risks excessive interpolation errors, and more than 20 would just be tedious to set up.

Here, the vessel's very own height/volume function is fundamentally piecewise, and so *nothing but* a piecewise characterizing function could possibly linearize the level measurement into a volume measurement!

Consider also the case of a spherical vessel with odd-shaped objects welded to the vessel walls, and/or inserted into the vessel's interior:

Figure 15.17: Another example requiring use of piecewise linear characterization

The volumetric space occupied by these structures will introduce all kinds of discontinuities into the transfer function, and so once again a continuous characterizing function cannot properly linearize the level signal into a volume measurement. Here, only a piecewise function will suffice.

To best generate the coordinate points for a proper multi-point characterizer function, one must collect data on the storage vessel in the form of a *strapping table*. This entails emptying the vessel completely, then filling it with measured quantities of liquid, one sample at a time, and taking level readings:

| Introduced liquid volume | Measured liquid level |
|---|---|
| 150 gallons | 2.46 feet |
| 300 gallons | 4.72 feet |
| 450 gallons | 5.8 feet |
| 600 gallons | (etc.) |
| 750 gallons | (etc.) |

Each of these paired numbers would constitute the coordinates to be programmed into the characterizer function computer by the instrument technician or engineer:

Figure 15.18: A plot of the characterizer function

Many "smart" level transmitter instruments possess enough computational power to perform the level-to-volume characterization directly, so as to transmit a signal corresponding directly to liquid volume rather than just liquid level. This eliminates the need for an external "level computer" to perform the necessary characterization. The following screenshot was taken from a personal computer running configuration software for a radar level transmitter[3], showing the strapping table data point fields where a technician or engineer would program the vessel's level-versus-volume piecewise function:

[3]The configuration software is Emerson's AMS, running on an engineering workstation in a DeltaV control system network. The radar level transmitter is a Rosemount model 3301 (guided-wave) unit.

Figure 15.19: A strapping table configuration for a radar level transmitter

This configuration window actually shows more than just a strapping table. It also shows the option of calculating volume for different vessel shapes (*vertical cylinder* is the option selected here) including horizontal cylinder and sphere. In order to use the strapping table option, the user would have to select "Strapping Table" from the list of Tank Types. Otherwise, the level transmitter's computer will attempt to calculate volume from an ideal tank shape.

## 15.3 Radiative temperature measurement

Temperature measurement devices may be classified into two broad types: *contact* and *non-contact*. Contact-type temperature sensors detect temperature by directly touching the material to be measured, and there are several varieties in this category. Non-contact temperature sensors work by detecting light emitted by hot objects.

Energy radiated in the form of electromagnetic waves (photons, or light) relates to object temperature by an equation known as the Stefan-Boltzmann equation, which tells us the rate of heat lost by radiant emission from a hot object is proportional to the *fourth power* of its absolute

temperature:

$$P = e\sigma AT^4$$

Where,
$P$ = Radiated energy power (watts)
$e$ = Emissivity factor (unitless)
$\sigma$ = Stefan-Boltzmann constant ($5.67 \times 10^{-8}$ W / m$^2 \cdot$ K$^4$)
$A$ = Surface area (square meters)
$T$ = Absolute temperature (Kelvin)
Solving for temperature ($T$) involves the use of the fourth root:

$$T = \sqrt[4]{\frac{P}{e\sigma A}}$$

Any optical temperature sensor measuring the emitted power ($P$) must characterize the power measurement using the above equation to arrive at an inferred temperature. This characterization is typically performed inside the temperature sensor by an integrated circuit.

## 15.4 Analytical measurements

A great many chemical composition measurements may be made indirectly by means of electricity, if those measurements are related to the concentration of ions (electrically charged molecules). Such measurements include:

- pH of an aqueous solution
- Oxygen concentration in air
- Ammonia concentration in air
- Lead concentration in water

The basic principle works like this: two different chemical samples are placed in close proximity to each other, separated only by an *ion-selective membrane* able to pass the ion of interest. As the ion activity attempts to reach equilibrium through the membrane, an electrical voltage is produced across that membrane. the relative activity of the ions on either side of the membrane may be inferred by measuring the voltage produced.

Not surprisingly, the function relating ion activity to the voltage generated is nonlinear. The standard equation describing the relationship between ionic activity on both sides of the membrane and the voltage produced is called the *Nernst equation*:

$$V = \frac{RT}{nF} \ln\left(\frac{a_1}{a_2}\right)$$

Where,
$V$ = Electrical voltage produced across membrane due to ion exchange (volts)
$R$ = Universal gas constant (8.315 J/mol·K)
$T$ = Absolute temperature (Kelvin)
$n$ = Number of electrons transferred per ion exchanged (unitless)
$F$ = Faraday constant (96,485 coulombs per mole)
$a_1$ = Activity of ion in measured sample

$a_2$ = Activity of ion in reference sample (on other side of membrane)

A practical application for this technology is in the measurement of oxygen concentration in the flue gas of a large industrial burner, such as what might be used to heat up water to generate steam. The measurement of oxygen concentration in the exhaust of a combustion heater (or boiler) is very important both for maximizing fuel efficiency and for minimizing pollution (specifically, the production of $NO_x$ molecules). Ideally, a burner's exhaust gas will contain no oxygen, having consumed it all in the process of combustion with a perfect stoichiometric mix of fuel and air. In practice, the exhaust gas of an efficiently-controlled burner will be somewhere near 2%, as opposed to the normal 21% of ambient air.

One way to measure the oxygen content of hot exhaust is to use a *high-temperature zirconium oxide* detector. This detector is made of a "sandwich" of platinum electrodes on either side of a solid zirconium oxide electrolyte. One side of this electrochemical cell is exposed to the exhaust gas (process), while the other side is exposed to heated air which serves as a reference:

Figure 15.20: Diagram of a zirconium oxide analytical element for detecting $O_2$ in exhaust

The electrical voltage generated by this "sandwich" of zirconium and platinum is sent to an electronic amplifier circuit, and then to a microcomputer which applies an inverse function to the measured voltage in order to arrive at an inferred measurement for oxygen concentration. This type of chemical analysis is called *potentiometric*, since it measures ("metric") based on an electrical voltage ("potential").

The Nernst equation is an interesting one to unravel, to solve for ion activity in the sample ($a_1$) given voltage ($V$):

$$V = \frac{RT}{nF} \ln \left( \frac{a_1}{a_2} \right)$$

*Multiplying both sides by nF:*

$$nFV = RT \ln\left(\frac{a_1}{a_2}\right)$$

*Dividing both sides by RT:*

$$\frac{nFV}{RT} = \ln\left(\frac{a_1}{a_2}\right)$$

*Applying the rule that the difference of logs is equal to the log of the quotient:*

$$\frac{nFV}{RT} = \ln a_1 - \ln a_2$$

*Adding* $\ln a_2$ *to both sides:*

$$\frac{nFV}{RT} + \ln a_2 = \ln a_1$$

*Making both sides of the equation a power of e:*

$$e^{\frac{nFV}{RT} + \ln a_2} = e^{\ln a_1}$$

*Canceling the natural log and exponential functions on the right-hand side:*

$$e^{\frac{nFV}{RT} + \ln a_2} = a_1$$

In most cases, the ionic activity of $a_2$ will be relatively constant, and so $\ln a_2$ will be relatively constant as well. With this in mind, the equation may be simplified further, using $k$ as a constant value:

*Substituting k for* $\ln a_2$*:*

$$e^{\frac{nFV}{RT} + k} = a_1$$

*Applying the rule that the sum of exponents is the product of powers:*

$$e^k e^{\frac{nFV}{RT}} = a_1$$

*If k is constant, then* $e^k$ *will be constant as well (calling the new constant C):*

$$Ce^{\frac{nFV}{RT}} = a_1$$

Analytical instruments based on potentiometry must evaluate this inverse function to "undo" the Nernst equation to arrive at an inferred measurement of ion activity in the sample given the small voltage produced by the sensing membrane. These instruments typically have temperature sensors as well built in to the sensing membrane assembly, since it is apparent that temperature ($T$) also plays a role in the generation of this voltage. Once again, this mathematical function is typically evaluated in a microprocessor.

# Chapter 16

# Final Control Elements

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Draw standard symbols used for final control elements
2. Explain how on/off valves differ from modulating valves
3. Draw the three inherent valve characteristic curves and explain how the characteristic changes when installed
4. Properly size a control valve
5. Explain the function of control valve accessories, including actuators, positioners, I/P converters, air sets, and position limit switches
6. Discuss causes of valve cavitation and methods of valve cavitation abatement
7. Describe methods of controlling the speed of DC and AC motors

The word "discrete" means *individual* or *distinct*. In engineering, a "discrete" variable or measurement refers to a true-or-false condition. Thus, a discrete control element is one that has but a limited number of states (usually two: on and off). In the case of valves, this means a valve designed to operate either in "open" mode or "closed" mode, not in-between. Some control systems refer to discrete outputs as "digital" or "binary" outputs.

A "continuous" control element is one that can take any state between a lower limit value and an upper limit value. In the case of valves, this means a valve designed to modulate continuously to any position between "open" mode or "closed" mode. These are typically referred to by control system manufacturer's as analog outputs.

This chapter introduces common discrete and continuous output devices that may be encountered in the world of instrumentation and control.

## 16.1 On/off valves

An on/off valve is the fluid equivalent of an electrical switch: a device that either allows unimpeded flow or acts to prevent flow altogether. These valves are often used for routing process fluid to different locations, starting and stopping batch processes, and engaging automated safety (shutdown) functions.

Valve styles commonly used for on/off service include ball, plug, butterfly (or disk), gate, and globe. Large on/off valves are generally of such a design that the full-open position provides a nearly unimpeded path for fluid to travel through. Ball, plug[1], and gate valves provide just this characteristic:

[1] A *plug* valve is very much like a ball valve, the difference being the shape of the rotating element. Rather than a spherical ball, the plug valve uses a truncated cone as the rotary element, a slot cut through the cone serving as the passageway for fluid. The conical shape of a plug valve's rotating element allows it to wedge tightly into the "closed" (shut) position for exceptional sealing.

Figure 16.1: Valve cross-section views

## 16.2 Solenoid valves

A very common form of on/off valve used for pneumatic and hydraulic systems alike is the *solenoid valve*. A "solenoid" is nothing more than a coil of wire designed to produce a magnetic field when energized. Solenoid actuators work by attracting a movable iron *armature* into the center of the solenoid coil when energized, the force of this attraction working to actuate a small valve mechanism.

Solenoid-actuated valves are usually classified according to the number of ports ("ways"). A simple on/off solenoid valve controlling flow into one port and out of another port is called a *2-way* valve. Another style of solenoid valve, where flow is directed in one path or to another path – much like a single-pole double-throw (SPDT) electrical switch – is called a *3-way* valve because it has

three fluid ports.

### 16.2.1 2-way solenoid valves

Fluid power valve symbols for solenoid valve symbols consist of "boxes" representing flow paths and directions between ports in each of the valve's states. Like electrical switches, these valve symbols are always drawn in their "normal" (de-energized) state, where the return spring's action determines the valve position:

Figure 16.2: 2-way solenoid valve symbols used in fluid power diagrams

A good way to make sense of these "box" valve symbols is to imagine the boxes sliding back and forth as the actuating elements work. For example, the two boxes in a normally-closed solenoid valve symbol may be thought of as pushed to one side by the spring (in its de-energized state), and pushed the other direction by the solenoid's force (when energized). Here, the color grey de-emphasizes the unselected box in each of the valve's two states:

Figure 16.3: Fluid power solenoid valve symbols showing effects of deactivation/activation

Just like electrical switches in schematic diagrams, fluid control valve symbols are always drawn in their "normal" (resting) states. For example, a normally-closed valve will always be drawn so that the box with the blocked ports aligns with the tubes leading to and from the valve. The above illustration shows a "dramatized" symbol, highlighting the valve's action by color and by re-positioning the boxes. In a fluid control schematic, it is left to the reader to visualize the valve symbol boxes moving to and fro, determining the flow path of fluid through the valve.

Unlike electrical switches, of course, the terms *open* and *closed* have opposite meanings for valves. An "open" electrical switch constitutes a break in the circuit, ensuring no current; an "open" valve, by contrast, freely allows fluid flow through it. A "closed" electrical switch has continuity, allowing current through it; a "closed" valve, on the other hand, shuts off fluid flow.

The arrow inside a solenoid valve symbol actually denotes a preferred direction of flow. Most solenoid valves use a "globe" or "poppet" style of valve element, where a metal plug covers up a hole (called the "seat"). Process fluid pressure should be applied to the valve in such a way that the pressure difference tends to hold the solenoid valve in its "normal" position (the same position as driven by the return spring). Otherwise[2], enough fluid pressure might override the return spring's action, preventing the valve from achieving its "normal" state when de-energized. Thus, the label "2-way" does not refer to two directions of flow as one might assume, but rather two *ports* on the valve for fluid to travel through.

Some solenoid valves are designed in such a way that the direction of fluid flow through them is irrelevant. In such valves, the arrow symbols will be double-headed (one head at each end, pointing in opposite directions) to show the possibility of flow in either direction.

[2]One could argue that enough fluid pressure could override the solenoid's energized state as well, so why choose to have the fluid pressure act in the direction of helping the return spring? The answer to this (very good) question is that the solenoid's energized force exceeds that of the return spring. This is immediately obvious on first inspection, as the solenoid *must* be stronger than the return spring or else the solenoid valve would never actuate! Thus, the spring is the weaker of the two forces, and thus it makes perfect sense why the valve should be used in such a way that the process pressure helps the spring: the solenoid's force has the best chance of overcoming the force on the plug produced by process pressure, so those two forces should be placed in opposition, while the return spring's force should work *with* (not against) the process pressure.

Figure 16.4: Bidirectional fluid power solenoid valve symbol

### 16.2.2 3-way solenoid valves

3-way solenoid valves have three ports for fluid, and like 2-way valves may be referred to either as *normally-open* and *normally-closed*. Ports on a pneumatic 3-way valve are commonly labeled with the letters "P," "E," and "C," representing *Pressure* (compressed air supply), *Exhaust* (vent to atmosphere), and *Cylinder* (the actuating mechanism), respectively. Alternatively, the cylinder port may be labeled "A" (for *actuator*) instead of "C". If the solenoid valve is intended for use in a hydraulic (liquid) system, the letter "T" is customarily used to identify the return port rather than "E" (*Tank* rather than *Exhaust*):

Figure 16.5: 3-way solenoid valve symbols used in fluid power diagrams

Alternatively, the numbers 1, 2, and 3 may be used to label the same ports. However, the numbers do not consistently refer to pressure source (P) and exhaust (E) ports, but rather to the 3-way valve's "normal" versus "actuated" statuses. A 3-way valve will pass fluid between ports 1 and 3 in its "normal" (de-energized) state, and pass fluid between ports 1 and 2 in its energized state. The following table shows the correspondence between port numbers and port letters for both styles of 3-way solenoid valve:

| **Valve type** | **Pressure port** (P) | **Exhaust port** (E) | **Cylinder port** (C) |
|---|---|---|---|
| Normally-closed | 2 | 3 | 1 |
| Normally-open | 3 | 2 | 1 |

Another way to think of this labeling is to consider port 1 the *common*, port 2 the *normally-closed*, and port 3 the *normally-open*. Again, bear in mind that the words "open" and "closed" do not mean the same for fluid valves as they do for electrical switches. A "normally-open" port on the valve permits fluid flow in its "normal" state, whereas a "normally-open" switch contact *prevents* electric current flow in its "normal" state.

As with 2-way solenoid valves, the arrows denote preferred direction of fluid flow. Bidirectional 3-way valves will be drawn with double-headed arrows (pointing both directions).

A different symbology is used in loop diagrams and P&IDs than that found in fluid power diagrams – one more resembling general instrumentation (ISA) valve symbols:

Figure 16.6: 2-way and 3-way solenoid valve symbols used in instrumentation diagrams

Regrettably, these symbols are not nearly as descriptive as those used in fluid power diagrams. In order to show directions of flow (especially for 3-way valves), one must add arrows showing "normal" (de-energized, $DE$) flow directions:

Figure 16.7: 3-way instrumentation solenoid valve symbols with deenergized arrows

Alternatively, a *pair* of arrows shows the directions of flow in both energized ($E$) and de-energized ($D$) states:

Figure 16.8: 3-way instrumentation solenoid valve symbols with energized and deenergized arrows

Photographs of an actual 3-way solenoid valve (this one manufactured by ASCO) appear here:

Figure 16.9: A 3-way solenoid valve by ASCO

A view of the nameplate for this particular solenoid valve reveals some of its ratings and characteristics:

Figure 16.10: A 3-way solenoid valve nameplate

### 16.2.3 Normal energization states

Solenoid valves may be used in such a way that they spend most of their time de-energized, energizing only for brief periods of time when some special function is required. Alternatively, solenoids may be maintained in an energized state, and deenergized to perform their design function. The choice to use a solenoid's energized or deenergized state to perform a specific function is left to the system designer, but nevertheless it is important for all maintenance personnel to know in order to perform work on a solenoid-controlled system.

Take the following segment of a P&ID for a steam turbine-driven pump control system for example, where a pair of 3-way solenoid valves control instrument air pressure[3] to a piston-actuated steam valve to start the turbine in the event that an electric motor-driven pump happens to fail:

Figure 16.11: Example P&ID showing 3-way solenoid valves used in control

If *either*[4] of the two solenoid valves de-energizes, instrument air pressure will vent from the top of the piston actuator to atmosphere, causing the steam valve to "fail" to the full-open position and send steam to the turbine. This much is evident from the curved arrows showing air flowing to the "Vent" ports in a deenergized (DE) condition. An additional valve (PY-590) ensures the piston actuator's motion by simultaneously applying air pressure to the bottom of the actuator if ever air is vented from the top. As an additional feature, the left-hand solenoid valve (SOV-590A) has a

[3]The letters "IAS" refer to *instrument air supply*.

[4]This solenoid valve arrangement would be designated *1oo2* from the perspective of starting the turbine, since only one out of the two solenoids needs to trip in order to initiate the turbine start-up.

manual "Reset" lever on it, symbolized by the letter "R" inside a diamond outline.

## 16.3 Control valves

One of the most common final control elements in industrial control systems is the *control valve*. A control valve works to restrict the flow of fluid through a pipe at the command of an automated signal, such as the signal from a loop controller or logic device (such as a PLC). Some control valve designs are intended for discrete (on/off) control of fluid flow, while others are designed to *modulate* or *throttle* fluid flow somewhere between fully open and fully closed (shut), inclusive. The electrical equivalent of an on/off valve is a switch, while the electrical equivalent of a throttling valve is a variable resistor.

Control valves are comprised of two major parts: the *valve body*, which contains all the mechanical components necessary to influence fluid flow; and the *valve actuator*, which provides the mechanical power necessary to move the components within the valve body. Often times, the major difference between an on/off control valve and a throttling control valve is the type of actuator applied to the valve[5]: on/off actuators need only position a valve mechanism two one of two extreme positions (fully open or fully closed). Throttling actuators must be able to accurately position a valve mechanism anywhere between those extremes.

Within a control valve body, the specific components performing the work of throttling (or completely shutting off) of fluid flow are collectively referred to as the valve *trim*. For each major type of control valve, there are usually many variations of trim design. The choice of valve type, and of specific trim for any type of valve, is a decision dictated by the type of fluid being controlled, the nature of the control action (on/off versus throttling), the process conditions (expected flow rate, temperature, pressures, etc.), and economics.

### 16.3.1 Sliding-stem valves

A *sliding-stem* valve body is one that actuates with a linear motion. Some examples of sliding-stem valve body designs are shown here:

[5] To be honest, there are some valve body designs that work far better in on/off service (e.g., ball valves and plug valves) while other designs do a better job at throttling (e.g., double-ported globe valves). Many valve designs, however, may be pressed into either type of service merely by attaching the appropriate actuator.

Figure 16.12: Cross-sections of various sliding stem valves

Most sliding-stem control valves are *direct acting*, which means the valve opens up wider as the stem is drawn out of the body. Conversely, a direct-acting valve shuts off (closes) when the stem is pushed into the body. Of course, a *reverse-acting* valve body would behave just the opposite: opening up as the stem is pushed in and closing off as the stem is drawn out.

### Globe valves

Globe valves restrict the flow of fluid by altering the distance between a movable plug and a stationary seat (in some cases, a pair of plugs and matching seats). Fluid flows through a hole in the center of the seat, and is more or less restricted by how close the plug is to that hole. The globe valve design is one of the most popular sliding-stem valve designs used in throttling service. A photograph of a small (2 inch) globe valve body appears here:

Figure 16.13: A globe valve

A set of three photographs showing a cut-away Masoneilan model 21000 globe valve body illustrates just how the moving plug and stationary seat work together to throttle flow in a direct-acting globe valve. The left-hand photo shows the valve body in the fully closed position, while the middle photo shows the valve half-open, and the right-hand photo shows the valve fully open:

(a) Closed (b) Halfway open (c) Open

Figure 16.14: A globe valve with the plug in various positions

As can be seen from these photographs, the valve plug is guided by the stem so it always lines up with the centerline of the seat. For this reason, this particular style of globe valve is called a *stem-guided* globe valve.

A variation on the stem-guided globe valve design is the *needle valve*, where the plug is extremely small in diameter and usually fits well into the seat hole rather than merely sitting on top of it. Needle valves are very common as manually-actuated valves used to control low flow rates of air or oil. A set of three photographs shows a needle valve in the fully-closed, mid-open, and fully-open positions (left-to-right):

(a) Closed (b) Halfway open (c) Open

Figure 16.15: A needle valve with the plug in various positions

Yet another variation on the globe valve is the *port-guided* valve, where the plug has an unusual shape that projects into the seat. Thus, the seat ring acts as a guide for the plug to keep the centerlines of the plug and seat always aligned, minimizing guiding stresses that would otherwise be placed on the stem. This means that the stem may be made smaller in diameter than if the valve trim were stem-guided, minimizing sliding friction and improving control behavior.

Figure 16.16: Cross-section of a port-guided globe valve

Some globe valves use a pair of plugs (on the same stem) and a matching pair of seats to throttle fluid flow. These are called *double-ported* globe valves. The purpose of a double-ported globe valve is to minimize the force applied to the stem by process fluid pressure across the plugs:

Figure 16.17: Cross-section of a double-ported globe valve

Differential pressure of the process fluid ($P_1 - P_2$) across a valve plug will generate a force parallel to the stem as described by the formula $F = PA$, with $A$ being the plug's effective area presented for the pressure to act upon. In a single-ported globe valve, there will only be one force generated by the process pressure. In a double-ported globe valve, there will be *two opposed* force vectors, one generated at the upper plug and another generated at the lower plug. If the plug areas are approximately equal, then the forces will likewise be approximately equal and therefore nearly cancel. This makes for a control valve that is easier to actuate (i.e., the stem position is less affected by process fluid pressures).

The following photograph shows a disassembled Fisher "A-body" double-ported globe valve, with the double plug plainly visible on the right:

Figure 16.18: A disassembled double-ported globe valve body (left) and plug (right)

While double-ported globe valves certainly enjoy the advantage of easier actuation compared to their single-ported cousins, they also suffer from a distinct disadvantage: the near impossibility of tight shut-off. With *two* plugs needing to come to simultaneous rest on *two* seats to achieve a fluid-tight seal, there is precious little room for error or dimensional instability. Even if a double-ported valve is prepared in a shop for the best shut-off possible[6], it may not completely shut off when installed due to dimensional changes caused by process fluid heating or cooling the valve stem and body. This is especially problematic when the stem is made of a different material than the body. Globe valve stems are commonly manufactured from stainless steel bar stock, while globe valve bodies are commonly cast of iron. Cold-formed stainless steel has a different coefficient of thermal expansion than hot-cast iron, which means the plugs will no longer simultaneously seat once the valve warms or cools much from the temperature it was at when it seated tightly.

A more modern version of the globe valve design uses a piston-shaped plug inside a surrounding *cage* with ports cast or machined into it. These *cage-guided* globe valves throttle flow by uncovering more or less of the port area in the surrounding cage as the plug moves up and down. The cage also serves to guide the plug so the stem need not be subjected to lateral forces as in a stem-guided valve design. A photograph of a cut-away control valve shows the appearance of the cage (in this case, with the plug in the fully closed position). Note the "T"-shaped ports in the cage, through which fluid flows as the plug moves up and out of the way:

[6]The standard preparatory technique is called *lapping*. To "lap" a valve plug and seat assembly, an abrasive paste known as *lapping compound* is applied to the valve plug(s) and seat(s) at the areas of mutual contact when the valve is disassembled. The valve mechanism is reassembled, and the stem is then rotated in a cyclic motion such that the plug(s) grind into the seat(s), creating a matched fit. The precision of this fit may be checked by disassembling the valve, cleaning off all remaining lapping compound, applying a metal-staining compound such as *Prussian blue*, then reassembling. The stem is rotated once more such that the plug(s) will rub against the seat(s), wearing through the applied stain. Upon disassembly, the worn stain may be inspected to reveal the extend of metal-to-metal contact between the plug(s) and the seat(s). If the contact area is deemed insufficient, the lapping process may be repeated.

Figure 16.19: A cutaway view of a cage-guided globe valve

An advantage of the cage-guided design is that the valve's flowing characteristics may be easily altered just by replacing the cage with another having different size or shape of holes. Many different cage styles are available for certain plug (piston) sizes, which means the plug need not be replaced while changing the cage. This is decidedly more convenient than the plug change necessary for changing characteristics of stem-guided or port-guided globe valve designs.

Cage-guided globe valves are available with both *balanced* and *unbalanced* plugs. A balanced plug has one or more ports drilled from top to bottom, allowing fluid pressure to equalize on both sides of the plug. This helps minimize the forces acting on the plug which must be overcome by the actuator:

Figure 16.20: Cross-section of cage-guided globe valves: unbalanced (left), balanced (right)

Unbalanced plugs generate a force equal to the product of the differential pressure across the plug and the plug's area ($F = PA$), which may be quite substantial in some applications. Balanced

plugs do not generate this same force because they equalize the pressure on both sides of the plug, however, they exhibit the disadvantage of one more leak path when the valve is in the fully closed position (through the balancing ports, past the piston ring, and out the cage ports):

Figure 16.21: Leak paths of cage-guided globe valves: unbalanced (left), balanced (right)

### Gate valves

Gate valves work by inserting a dam ("gate") into the path of the flow to restrict it, in a manner similar to the action of a sliding door. Gate valves are more often used for on/off control than for throttling.

The following set of photographs shows a hand-operated gate valve (cut away and painted for use as an instructional tool) in three different positions, from full closed to full open (left to right):

(a) Closed (b) Halfway open (c) Open

Figure 16.22: A gate valve with the gate in various positions

### Diaphragm valves

Diaphragm valves use a flexible sheet pressed close to the edge of a solid dam to narrow the flow path for fluid. These valves are well suited for flows containing solid particulate matter such as slurries, although precise throttling may be difficult to achieve due to the elasticity of the diaphragm. The next photograph shows a diaphragm valve actuated by an electric motor, used to control the flow of treated sewage:

Figure 16.23: A motor-operated diaphragm valve

### 16.3.2 Rotary-stem valves

A different strategy for controlling the flow of fluid is to insert a rotary element into the flow path. Instead of sliding a stem into and out of the valve body to actuate a throttling mechanism, rotary valves rely on the rotation of a shaft to actuate the trim. An important advantage of rotary control valves over sliding-stem designs such as the globe valve and diaphragm valve is a virtually obstructionless path for fluid when the valve is wide-open[7].

[7] Of course, gate valves also offer obstructionless flow when wide-open, but their poor throttling characteristics give most rotary valve designs the overall advantage.

Figure 16.24: Types of rotary-operated valves

### Ball valves

In the ball valve design, a spherical ball with a passageway cut through the center rotates to allow fluid more or less access to the passageway. When the passageway is parallel to the direction of fluid motion, the valve is wide open; when the passageway is aligned perpendicular to the direction of fluid motion, the valve is fully shut (closed).

The following set of photographs shows a hand-operated ball valve in three different positions, from nearly full closed to nearly full open (left to right):

(a) Closed (b) Halfway open (c) Open

Figure 16.25: A ball valve with the ball in various positions

Simple ball valves with full-sized bores in the rotating ball are generally better suited for on/off service than for throttling (partially-open) service. A better design of ball valve for throttling service is the *characterized* or *segmented* ball valve, shown in various stages of opening in the following set of photographs:

(a) Closed (b) Halfway open (c) Open

Figure 16.26: A characterized ball valve with the ball in various positions

The V-shaped notch cut into the opening lip of the ball provides a narrower area for fluid flow at low opening angles, providing more precise flow control than a plain-bore ball valve.

### Butterfly valves

Butterfly valves are quite simple to understand: the "butterfly" element is a disk that rotates perpendicular to the path of fluid flow. When parallel to the axis of flow, the disk presents minimal obstruction; when perpendicular to the axis, the disk completely blocks any flow. Fluid-tight shut-off is difficult to obtain in the classic butterfly design unless the seating area is lined with a soft (elastic) material.

### Disk valves

Disk valves (often referred to as *eccentric disk valves*, or as *high-performance butterfly valves*) are a variation on the butterfly design intended to improve seat shut-off. The disk's center is offset from the shaft centerline, causing it to approach the seat with a "cam" action that results in high seating pressure. Thus, tight shut-off of flow is possible even when using metal seats and disks.

The following photograph shows the body of a Fisher E-plug control valve, with the disk in a partially-open position:

Figure 16.27: A high-performance butterfly valve with the disk partially open

### 16.3.3 Valve packing

Regardless of valve type, all stem-actuated control valves require some form of seal allowing motion of the stem from some external device (an *actuator*) while sealing process fluid so no leaks occur between the moving stem and the body of the valve. The general term for this sealing mechanism is *packing*.

This mechanical feature is not unlike the *stuffing box* used to seal seawater from entering a boat or ship at the point where the propeller shaft penetrates the hull:

Figure 16.28: Use of a stuffing box to seal around a shaft

The fundamental problem is the same for the ship as it is for the control valve: how to allow a moving shaft to pass through what needs to be an impenetrable barrier to some fluid (in the case

of the ship's hull, the fluid is seawater). The solution is to wrap the shaft in a flexible material that maintains a close fit to the shaft without binding its motion. A traditional packing material for ship propeller shafts is *flax*. Some form of lubrication is usually provided so this packing material does not impose excessive friction on the shaft's motion[8].

Modern marine stuffing boxes use advanced materials such as Teflon (PTFE) or graphite instead of flax, which wear longer and leak less water. In the world of control valves, the traditional packing material used to be asbestos (shaped into rings or ropes, much like flax used to be shaped for use in stuffing boxes), but is now commonly Teflon or graphite as well.

In the case of a ship's stuffing box, a little bit of water leakage is not a problem since all ships are equipped with bilge pumps to pump out collected water over time. However, leakage is simply unacceptable in many industrial control valve applications where *fugitive emissions* must be minimized. A "fugitive emission" is any unwanted escape of process substance into the surrounding environment, usually from leaks around pump and valve shafts. Special "environmental" packing sets are available for control valve applications where this is a concern.

Packing in a sliding-stem valve fits in a section of the valve body called the *bonnet*, shown in this simplified diagram of a single-ported, stem-guided globe valve:

[8] Some packing materials, most notably Teflon and graphite, tend to be *self-lubricating*.

Figure 16.29: Cross-section diagram of a globe valve illustrating the packing and associated parts

Here, the packing material takes the form of several concentric rings, stacked on the valve stem like washers on a bolt. These packing rings are forced down from above by the *packing flange* to apply a compressive force around the circumference of the valve stem.

Two nuts threaded onto studs maintain proper force on the packing rings. Care must be taken not to over-tighten these nuts and over-compress the packing material, or else the packing will create excessive friction on the valve stem. Not only will this friction impede precise valve stem motion, but it will also create undue wear on the stem and packing, increasing the likelihood of future packing leakage.

A closer look at the bonnet shows a multitude of components working together to form a low-friction, pressure-tight seal for the moving valve stem:

Figure 16.30: Close-up cross-section diagram illustrating packing and associated parts

In this diagram, two sets of packing rings are shown separated by a metal piece called a *lantern ring*. The lantern ring acts as a spacer allowing lubricant introduced through the lubrication port to enter into both packing sets from the middle of the bonnet.

Photographs taken of an actual valve packing assembly removed from the bonnet (left), and reassembled on the valve stem (right) reveal the structure of the packing and associated components.

(a) Removed

(b) Reassembled

Figure 16.31: Valve packing

There is no lantern ring in this packing assembly, but there is a coil spring[9]. This makes it a *live-loaded* packing as opposed to a *jam* packing. Jam packings (without springs) must be periodically adjusted to compensate for compressive fatigue and/or wear of the packing rings.

In packing applications requiring external lubrication, a *stem packing lubricator* may be connected to the lubrication port on the bonnet. This device uses a long, threaded bolt as a piston to push a quantity grease into the packing assembly:

[9] An alternative to coil-shaped springs for live-loading of valve packing are *Belleville washers*. These washers are made of spring steel and have a slightly concave shape, giving them resistance to compression along the shaft axis. Belleville washers are always stacked in opposed pairs.

Figure 16.32: A stem packing lubricator (white) connected to a valve

To operate a lubricator, the hand valve on the lubricator is first secured in the closed (shut) position, then the bolt is fully unscrewed until it falls out of the lubricator body. An appropriate lubricating grease is squeezed into the bolt hole in the lubricator body, and the bolt threaded back into place until hand-tight. Using a wrench or socket to tighten the bolt a bit more (generating pressure in the grease) and opening the hand valve allows grease to enter the packing chamber. The bolt is then screwed in fully, pushing the entire quantity of grease into the packing. As a final step, the hand valve is fully shut so there is no way for process liquid to leak out past the bolt threads.

The two most common packing materials in use today are Teflon (PTFE) and graphite. Teflon is the better of the two with regard to friction and stem wear[10]. Teflon is also quite resistant to attack from a wide variety of chemical substances. Unfortunately, it has a limited temperature range and cannot withstand intense nuclear radiation (making it unsuitable for use near reactors in nuclear power plants). Graphite is another self-lubricating packing material, and it has a far greater

[10]Based on friction values shown on page 131 of Fisher's *Control Valve Handbook* (Third Edition), Teflon packing friction is typically 5 to 10 times less than graphite packing for the same stem size!

temperature range than Teflon[11] as well as the ability to withstand harsh nuclear radiation, but creates much more stem friction than Teflon. Graphite packing also has the unfortunate property of permitting *galvanic corrosion* between the stem and bonnet metals due to its electrical conductivity. Sacrificial zinc washers are sometimes added to graphic packing assemblies to help mitigate this corrosion, but this only postpones rather than prevents corrosive damage to the stem.

Hybrid packing materials, such as carbon-reinforced Teflon, also exist in an attempt to combine the best characteristics of both technologies.

A completely different approach to packing is a device called a *bellows seal*: an accordion-like metal tube fastened to the valve stem and to the bonnet, forming a leak-proof seal with negligible friction. The accordion ribs give the bellows seal an ability to stretch and compress with a sliding stem's linear motion. Since the bellows is an uninterrupted metal tube, there is no place at all for leaks to develop, and no sliding friction to bind the valve stem's motion.

### 16.3.4 Valve seat leakage

In many process applications, it is important that the control valve be able to completely stop fluid flow when placed in the "closed" position. Although this may seem to be a fundamental requirement of any valve, it is not necessarily so. Many control valves spend most of their operating lives in a partially-open state, rarely opening or closing fully. Additionally, some control valve designs are notorious for the inability to completely shut off (e.g., double-ported globe valves). Given the common installation of manual "block" valves upstream and downstream of a control valve, there is usually a way to secure zero flow through a pipe even if a control valve is incapable of tight shut-off. For some control valve applications, however, tight shut-off is a mandatory requirement.

For this reason there are several classifications for control valves rating them in their ability to fully shut off. Seat leakage tolerances are given roman numeral designations, as shown in this table[12]:

| Class | Maximum allowable leakage rate | Test pressure drop |
|---|---|---|
| I | (no specification given) | (no specification given) |
| II | 0.5% of rated flow capacity, air or water | 45-60 psig or max. operating |
| III | 0.1% of rated flow capacity, air or water | 45-60 psig or max. operating |
| IV | 0.01% of rated flow capacity, air or water | 45-60 psig or max. operating |
| V | 0.0005 ml/min water per inch orifice size per psig | Max. operating |
| VI | Bubble test, air or nitrogen | 50 psig or max. operating |

The "bubble test" used for Class VI seat leakage is based on the leakage rate of air or nitrogen gas past the closed valve seat as measured by counting the rate of gas bubbles escaping a bubble tube submerged under water. For a 6 inch valve, this maximum bubble rate is 27 bubbles per minute (or about 1 bubble every two seconds):

[11] Graphite packing is usable in services ranging from cryogenic temperatures to 1200 degrees Fahrenheit, as opposed to Teflon which is typically rated between -40$^o$ F and 450$^o$ F.

[12] Data in this table taken from Fisher's *Control Valve Handbook*.

Figure 16.33: Diagram of a valve seat-leakage test

It is from this leakage test procedure that the term *bubble-tight shut-off* originates. Class VI shut-off is often achievable only through the use of "soft" seat materials such as Teflon rather than hard metal-to-metal contact between the valve plug and seat. Of course, this method of achieving bubble-tight shut-off comes at the price of limited operating temperature range and the inability to withstand nuclear radiation exposure.

### 16.3.5 Control valve actuators

The purpose of a control valve *actuator* is to provide the motive force to operate a valve mechanism. Both sliding-stem and rotary control valves enjoy the same selection of actuators: *pneumatic*, *hydraulic*, *electric motor*, and *hand* (manual).

#### Pneumatic actuators

Pneumatic actuators use air pressure pushing against either a flexible diaphragm or a piston to move a valve mechanism. The following photograph shows a cut-away control valve, with a pneumatic diaphragm actuator mounted above the valve body. The large coil spring provides default positioning of the valve (air pressure acting against the diaphragm moves the valve against the spring) and the rubber diaphragm at the very top. Air pressure applied to the bottom side of the diaphragm lifts the sliding stem of the valve in the upward direction, against the spring's force which tries to push the stem down:

Figure 16.34: Cutaway view of valve diaphragm actuator internals

The air pressure required to motivate a pneumatic actuator may come directly from the output of a pneumatic process controller, or from a *signal transducer* (or *converter*) translating an electrical signal into an air pressure signal. Such transducers are commonly known as *I/P* or "I to P" converters, since they typically translate an electric current signal (I) of 4 to 20 mA DC into an air pressure signal (P) of 3 to 15 psig.

The following photographs show I/P transducers of different make and model. A Fisher model 846 appears in the upper-left photograph, while an older Fisher model 546 appears in the upper-right (with cover removed). A Foxboro model E69F I/P appears in the lower-left photograph, while a Moore Industries model IPT appears in the lower-right:

(a) Model 846

(b) Model 546

Figure 16.35: Fisher I/P converters

(a) Foxboro E69F

(b) Moore IPT

Figure 16.36: Other I/P converters

Despite their differing designs and appearances, they all function the same: accepting an analog DC current signal input and a clean supply air pressure of about 20 psig, outputting a variable air pressure signal proportional to the electric current input. An interesting feature to compare between these four I/P transducers is their relative ruggedness. Every transducer shown except the Moore Industries model (lower-right) is built to withstand direct exposure to a process atmosphere, hence the heavy cast-metal housings and electrical conduit fittings. The Moore Industries unit is intended for a sheltered location, and may be plugged in to a "manifold" with several other I/P transducers to form a compact bank of transducers capable of driving air pressure signals to several valve actuators.

Some pneumatic valve actuators are equipped with *hand jacks* which are used to manually position the valve in the event of air pressure failure. The next photograph shows a sliding-stem control valve with pneumatic diaphragm actuator and a "handwheel" on the top:

Figure 16.37: A control valve with a hand jack

Note the three manual valves located around the control valve: two to *block* flow through the control valve and one to *bypass* flow around the control valve in the event of control valve failure or maintenance. These manual valves happen to be of the *gate* design, with *rising-stem* actuators to clearly show their status (stem protruding = open valve ; stem hidden = closed valve). Such block-and-bypass manual valve arrangements are quite common in the process industries where control valves fulfill critical roles and some form of manual control is needed as an emergency alternative.

Note also the air pressure tubing between the valve actuator and the air supply pipe, bent into a loop. This is called a *vibration loop*, and it exists to minimize strain on the metal tubing from vibration that may occur.

Pneumatic actuators may take the form of pistons rather than diaphragms. Illustrations of each type are shown here for comparison:

Figure 16.38: Diagram comparing diaphragm and piston actuators

Piston actuators generally have longer stroke lengths than diaphragm actuators, and are able to operate on much greater air pressures[13]. Since actuator force is a function of fluid pressure and actuator area ($F = PA$), this means piston actuators are able to generate more force than diaphragm actuators. The combination of greater force and greater displacement yields more work potential for piston actuators than diaphragm actuators of equivalent size, since mechanical work is the product of force and displacement ($W = Fx$).

Perhaps the greatest disadvantage of piston actuators as applied to control valves is friction between the piston's pressure-sealing ring and the cylinder wall. This is not a problem for on/off control valves, but it can be a significant problem for throttling valves where precise positioning is desired. Diaphragm actuators do not exhibit the same degree of friction as piston actuators because the elastic diaphragm rolls and flexes rather than rubs against a stationary surface as is the case with piston sealing rings.

A double-piston pneumatic actuator appears in the next photograph, providing the mechanical force needed to turn an on/off butterfly valve:

[13]The greater pressure rating of a piston actuator comes from the fact that the only "soft" component (the sealing ring) has far less surface area exposed to the high pressure than a rolling diaphragm. This results in significantly less stress on the elastic ring than there would be on an elastic diaphragm exposed to the same pressure. There really is no limit to the stroke length of a piston actuator as there is with the stroke length of a diaphragm actuator. It is possible to build a piston actuator miles long, but such a feat would be impossible for a diaphragm actuator, where the diaphragm must stretch (or roll) the entire stroke length.

Figure 16.39: A double-piston actuator installed on a rotary valve

In this particular actuator design, a pair of pneumatically-actuated pistons move a rack-and-pinion mechanism to convert linear piston motion into rotary shaft motion to move the butterfly trim. Note the rotary indicator (yellow in color) at the end of the rotary valve stem, showing what position the butterfly valve is in. Note also the travel switch box (black in color) housing multiple limit switches providing remote indication of valve position to the control room.

Photographs of a cut-away rack-and-pinion piston actuator (same design, just smaller) show how the pistons' linear motion is converted into rotary motion to actuate a rotary valve:

(a) Position 1 (b) Position 2

Figure 16.40: Cutaway view of rack-and-pinion piston actuator operation

Another pneumatic piston actuator design uses a simple crank lever instead of a rack-and-pinion gear set to convert linear piston motion into rotary motion. This next photograph shows such a piston actuator connected to a ball valve:

Figure 16.41: A crank lever piston actuator installed on a rotary valve

### Hydraulic actuators

Hydraulic actuators use liquid pressure rather than gas pressure to move the valve mechanism. Nearly all hydraulic actuator designs use a piston rather than a diaphragm to convert fluid pressure into mechanical force. The high pressure rating of piston actuators lends itself well to typical hydraulic system pressures, and the lubricating nature of hydraulic oil helps to overcome the characteristic friction of piston-type actuators. Given the high pressure ratings of most hydraulic pistons, it is possible to generate tremendous actuating forces with a hydraulic actuator, even if the piston area is modest. For example, an hydraulic pressure of 2,000 psig applied to one side of a 3 inch diameter piston will generate a linear thrust of over 14,000 pounds (7 tons)!

In addition to the ability of hydraulic actuators to easily generate extremely large forces, they also exhibit very *stable* positioning owing to the non-compressibility of hydraulic oil. Unlike pneumatic actuators, where the actuating fluid (air) is "elastic," the oil inside a hydraulic actuator cylinder does not yield appreciably under stress. If the passage of oil to and from a hydraulic cylinder is blocked by small valves, the actuator will become firmly "locked" into place. This may be a decided advantage for certain valve-positioning applications, where the actuator must resist forces generated on the valve trim by process fluid pressures.

Some hydraulic actuators contain their own electrically-controlled pumps to provide the fluid power, so the valve is actually controlled by an electric signal. Other hydraulic actuators rely on a separate fluid power system (pump, reservoir, cooler, hand or solenoid valves, etc.) to provide hydraulic pressure on which to operate. Hydraulic pressure supply systems, however, tend to be more limited in physical span than pneumatic distribution systems due to the need for thick-walled tubing (to contain the high oil pressure), the need to purge the system of all gas bubbles, and the problem of maintaining a leak-free distribution network. It is usually not practical to build a hydraulic oil supply and distribution system large enough to cover the entirety of an industrial facility. Another disadvantage of hydraulic systems compared to pneumatic is lack of intrinsic power storage. Compressed air systems, by virtue of air's compressibility (elasticity), naturally store energy in any pressurized volumes, and so provide a certain degree of "reserve" power in the event that the main compressor shut down. Hydraulic systems do not naturally exhibit this desirable trait.

A hydraulic piston actuator attached to a large shut-off valve (used for on/off control rather than throttling) appears in the next photograph. Two hydraulic cylinders may be seen above the round valve body, mounted horizontally. Like the pneumatic piston valve shown earlier, this valve actuator uses a rack-and-pinion mechanism to convert the hydraulic pistons' linear motion into rotary motion to turn the valve trim:

Figure 16.42: A double-piston hydraulic actuator installed on a rotary valve

A feature not evident in this photograph is a hydraulic hand pump that may be used to manually actuate the valve in the event of hydraulic system failure.

**Electric actuators**

Electric motors have long been used to actuate large valves, either in on/off mode or in throttling services. Advances in motor design and motor control circuitry has brought motor-operated valve (MOV) technology to the point where it regularly competes with legacy actuator technologies such as pneumatic.

An electric actuator appears in the next photograph, providing rotary actuation to a ball valve. This particular electric actuator comes with a hand crank for manual operation, in the event that the electric motor (or the power provided to it) fails:

Figure 16.43: An electric motor actuator installed on a rotary valve

### Hand (manual) actuators

Valves may also be actuated by hand power alone. The following valves are all "manual" valves, requiring the intervention of a human operator to actuate:

(a) Handwheel

(b) Chainwheel

Figure 16.44: Manual valve actuators

(a) Rising stem handwheel

(b) Handwheel

Figure 16.45: Additional manual valve actuators

Note the threaded stem of the left-hand valve in the last photograph pair. This stem rises and falls with the handle's turning, providing visual indication of the valve's status. Such an actuator is called a *rising-stem* design.

### 16.3.6 Valve failure mode

An important design parameter of a control valve is the position it will "fail" to if it loses motive power. For electrically actuated valves, this is typically the last position the valve was in before loss of electric power. For pneumatic and hydraulic actuated valves, the option exists of having a large spring provide a known "fail-safe" position (either open or closed) in the event of fluid pressure (pneumatic air pressure or hydraulic oil pressure) loss.

The fail-safe mode of a pneumatic/spring valve is a function of both the actuator's action and the valve body's action. For sliding-stem valves, a *direct-acting* actuator pushes down on the stem with increasing pressure while a *reverse-acting* actuator pulls up on the stem with increasing pressure. Sliding-stem valve bodies are classified as *direct-acting* if they open up when the stem is lifted, and

classified as *reverse-acting* if they shut off (close) when the stem is lifted. Thus, a sliding-stem, pneumatically actuated control valve may be made *air-to-open* or *air-to-close* simply by matching the appropriate actuator and body types. The most common combinations mix a direct-acting valve body with either a reverse- or direct-acting valve actuator, as shown in this illustration:

Figure 16.46: Gate valve illustrating valve failure positions

Valve fail mode may be shown in instrument diagrams by either an arrow pointing in the direction of failure (assuming a direct-acting valve body where stem motion toward the body closes and stem motion away from the body opens the valve trim) and/or the abbreviations "FC" (fail closed) and "FO" (fail open). Other failure modes are possible, as indicated by this set of valve symbols:

Figure 16.47: Instrumentation diagram symbols used for valve failure position

In order for a pneumatic or hydraulic valve to fail in the *locked* state, an external device must trap fluid pressure in the actuator's diaphragm or piston chamber in the event of supply pressure loss.

Valves that fail in place and drift in a particular direction are usually actuated by double-acting pneumatic piston actuators. These actuators do not use a spring to provide a definite fail mode, but rather use air pressure both to open and to close the valve. In the event of an air pressure loss, the actuator will neither be able to open nor close the valve, and so it will tend to remain in position. If the valve is of the globe design with unbalanced trim, forces exerted on the valve plug will move it in one direction (causing drift).

It is important to note how the failure mode of a valve is often linked to its control action (air-to-open, air-to-close)[14]. That is, an air-to-open pneumatic control valve will fail closed on loss of air pressure, and visa-versa. This is an important fact because good safety engineering demands that the risk factors of the process determine proper valve failure mode rather than control system

[14]Exceptions exist for valves designed to fail in place, where a valve may be engineered to "lock" in position through the action of an external device whether the valve itself is air-to-open or air-to-close.

convention or habit. People may find it easier to understand the operation of an air-to-open control valve than an air-to-close valve (more signal = more process fluid flow), but this should not be a guiding principle in valve selection. Air-to-open control valves fail closed by their very nature (unless equipped with automatic "locking" devices in which case they will fail in place), which means they are appropriate for a particular process control application *only* if that process is safer with a failed-closed valve than with a failed-open valve. If the process is safer with a fail-open valve, then the pneumatically-actuated control valve specified for that application needs to be air-to-close.

In fact, this basic principle forms the basis – or at least it *should* form the basis – of decisions made for all instrument actions in critical control loops: first determine the safest mode of valve failure, then select and/or configure instrument actions in such a way that the most probable modes of signal path failure will result in the control valve consistently moving to that (safest) position.

For example, consider this automated cooling system for a large power-generating engine:

Figure 16.48: An automated cooling system for an engine generator

Clearly, it is more hazardous to the engine for the valve to fail closed than it would be for the valve to fail open. If the valve fails closed, the engine will surely overheat from lack of cooling. If it fails open, the engine will merely run cooler than designed, the only negative consequence being decreased efficiency. With this in mind, the only sensible choice for a control valve is one that fails open (air-to-close).

However, our choices in instrument action do not end with the control valve. How should the temperature transmitter, the controller, and the I/P transducer be configured to act? In each case, the answer should be to act in such a way that the valve will default to its fail-safe position (wide open) in the event of the most likely input signal fault.

Stepping "backward" through the control system from the valve to the temperature sensor, the next instrument encountered is the I/P transducer. Its job, of course, is to convert a 4-20 mA current signal into a corresponding pneumatic pressure that the valve actuator can use. Since the

valve's failure mode is based on a loss of actuating air pressure, the I/P must be configured in such a way that it outputs minimum pressure in the event of an electrical fault in its 4-20 mA input signal wiring. The most common fault for a current loop is an *open*, where current goes to 0 milliamps. Thus, the configuration of the I/P transducer should be *direct*, such that a 4 to 20 mA input signal produces a 3 to 15 psig output pressure, respectively (i.e., minimum input current yields minimum output pressure).

The next instrument in the loop is the controller. Here, the most likely input signal failure must result in a minimum output signal so the valve will (once again) default to its "fail safe" position. Consequently, the controller should be configured for *direct* action just like the I/P transducer (i.e., a decreasing PV signal from a broken wire or loose connection in the input circuit results in a decreasing output signal).

Finally, there is the last instrument in the control loop: the temperature transmitter (TT). As with most instruments, it may be configured for direct or reverse action. Should direct be chosen (i.e., hotter engine = more mA output) or reverse (hotter engine = less mA output)? Here, a choice needs to be made in such a way that the overall effect of the control system is *negative feedback*. In other words, the transmitter needs to be configured such that a hotter engine results in increased coolant flow (a wider-open control valve). Since the rest of the system has been designed so a minimum signal anywhere tends to drive the valve to its fail-safe mode (wide open), choose a *reverse-acting* transmitter, so a hotter engine results in a decreased milliamp signal from the transmitter. If the transmitter has a sensor "burnout" mode switch, this switch should be flipped into the low-scale burnout position so a burned-out sensor will result in 4 mA output (low end of the 4-20 mA scale), thus driving the valve into its safest (wide-open) position.

Such a configuration – with its air-to-close control valve and a reverse-acting transmitter – may seem strange and counter-intuitive, but it is the safest design for this engine cooling system. This "odd" configuration of instruments was arrived at by first choosing the safest control valve failure mode, then choosing the other device actions in such a way that the most likely signal-path failures anywhere in the system would result in the same, consistent valve response. Of course it should go without saying that accurate documentation in the form of a loop diagram with instrument actions clearly shown is an absolutely essential piece of the whole system. If the safety of a control system depends on using any "non-standard" instrument configurations, those configurations had better be documented so those maintaining the system in the future will know what to expect!

### 16.3.7 Actuator bench-set

Valve actuators provide force to move control valve trim. For precise positioning of a control valve, there must be a calibrated relationship between applied force and valve position. Most pneumatic actuators exploit *Hooke's Law* to translate applied air pressure to valve stem position.

$$F = kx$$

Where,

$F$ = Force applied to spring in newtons (metric) or pounds (British)

$k$ = Constant of elasticity, or "spring constant" in newtons per meter (metric) or pounds per foot (British)

$x$ = Displacement of spring in meters (metric) or feet (British)

Hooke's Law is a linear function, which means that spring motion will be linearly related to applied force from the actuator element (piston or diaphragm). Since the working area of a piston or diaphragm is constant, the relationship between actuating fluid pressure and force will be a simple proportion ($F = PA$). By algebraic substitution, Hooke's Law may be altered to include pressure and area:

$$F = kx$$

$$PA = kx$$

Solving for spring compression as a function of pressure, area, and spring constant:

$$x = \frac{PA}{k}$$

Figure 16.49: A pneumatic actuator

When a control valve is assembled from an actuator and a valve body, the two mechanisms must be coupled together in such a way that the valve moves between its fully closed and fully open positions with an expected range of air pressures. A common standard for pneumatic control valve actuators is 3 to 15 psig[15].

[15] 3 psig could mean fully closed and 15 psig fully open, or visa-versa, depending on what form of actuator is coupled to what form of valve body. A direct-acting actuator coupled to a direct-acting valve body will be open at low pressure and closed at high pressure (increasing pressure pushing the valve stem toward the body, closing off the valve trim), resulting in *air-to-close* action. Reversing either actuator or valve type (e.g., reverse actuator with direct valve or direct actuator with reverse valve) will result in *air-to-open* action.

There are really only two mechanical adjustments that need to be made when coupling a pneumatic diaphragm actuator to a sliding-stem valve: the *stem connector* and the *spring adjuster*. The stem connector mechanically joins the sliding stems of both actuator and valve body so they move together as one stem. This connector must be adjusted so neither the actuator nor the valve trim prevents full travel of the valve trim:

Figure 16.50: Valve operation via a diaphragm pneumatic actuator

Note how the plug is fully against the seat when the valve is closed, and how the travel indicator indicates fully open at the point where the actuator diaphragm nears its fully upward travel limit. This is how things should be when the stem connector is properly adjusted.

If the stem connector is set with the actuator and valve stems spaced too far apart (i.e., the total stem length is too long), the actuator diaphragm will bind travel at the upper end and the valve plug will bind travel at the lower end. The result is a valve that cannot ever fully open:

Figure 16.51: Valve failing to open fully with an improper actuator setup

A control valve improperly adjusted in this manner will never achieve full-flow capacity, which may have an adverse impact on control system performance.

If the stem connector is set with the actuator and valve stems too closely coupled (i.e., the total stem length is too short), the actuator diaphragm will bind travel at the lower end and the valve plug will bind travel at the upper end. The result is a valve that cannot ever fully close:

Figure 16.52: Valve failing to shut fully with an improper actuator setup

This is a very dangerous condition: a control valve that lacks the ability to fully shut off. The process in which this valve is installed may be placed in jeopardy if the valve lacks the ability to stop the flow of fluid through it!

Once the stem length has been properly set by adjusting the stem connector, the spring adjuster must be set for the proper *bench set* pressure. This is the pneumatic signal pressure required to lift the plug off the seat. For an air-to-open control valve with a 3 to 15 psig signal range, the "bench set" pressure would be 3 psig.

Bench set is a very important parameter for a control valve because it establishes the seating pressure of the plug when the valve is fully closed. Proper seating pressure is critical for tight shut-off, which carries safety implications in some process services. Consult the manufacturer's instructions when adjusting the bench set pressure for any sliding-stem control valve. These instructions will typically describe both the stem connector and the spring adjuster procedures, to ensure both parameters are correctly set.

### 16.3.8 Pneumatic actuator response

A limitation inherent to pneumatic valve actuators is the amount of air flow required to or from the actuator to cause rapid valve motion. This is an especially acute problem in all-pneumatic control systems, where the distance separating a control valve from the controller may be substantial:

Figure 16.53: Transient response due to long pneumatic transmission lines affects valve dynamics

The combined effect of air-flow friction in the tube, flow limitations inherent to the controller mechanism, and volume inside the valve actuator conspire to create a sluggish valve response to sudden changes in controller output signal, not unlike the response of an RC (resistor-capacitor) time-delay circuit where a step-change in voltage input results in an inverse exponential output signal.

If the pneumatic valve actuator is driven by an I/P transducer instead of directly by a pneumatic controller, the problem is lessened by the ability to locate the I/P close to the actuator, thus greatly minimizing tube friction and thus minimizing the "time constant" ($\tau$) of the control valve's response:

Figure 16.54: Transient response due to electronic signal transmission is faster than pneumatic

Still, if the pneumatic actuator is particularly large in volume, an I/P transducer may experience trouble supplying the necessary air flow rate to rapidly actuate the control valve. Certainly the problem of time delay is reduced, but not eliminated, by the close-coupled location of the I/P transducer to the actuator.

One way to improve valve response in either type of system (full-pneumatic or I/P-driven) is to use a device known as a *volume booster* to source and vent compressed air for the valve actuator. A "volume booster" is a pneumatic device designed to reproduce a pneumatic pressure signal (1:1 ratio), but with far greater output flow capacity. In electrical terms, a volume booster is analogous to a *voltage follower*: a circuit designed to boost current to a load, without boosting or diminishing voltage. A 3 to 15 psig pneumatic pressure signal applied to the input of a volume booster will result in an identical output signal (3 to 15 psig), but with greatly enhanced flow capacity.

A pneumatic control system equipped with a volume booster would look something like this:

Figure 16.55: A local volume booster used with pneumatic signal transmission improves valve transient response

Of course, enhanced air flow to and from the actuator does not completely eliminate time delays in valve response. So long as the flow rate into or out of an actuator is finite, some time will be required to change pressure inside the actuator and thus change valve position. However, if the actuator volume cannot be reduced for practical reasons of actuating force (larger diaphragm or piston area needed for more force, also resulting in more volume for any given stroke length), then the only variable capable of reducing time lag is increased air flow rate, and a volume booster directly addresses that deficiency.

Internally, a volume booster's construction is not unlike a manually-adjusted pressure regulator[16]:

[16]The volume booster design shown here is loosely based on the Fisher model 2625 volume boosting relay.

Figure 16.56: Conceptual diagram illustrating the function of a volume booster

In either mechanism, an internal diaphragm senses output pressure and acts against a restraining force (either a spring preloaded by a hand adjustment screw or an external pressure signal acting on another diaphragm) to position an air flow throttling/venting mechanism. If the output pressure is less than desired, the diaphragm moves down to open the air sourcing plug and supply additional air volume to the output. If the output pressure is greater than desired, the diaphragm moves up to shut off the sourcing plug and open up the venting port to relieve air pressure to atmosphere.

### 16.3.9 Valve positioners

Springs work quite nicely to convert mechanical force into mechanical motion (Hooke's Law – $F = kx$) for valve actuators if and only if the sole forces involved are the diaphragm or piston force against the spring's resistance force. If any other force acts upon the system, the relationship between actuating fluid pressure and valve stem travel will not necessarily be proportional.

Unfortunately, there typically are other forces acting on a valve stem besides the actuating fluid pressure's force and the spring's reaction force. Friction from the stem packing is one force, and reaction force at the valve plug caused by differential pressure across the plug's area is another[17]. These forces conspire to re-position the valve stem so stem travel does not precisely correlate to actuating fluid pressure.

A common solution to this dilemma is to add a *positioner* to the control valve assembly. A positioner is a motion-control device designed to actively compare stem position against the control signal, adjusting pressure to the actuator diaphragm or piston until the correct stem position is reached:

[17] One way to minimize dynamic forces on a globe valve plug is to use a *double-ported* plug design, or to use a *balanced* plug on a cage-guided globe valve. A disadvantage to both these valve plug designs, though, is greater difficulty achieving tight shut-off.

Figure 16.57: Diagram showing a positioner mounted on a control valve

Positioners essentially act as control systems within themselves[18]: the valve's stem position is the process variable (PV), the command signal to the positioner is the setpoint (SP), and the positioner's signal to the valve actuator is the manipulated variable (MV) or output. The controller within the positioner is a high-gain amplifier that acts on the difference between the SP and the PV. Thus, when a process controller sends a command signal to a valve equipped with a positioner, the positioner receives that command signal and does its best to ensure the valve stem position follows along.

The following photograph shows a Fisher model 3582 pneumatic positioner mounted to a control valve. The positioner is the grey-colored box with three pressure gauges on the right-hand side:

[18]The technical term for this type of control system is *cascade*, where one controller's output becomes the setpoint for a different controller. In the case of a valve positioner, the positioner receives a valve stem position setpoint from the main process controller.

Figure 16.58: An industrial control valve with a positioner

A more modern positioner appears in the next photograph, the Fisher DVC6000 (again, the grey-colored box with pressure gauges on the right-hand side):

Figure 16.59: An industrial control valve with a digital positioner

Positioners such as the DVC6000 are considered "smart" devices, containing digital electronic microprocessors to monitor and control valve stem position in accordance with the control signal, and also store data relevant to diagnostics[19].

Control valve positioners are typically constructed in such a way to source and vent high air flow rates, such that the positioner also fulfills the functionality of a volume booster. Thus, a positioner not only ensures more precise valve stem positioning, but also faster stem velocity (and shorter time delays) than if the valve actuator were directly "powered" by an I/P transducer.

While beneficial on spring-equipped valve actuators, positioners are absolutely essential for positioners lacking springs such as double-acting pneumatic piston actuators. Some form of positioning mechanism is also required on electric motor actuators intended for throttling service, because an electric motor is not "aware" of its own shaft position in order that it may precisely move a control valve. Thus, a positioner circuit using a potentiometer or LVDT/RVDT sensor to

[19] Examples of diagnostic data recorded by smart positioners includes error (command signal − actual valve position), pressure versus motion relationships (used to measure valve packing friction), supply air pressure, and ambient temperature. Smart positioners even have the ability to totalize valve stem travel over long periods of time, enabling predictive maintenance alerts for wearing components such as packing and piston sealing rings.

detect valve stem position and a set of transistor outputs to drive the motor is necessary to make an electric actuator responsive to an analog control signal.

A simple force-balance pneumatic valve positioner design appears in the following illustration:

Figure 16.60: Diagram of a force-balance pneumatic positioner

The control signal for this valve is a 3 to 15 psig pneumatic signal, coming from either an I/P transducer or a pneumatic controller (neither one shown in the illustration). This control signal pressure applies an upward force on the force beam, such that the baffle tries to approach the

nozzle. Increasing backpressure in the nozzle causes the pneumatic amplifying relay to output a greater air pressure to the valve actuator, which in turn lifts the valve stem up (opening up the valve). As the valve stem lifts up, the spring connecting the force beam to the valve stem becomes further stretched, applying additional force to the right-hand side of the force beam. When this additional force balances the bellows' force, the system stabilizes at a new equilibrium.

Like all force-balance systems, the force beam motion is greatly constrained by the balancing forces, such that its motion is negligible for all practical purposes. In the end, equilibrium is achieved by one force balancing another, like two teams of people pulling oppositely on a length of rope: so long as the two teams' forces remain equal in magnitude and opposite in direction, the rope will not deviate from its original position.

The following photograph shows a PMV model 1500 force-balance positioner used to position a rotary valve actuator, with the cover on (left) and removed (right):

(a) Positioner (b) Cover removed

Figure 16.61: An industrial force-balance positioner

The 3-15 psig pneumatic control signal enters into the bellows, which pushes downward on the horizontal force beam. A pneumatic pilot valve assembly at the left-hand side of the force beam detects any motion, increasing air pressure to the valve actuating diaphragm if any downward motion is detected and releasing air pressure from the actuator if any upward motion is detected:

Figure 16.62: Internals of a force-balance pneumatic positioner

As compressed air is admitted to the valve actuator by this pilot valve assembly, the rotary valve will begin to rotate open. The shaft's rotary motion is converted into a linear motion inside the positioner by means of a *cam*: a disk with an irregular radius designed to produce linear displacement from angular displacement:

Figure 16.63: Cam and follower inside a force-balance pneumatic positioner

A roller-tipped *follower* at the end of another beam rides along the cam's circumference. Beam motion caused by the cam is translated into linear force by the compression of a coil spring directly against the force of the pneumatic bellows on the force beam. When the cam moves far enough to

compress the spring enough to balance the additional force generated by the bellows, the force beam will come to an equilibrium position and the valve will stop moving.

Motion-balance pneumatic valve positioner designs also exist, whereby the motion of the valve stem counteracts motion (not force) from another element. The following illustration shows how a simple motion-balance positioner would work:

Figure 16.64: Diagram of a motion-balance pneumatic positioner

In this mechanism, an increasing signal pressure causes the beam to advance toward the nozzle, generating increased nozzle backpressure which then causes the pneumatic amplifying relay to send more air pressure to the valve actuator. As the valve stem lifts up, the upward motion imparted

to the right-hand end of the beam counters the beam's previous advance toward the nozzle. When equilibrium is reached, the beam will be in an angled position with the bellows' motion balanced by valve stem motion.

The following photograph shows a close view of a Fisher model 3582 pneumatic motion-balance positioner's mechanism:

Figure 16.65: Internals of a motion-balance pneumatic positioner

At the heart of this mechanism is a D-shaped metal ring that translates bellows motion and valve stem motion into flapper (baffle) motion. As the bellows (located underneath the upper-right corner of the D-ring) expands with increasing pneumatic signal pressure, it rocks the beam along its vertical axis. With the positioner set for direct-acting operation, this rocking motion drives the flapper closer to the nozzle, increasing backpressure and sending more compressed air to the valve actuator.

As the valve stem moves, a feedback lever rotates a cam underneath the bottom-most portion of the D-ring. A roller follower riding on that cam translates the valve stem's motion to another rocking motion on the beam, this time along the horizontal axis. Depending on how the cam has been fixed to the feedback shaft, this motion may rock the flapper away from the nozzle or further toward the nozzle. This selection of cam orientation must match the action of the actuator: either direct (air to extend the stem) or reverse (air to retract the stem).

The D-ring mechanism is rather ingenious, for it allows convenient adjustment of span by angling the flapper (baffle) assembly at different points along the ring's circumference. If the flapper assembly is set close to horizontal, it will be maximally sensitive to bellows motion and minimally sensitive to valve stem motion, forcing the valve to move further to balance small motions of the bellows (long stroke length). Conversely, if the flapper assembly is set close to vertical, it will be maximally sensitive to valve stem motion and minimally sensitive to bellows motion, resulting in little valve stroke (i.e., the bellows needs to expand greatly in order to balance a small amount of stem motion).

Electronic valve positioners, such as the Fisher model DVC6000, use an electronic sensor to

detect valve stem position, compare that sensed position against the control signal by subtraction (error = position − signal), then send an appropriate pneumatic pressure to the valve actuator to minimize that error. A photograph of the potentiometer from a Fisher DVC6000 positioner appears here:

Figure 16.66: A potentiometer used as a valve stem position detector

The DVC6000 positioner also contains air pressure sensors to monitor actuator air pressure as the valve moves. Being able to measure both stem position and actuator air pressure in real time allows the positioner to correlate one variable to the other in the form of a graph. Such a graph contains much useful diagnostic information for troubleshooting valve problems such as excessive packing friction, bent valve stems, and valve trim damage.

"Smart" positioners used on electric actuators have the capability to provide similar diagnostic data, correlating stem position with actuator torque (measured either indirectly by motor current or directly by a torque sensor in the gear train). Such data is quite valuable in predictive maintenance programs, used to identify when valve packing friction becomes excessive, or if valve trim components become damaged and no longer seat together properly. These diagnostic tools apply even to open/close motor-operated valves not used for throttling service, and are especially useful on gate, plug, and ball-type shut-off valves where seat engagement is substantial for tight shut-off.

### 16.3.10 Control valve sizing

When control valves operate between fully open and fully shut, they serve much the same purpose in process systems as resistors do in electric circuits: to dissipate energy. Like resistors, the form

that this dissipated energy takes is mostly heat, although some of the dissipated energy manifests in the form of vibration and noise[20].

In most control valves, the dominant mechanism of energy dissipation comes as a result of turbulence introduced to the fluid as it travels through constrictive portions of the valve trim. The following illustration shows these constrictive points within two different control valve types (shown by arrows):

Figure 16.67: Turbulent energy dissipation in control valves

The act of choosing an appropriate control valve for the expected energy dissipation is called *valve sizing*. ISA Standard 75.01.01, *Flow Equations for Sizing Control Valves*, provides in-depth calculations needed to properly size a control valve, but a simple method is presented later in the chapter to get a rough estimate of proper size.

### Physics of energy dissipation in a turbulent fluid stream

As one might expect, control valves are rated in their ability to throttle fluid flow, much as resistors are rated in their ability to throttle the flow of electrons in a circuit. For resistors, the unit of measurement for electron flow restriction is the *ohm*: 1 ohm of resistance results in a voltage drop of 1 volt across that resistance given a current through the resistance equal to 1 ampere:

[20] Valve noise may be *severe* in some cases, especially in certain gas flow applications. An important performance metric for control valves is noise production expressed in decibels (dB).

Figure 16.68: Voltage and current in a resistor

The mathematical relationship between current, voltage, and resistance for any resistor is *Ohm's Law*:

$$R = \frac{V}{I}$$

Where,
$R$ = Electrical resistance in ohms
$V$ = Electrical voltage drop in volts
$I$ = Electrical current in amperes

Ohm's Law is a simple, linear relationship, expressing the "friction" encountered by electric charge carriers as they slowly drift through a solid object.

When a fluid moves turbulently through any restriction, energy is inevitably dissipated in that turbulence. The amount of energy dissipated is proportional to the kinetic energy of the turbulent motion, which is proportional to the square of velocity according to the classic kinetic energy equation for moving objects:

$$E_k = \frac{1}{2}mv^2$$

To rewrite this equation to express the amount of kinetic energy represented by a volume of moving fluid with velocity $v$:

$$\text{Kinetic energy per unit volume} = \frac{1}{2}\rho v^2$$

The amount of energy dissipated by turbulence in such a fluid stream is known to be some fraction ($k$, $0 < k < 1$) of the total kinetic energy, so:

$$\text{Energy dissipated per unit volume} = \frac{1}{2}k\rho v^2$$

Any energy lost in turbulence eventually manifests as a loss in fluid pressure. Thus, a control valve throttling a fluid flowstream will have a greater upstream pressure than downstream pressure (assuming all other factors such as pipe size and height above ground level being the same downstream as upstream):

Figure 16.69: Pressure and flow in a restriction

This pressure drop ($P_1 - P_2$, or $\Delta P$) is equivalent to the voltage drop seen across any current-carrying resistor, and may be substituted for dissipated energy per unit volume in the previous equation[21]. Also, $\frac{Q}{A}$ may be substituted for velocity $v$ because volumetric flow rate ($Q$) is the product of fluid velocity and pipe cross-section area ($Q = Av$) for incompressible fluids such as liquids:

$$P_1 - P_2 = \frac{1}{2}k\rho\left(\frac{Q}{A}\right)^2$$

Next, solve for a quotient with pressure drop ($P_1 - P_2$) in the numerator and flow rate $Q$ in the denominator so the equation bears a resemblance to Ohm's Law ($R = \frac{V}{I}$):

$$P_1 - P_2 = \frac{1}{2}k\rho\frac{Q^2}{A^2}$$

$$\frac{P_1 - P_2}{Q^2} = \frac{k\rho}{2A^2}$$

$$\frac{\sqrt{P_1 - P_2}}{Q} = \sqrt{\frac{k\rho}{2A^2}}$$

Either side of the last equation represents a sort of "Ohm's Law" for turbulent liquid restrictions: the left-hand side expressing fluid "resistance" in the state variables of pressure drop and volumetric flow, and the right-hand term expressing fluid "resistance" as a function of fluid density and restriction geometry. Obviously, pressure drop ($P_1 - P_2$) and volumetric flow rate ($Q$) are not linearly related as voltage and current are for resistors, but that nevertheless there still is a quantity that acts like a "resistance" term:

$$R = \frac{\sqrt{P_1 - P_2}}{Q} \qquad\qquad R = \sqrt{\frac{k\rho}{2A^2}}$$

Where,
$R$ = Fluid "resistance"
$P_1$ = Upstream fluid pressure
$P_2$ = Downstream fluid pressure
$Q$ = Volumetric fluid flow rate
$k$ = Turbulent energy dissipation factor

[21] It is appropriate to express energy loss per unit volume in the same units of measurement as pressure (when Bernoulli's equation is examined, it can be seen that the units of $\frac{1}{2}\rho v^2$ and $P$ are actually the same.

$\rho$ = Mass density of fluid
$A$ = Cross-sectional area of restriction

The fluid "resistance" of a restriction depends on several variables: the proportion of kinetic energy lost due to turbulence ($k$), the density of the fluid ($\rho$), and the cross-sectional area of the restriction ($A$). In a control valve throttling a liquid flow stream, only the first and last variables are subject to change with stem position, fluid density remaining relatively constant.

In a wide-open control valve, especially valves offering a nearly unrestricted path for moving fluid (e.g., ball valves, eccentric disk valves), the value of $A$ will be at a maximum value essentially equal to the pipe's area, and $k$ will be nearly zero[22]. In a fully shut control valve, $A$ is zero, creating a condition of infinite "resistance" to fluid flow.

It is customary in control valve engineering to express the "restrictiveness" of any valve in terms of how much flow it will pass given a certain pressure drop and fluid specific gravity ($G_f$). This measure of valve performance is called *flow capacity* or *flow coefficient*, symbolized as $C_v$. A greater flow capacity value represents a less restrictive (less "resistive") valve, able to pass greater rates of flow for the same pressure drop. This is analogous to expressing an electrical resistor's rating in terms of conductance ($G$) rather than resistance ($R$): how many amperes of current it will pass with 1 volt of potential drop ($I = GV$ instead of $I = \frac{V}{R}$).

One of the equations presented earlier expressing pressure drop in terms of flow rate, restriction area, dissipation factor, and densitymay be manipulated into a form expressing flow rate ($Q$) in terms of pressure drop and density, collecting $k$ and $A$ into a third term which will become flow capacity ($C_v$):

$$P_1 - P_2 = \frac{1}{2} k \rho \frac{Q^2}{A^2}$$

First, substitute specific gravity ($G_f$) for mass density ($\rho$) using the following definition of specific gravity:

$$G_f = \frac{\rho}{\rho_{water}}$$

$$\rho_{water} G_f = \rho$$

Substituting and continuing with the algebraic manipulation:

$$P_1 - P_2 = \frac{1}{2} k \rho_{water} G_f \frac{Q^2}{A^2}$$

$$\frac{P_1 - P_2}{G_f} = \frac{1}{2} k \rho_{water} \frac{Q^2}{A^2}$$

$$\left(\frac{2A^2}{k \rho_{water}}\right)\left(\frac{P_1 - P_2}{G_f}\right) = Q^2$$

[22] In a case of minimal throttling, almost none of the fluid's kinetic energy is lost to turbulence, but rather passes right through the valve unrestricted.

$$Q = \sqrt{\frac{2A^2}{k\rho_{water}}}\sqrt{\frac{P_1 - P_2}{G_f}}$$

The first square-rooted term in the equation, $\sqrt{\frac{2A^2}{k\rho_{water}}}$, is the valve capacity or $C_v$ factor. Substituting $C_v$ for this term results in the simplest form of valve sizing equation (for incompressible fluids):

$$Q = C_v\sqrt{\frac{P_1 - P_2}{G_f}}$$

In the United States of America, $C_v$ is defined as the number of gallons per minute of water that will flow through a valve with 1 psig of pressure drop[23]. A similar valve capacity expression used elsewhere in the world rates valves in terms of how many cubic meters per hour of water will flow through a valve with a pressure drop of 1 bar. This latter flow capacity is symbolized as $K_v$.

Thus, to size a valve properly for liquid service, the minimum, nominal, and maximum flows are required, along with the allowable pressure drop across the valve at each flow. A $C_v$ can be calculated for each flow rate, then the maximum of the three calculated $C_v$ values is chosen. Typically, it is desirable for a control valve to only operate between about 20% and 80% open, as most valves have poor throttling characteristics when in the upper and lower ends of the operating range. Taking this into account, the maximum $C_v$ can be multiplied by 1.25 and used as the desired $C_v$ to allow the maximum flow through at about 80% open. Once the desired $C_v$ is obtained, manufacturer's catalogs allow the engineer to select the right size of the valve. A valve with a $C_v$ value near the desired $C_v$ value should be chosen. After the size has been selected, the engineer should check to ensure the valve will operate in the desired range for all three flow conditions.

For the best results predicting required $C_v$ values for control valves in any service, use valve sizing software provided by control valve manufacturers. Modern valve sizing software is easy to use, especially when referenced to specific models of control valve sold by that manufacturer, and is able to account for a diverse multitude of factors affecting proper sizing.

Control valve sizing is complex enough that some valve manufacturers used to give away "slide rule" calculator devices so customers could choose the $C_v$ values they needed with relative ease. Photographs of a two-sided valve sizing slide rule are shown here for historical reference:

[23] The specification of certain British units of measurement for flow and pressure drop means that there is more to $C_v$ than just $\sqrt{\frac{2A^2}{k\rho_{water}}}$. $C_v$ also incorporates a factor necessary to account for the arbitrary choice of units.

Figure 16.70: Side 1 of a slide rule calculator for sizing valves

Figure 16.71: Side 2 of a slide rule calculator for sizing valves

### Importance of proper valve sizing

The flow coefficient of a control valve is a numerical value usually expressing maximum flow capacity. For example, a control valve with a $C_v$ rating of 45 should flow 45 gallons per minute of water through it with a 1 psig pressure drop *when wide open*. The flow coefficient value for this same control valve will be less than 45 when the valve position is anything less than fully open. When the control valve is in the fully shut position, its $C_v$ value will be zero. Thus, it should be understood that $C_v$ is truly

a variable – not a constant – for any control valve, even though control valves are often specified simply by their maximum flow capacity. The $C_v$ for a valve is actually a function of the valve's position (this is discussed later in the section on control valve characterization).

It should be obvious that any control valve must be sized large enough (i.e., possess sufficient maximum $C_v$ capacity) to flow the greatest expected flow rate in any given process installation. A valve that is too small for an application will not be able to pass enough process fluid through it when needed.

Given this fact, it may seem safe to choose a valve sized much larger than what is needed, just to avoid the possibility of not having enough flow capacity. For instance, consider this control valve sizing problem, where a characterized ball valve controls the flow rate of water out of a surge tank to maintain a constant water level 25 feet higher than the height of the valve:

Figure 16.72: A control valve sizing problem

According to the process engineers, the maximum expected flow rate for this valve is 470 gpm. What should the maximum $C_v$ rating be for this valve? To begin, the expected pressure drop must be known across the valve. The 25 foot water column height upstream provides the means to calculate $P_1$:

$$P = \gamma h$$

$$P_1 = (62.4 \text{ lb/ft}^3)(25 \text{ feet})$$

$$P_1 = 1560 \text{ psf} = 10.8 \text{ psig}$$

There is no need to calculate $P_2$, since the P&ID shows that the downstream side of the valve is vented to atmosphere, and is thus 0 psig gauge pressure. This gives a pressure drop of 10.8 psig

across the control valve, with an expected maximum flow rate of 470 gpm. Manipulating our flow capacity equation to solve for $C_v$:

$$Q = C_v \sqrt{\frac{P_1 - P_2}{G_f}}$$

$$C_v = \frac{Q}{\sqrt{\frac{P_1 - P_2}{G_f}}}$$

$$C_v = \frac{470 \text{ gpm}}{\sqrt{\frac{10.8 \text{ psig}}{1}}}$$

$$C_v = 143$$

This says a control valve with a $C_v$ value of *at least* 143 is needed to meet the specified (maximum) flow rate. A valve with insufficient $C_v$ would not be able to flow the required 470 gallons per minute of water with only 10.8 psig of pressure drop. However, since the *effective* maximum throttling occurs at approximately 80% open, many engineers will divide the calculated $C_v$ by 80% (0.80) and select the next higher published $C_v$ so that the valve can still effectively throttle the maximum flow. This means that a valve with a $C_v$ value of at least 179 is needed.

Does this mean the valve may be safely over-sized by any factor? Would there be any problem with installing a control valve with a $C_v$ value of 300? The general answer to these questions is that over-sized valves may be problematic. Not only is there the possibility of allowing too much flow under wide-open conditions (consider whatever process vessels and equipment lie downstream of the oversized valve), but also that the process will be difficult to control under low-flow conditions.

In order to understand how an over-sized control valve leads to unstable control, an exaggerated example is helpful to consider: imagine installing a fire hydrant valve on a kitchen sink faucet[24]. Certainly, a wide-open hydrant valve would allow sufficient water flow into a kitchen sink. However, most of this valve's usable range of throttling will be limited to the first *percent* of stem travel. After the valve is opened just a few percent from fully shut, restrictions in the piping of the house's water system will have limited the flow rate to its maximum, thus rendering the rest of the valve's stem travel capacity utterly useless. It would be challenging indeed to try filling a drinking cup with water from this hydrant valve: just a little bit too much stem motion and the cup would be subjected to a full-flow stream of water!

Control valve over-sizing is a common problem in industry, often created by future planning for expanded process flow. "If a large valve is installed now," so the reasoning goes, "it won't have to replace a smaller valve when the time comes to increase production rate." In the interim period when that larger valve must serve to control a meager flow rate, however, problems caused by poor control quality may end up costing the enterprise more than the cost of an additional valve.

[24] For those readers who may be unfamiliar with American terminology, a *fire hydrant* is a large hand valve installed at intervals along public roadways, allowing connection of fire hoses to an underground water supply pipe in the event of an emergency fire. These valves are quite large, and would be comically oversized if installed inside a person's house, for any purpose.

### Gas valve sizing

Sizing a control valve for gas or vapor service is more complicated than for liquid service, due to the compressibility of gases and vapors. As a gas or vapor compresses with changes in pressure, its density changes correspondingly. In previous mathematical analyses of fluid flow restriction, one of our assumptions was that fluid density ($\rho$) remained constant. This assumption may hold true for some flowing gas conditions as well, provided minimal pressure changes within the path of flow. However, for most control valve applications where the very purpose of the valve is to introduce substantial pressure changes in a fluid stream, the assumption of constant fluid density is unrealistic.

Shown here is one of the simpler gas valve sizing equations:

$$Q = 963\, C_v \sqrt{\frac{\Delta P(P_1 + P_2)}{G_g T}}$$

Where,

$Q$ = Gas flow rate, in units of Standard Cubic Feet per Hour (SCFH)
$C_v$ = Valve capacity coefficient
$\Delta P$ = Pressure dropped across valve, pounds per square inch differential (psid)
$P_1$ = Upstream valve pressure, pounds per square inch absolute (psia)
$P_2$ = Downstream valve pressure, pounds per square inch absolute (psia)
$G_g$ = Specific gravity of gas (Air at standard temperature and pressure = 1.0)
$T$ = Absolute temperature of gas in degrees Rankine ($^o$R)

This equation holds true only for "subcritical" flow, where the moving gas stream velocity never approaches the speed of sound[25]. Other equations exist for calculating flow rates of gas through control valves when sonic flows are achieved. Note the inclusion of absolute pressures in this equation, and not just differential pressure ($\Delta P$, or $P_1 - P_2$). This is intended to correct for effects related to compression of the gas under pressure.

Valve sizing is complicated enough, both for liquid and gas service, that the use of valve sizing computer software is strongly recommended as opposed to hand-calculations. The number of important parameters, nonlinear factors, and alternative equations relevant to control valve sizing are numerous enough to bewilder most technicians (and more than a few engineers). Valve sizing software will also predict noise levels generated by the valve, and in many cases specify actual valve trim styles offered by the manufacturer for mitigating problems such as noise.

### Relative flow capacity

The flow capacity of a valve ($C_v$) is a quantitative rating of its ability to pass a fluid flow for a set of given pressure and density conditions. $C_v$ may be predicted, or empirically measured, for any type of control valve given the proper information.

Not all control valve types exhibit the same $C_v$ coefficients, however, for the same pipe size. A 4 inch butterfly valve, for example, has a much greater full-open $C_v$ rating than a 4 inch globe valve,

[25] The *ISA Handbook of Control Valves* cites this equation as being valid for conditions where the valve's downstream pressure ($P_2$) is equal to or greater than one-half the upstream pressure ($P_1$), with both pressures expressed in absolute units. In other words, $P_2 \geq 0.5P_1$ or $P_1 \leq 2P_2$. An upstream:downstream pressure ratio in excess of 2:1 usually means flow through a valve will become *choked*.

due to the much more direct path it offers to a moving fluid. A simple comparison of these two valve types clearly shows why this is true (note the "constriction" points labeled with arrows) as seen in Figure 16.67.

A globe valve is simply more efficient at generating fluid turbulence – and therefore dissipating fluid kinetic energy – than a butterfly valve of the same pipe size, because the globe valve design forces the fluid to change direction more often and in different ways.

One way to help quantify a particular valve design's ability to throttle fluid flow is to express this ability as a ratio of flow coefficient ($C_v$) versus cross-sectional pipe area. The basic principle here is to expect the $C_v$ of any particular valve design to be proportional to pipe area (e.g., a ball valve with twice the pipe area should have twice the flow capacity, all other factors being equal), and therefore a ratio of these two quantities should be fairly constant for any valve design. Since area of a pipe is proportional to the square of either radius or diameter ($A = \frac{\pi d^2}{4}$ or $A = \pi r^2$), this ratio may be simplified by omitting all constants such as $\pi$ and simply relating $C_v$ factor to the square of pipe diameter ($d^2$). This ratio is called the *relative flow capacity*, or $C_d$:

$$C_d = \frac{C_v}{d^2}$$

Several valve capacity factors ($C_d$) for different control valve types are shown here, assuming full-area trim and a full-open position[26]:

| *Valve design type* | $C_d$ |
|---|---|
| Single-port globe valve, ported plug | 9.5 |
| Single-port globe valve, contoured plug | 11 |
| Single-port globe valve, characterized cage | 15 |
| Double-port globe valve, ported plug | 12.5 |
| Double-port globe valve, contoured plug | 13 |
| Rotary ball valve, segmented | 25 |
| Rotary ball valve, standard port (diameter $\approx 0.8d$) | 30 |
| Rotary butterfly valve, $60^o$, no offset seat | 17.5 |
| Rotary butterfly valve, $90^o$, offset seat | 29 |
| Rotary butterfly valve, $90^o$, no offset seat | 40 |

As can be seen from a comparison of $C_d$ values, a no-offset butterfly valve has nearly *4 times* the flow capacity of a single-ported contoured-plug globe valve of the same pipe size ($C_d = 40$ versus $C_d = 11$). This makes butterfly valves advantageous in applications where large flow capacities must be achieved at minimal cost, such as in air handling (HVAC) systems for commercial buildings and combustion air controls for large industrial burners.

### 16.3.11 Control valve characterization

When control valves are tested in a laboratory setting, they are connected to a piping system that is able to provide a nearly constant pressure difference between upstream and downstream ($P_1 - P_2$). With a fluid of constant density and a constant pressure drop across the valve, flow rate becomes a

[26]Source for $C_d$ factors: Chapter 4.17: Valve Sizing of Béla Lipták's *Instrument Engineer's Handbook: Process Control and Optimization*

direct function of flow coefficient ($C_v$). This is clear from an examination of the basic valve capacity equation:

$$Q = C_v \sqrt{\frac{P_1 - P_2}{G_f}}$$

(If pressures and specific gravity are constant . . .)

$$Q = kC_v$$

As discussed in an earlier section of this chapter (see page 704), the amount of "resistance" offered by a restriction of any kind to a turbulent fluid depends on the cross-sectional area of that restriction and also the proportion of fluid kinetic energy dissipated in turbulence. If a control valve is designed such that the combined effect of these two parameters vary linearly with stem motion, the $C_v$ of the valve will likewise be proportional to stem position. That is to say, the $C_v$ of the control valve will be approximately half its maximum rating with the stem position at 50%; approximately one-quarter its maximum rating with the stem position at 25%; and so on.

If such a valve is placed in a laboratory flow test piping system with constant differential pressure and constant fluid density, the relationship of flow rate to stem position will be linear:

Figure 16.73: A linear valve's inherent characteristic

However, most real installations do not place the control valve under the same conditions. Due to frictional pressure losses in piping and changes in supply/demand pressures that vary with flow rate, a typical control valve "sees" substantial changes in differential pressure as its controlled flow rate changes. Generally speaking, the pressure drop available to the control valve will *decrease* as flow rate *increases*.

The result of this pressure drop versus flow relationship is that the actual flow rate of the same valve installed in a real process will *not* linearly track valve stem position. Instead, it will "droop" as the valve is further opened. This "drooping" graph is called the valve's *installed characteristic*, in contrast to the *inherent characteristic* exhibited in the laboratory with constant pressure drop:

Figure 16.74: A linear valve's installed characteristic

Each time the stem lifts up a bit more to open the valve trim further, flow increases, but not as much as at lower-opening positions. It is a situation of *diminishing returns*, where there are still increases in flow as the stem lifts up, but to a lesser and lesser degree.

This concept of "installed characteristic" is often elusive for many students to grasp. In the interest of clarifying the concept, here are a pair of contrasting scenarios using realistic numbers.

First, imagine a control valve installed at the base of a dam, letting water out of the reservoir. Given a constant height of water in the reservoir, the upstream (hydrostatic) pressure at the valve will likewise be constant. Let's assume this constant upstream pressure will be 20 psig (corresponding to approximately 46 feet of water column above the valve inlet). With the valve discharging into the air, downstream pressure will essentially be zero. This set of upstream and downstream conditions guarantees a constant pressure drop of 20 psig across our control valve at all times, for all flow conditions:

Figure 16.75: A control valve letting water out of a reservoir

Furthermore, assume the control valve has a "linear" inherent characteristic and a maximum flow capacity ($C_v$ rating) of 18. This means the valve's $C_v$ will be 18 at 100% open, 13.5 at 75% open, 9 at 50% open, 4.5 at 25% open, and 0 at fully closed (0% open). The behavior of this control valve may be plotted at these four stem positions by graphing the amount of flow through the valve for varying degrees of pressure drop across the valve. The result is a set of *characteristic curves*[27] for our hypothetical control valve:

Figure 16.76: Flow as a function of differential pressure for various valve open positions

[27] For those readers with an electronics background, the concept of "characteristic curves" for a control valve is *exactly* the same as that of characteristic curves for transistors. Instead of plotting the amount of current a transistor will pass given varying amounts of supply voltage, this is plotting how much water a valve will flow given varying amounts of supply pressure.

Each curve on the graph traces the amount of flow through the valve at a constant stem position, for different amounts of applied pressure drop. For example, looking at the curve representing 50% open ($C_v = 9$), the valve should flow about 42 gpm at 22 psig, about 35 gpm at 15 psig, about 20 gpm at 5 psig, and so on. Of course, these same flow figures can be obtained just by evaluating the formula $Q = C_v\sqrt{\Delta P}$ (which is in fact what was used to plot these curves), but the point here is to learn how to interpret the graph.

This set of characteristic curves may be used to determine how this valve will respond in *any* installation by superimposing another curve on the graph called a *load line*[28], describing the pressure drop available to the valve at different flow rates. Since the hypothetical dam applies a constant 20 psig across the control valve for all flow conditions, the load line for the dam will be a vertical line at 20 psig:

Figure 16.77: Flow at a constant differential pressure for various valve open positions

By noting the points of intersection[29] between the valve's characteristic curves and the load line, the flow rates from the dam may be determined at those stem positions:

[28] Once again, the exact same concept applied in transistor circuit analysis finds application here in control valve behavior!

[29] Load line plots are a graphical method of solving nonlinear, simultaneous equations. Since each curve represents a set of solutions to a particular equation, the intersection of two curves represents values uniquely satisfying both equations at the same time.

| Opening (%) | $C_v$ | Flow rate (gpm) |
|---|---|---|
| 0 | 0 | 0 |
| 25 | 4.5 | 20.1 |
| 50 | 9 | 40.2 |
| 75 | 13.5 | 60.4 |
| 100 | 18 | 80.5 |

If *this* table were to be graphed, plotting flow versus stem position, a very linear graph would result. Note how 50% open gives twice as much flow as 25% open, and 100% open nearly twice as much flow as 50% open. This says the control valve will respond linearly when pressed into service on this dam, with a constant pressure drop.

But what if the scenario is altered so that the pressure drop across the valve does *not* remain constant as flow through the valve changes? Suppose the valve is not closely coupled to the dam, but rather receives water through a narrow (restrictive) pipe:

Figure 16.78: A control valve letting water out of a reservoir through a pipe

In this installation, the narrow pipe drops pressure of its own due to friction between the rushing water and the interior walls, leaving less upstream pressure at the valve with greater amounts of flow. The control valve still drains to atmosphere, so its downstream pressure is still a constant 0 psig, but now its upstream pressure will diminish as flow increases. How will this affect the valve's performance?

Turn to the same set of characteristic curves for an answer to this question. Plot a new load line describing the pressure available to the valve at different flow rates, and once again look for the points of intersection between this load line and the valve's characteristic curves. For the sake of our hypothetical example, an arbitrary "load line" (actually a load *curve*) has been sketched showing how the valve's pressure falls off as flow rises:

Figure 16.79: Flow at a varying differential pressure for various valve open positions

Now there is a definite nonlinearity in the control valve's behavior. No longer does a doubling of stem position (from 25% to 50%, or from 50% to 100%) result in a doubling of flow rate[30]:

| Opening (%) | $C_v$ | Flow rate (gpm) |
|---|---|---|
| 0 | 0 | 0 |
| 25 | 4.5 | 18.7 |
| 50 | 9 | 35.5 |
| 75 | 13.5 | 48 |
| 100 | 18 | 57 |

Plotting the valve's performance in both scenarios (close-coupled to the dam, versus at the end of a restrictive pipe) shows the difference very clearly:

[30]Not only is the response of the valve altered by this degradation of upstream pressure, but the load line also shows that a certain maximum flow rate has been asserted by the narrow pipe which did not previously exist: 75 gpm. Even if the control valve was unbolted from the pipe and water gushed freely into the atmosphere, the flow rate would saturate at only 75 gpm because that is the amount of flow where all 20 psig of hydrostatic "head" is lost to friction in the pipe. Contrast this against the close-coupled scenario, where the load line was vertical on the graph, implying no theoretical limit to flow at all! With an absolutely constant upstream pressure, the only limit on flow rate was the maximum $C_v$ of the valve (analogous to a perfect electrical voltage source with zero internal resistance, capable of sourcing any amount of current to a load).

Figure 16.80: Comparison of a valve's inherent and installed characteristic

The "drooping" graph shows how the valve responds when it does not receive a constant pressure drop throughout the flow range. This is how the valve responds when *installed* in a non-ideal process, compared to the straight-line response it exhibits under *ideal* conditions of constant pressure. This what is meant by "installed" characteristic versus "ideal" or "inherent" characteristic.

Pressure losses due to fluid friction as it travels down pipe is just one cause of valve pressure changing with flow. Other causes exist as well, including pump curves and frictional losses in other system components such as filters and heat exchangers. Whatever the cause, any piping system that fails to provide constant pressure across a control valve will "distort" the valve's inherent characteristic in the same "drooping" manner, and this must be compensated in some way if a linear response is desired from the valve.

Not only does the diminishing pressure drop across the valve mean the same full-open flow rate cannot be achieved as in the laboratory (with a constant pressure drop), but it also means the control valve responds differently at various points along its range. Note how the installed characteristic graph is relatively steep at the beginning where the valve is nearly closed, and how the graph is almost flat at the end where the valve is nearly full-open. The rate of response (rate-of-change of flow $Q$ compared to stem position $x$, which may be expressed as the derivative $\frac{dQ}{dx}$) is much greater at low flow rates than it is at high flow rates, all due to diminished pressure drop at higher flow rates. This means the valve will respond more "sensitively" at the low end of its travel and more "sluggishly" at the high end of its travel.

From the perspective of a flow control system, this varying valve responsiveness means the system will be unstable at low flow rates and slow-responding at high flow rates. At low flow rates, there the valve is nearly closed, any small movement of the valve stem will have a relatively large effect on flow. However, at high flow rates, a much greater stem motion will be required to effect the same

change in flow. Thus, the control system will tend to over-react at low flow rates and under-react at high flow rates. Oscillations may occur at low flow rates, and large deviations from setpoint at high flow rates as a result of this "distorted" valve behavior.

The root cause of the problem – a varying pressure drop caused by frictional losses in the piping and other factors – generally cannot be eliminated. This means there is no way to regain maximum flow capacity short of replacing the control valve with one having a greater $C_v$ rating[31]. However, there is a clever way to flatten the valve's responsiveness to achieve a more linear characteristic, and that is to purposely design the valve such that its inherent characteristic complements the process "distortion" caused by changing pressure drop. In other words, the control valve trim is designed so it opens up gradually during the initial stem travel (near the closed position), then opens up more rapidly during the final stages of stem travel (near the full-open position). With the valve made to open up in a nonlinear fashion inverse to the "droop" caused by the installed pressure changes, the two non-linearities should cancel each other and yield a more linear response.

This re-design will give the valve a nonlinear characteristic when tested in the laboratory with constant pressure drop, but the installed behavior should be more linear:

Figure 16.81: A control valve with a linear installed characteristic

Now, control system response will be consistent at all points within the controlled flow range, which is a significant improvement over the original state of affairs.

Control valve trim is manufactured in a variety of different "characteristics" to provide the desired installed behavior. The two most common inherent characteristics are *linear* and *equal percentage*. "Linear" valve trim exhibits a fairly proportional relationship between valve stem travel and flow capacity ($C_v$), while "equal percentage" trim is decidedly nonlinear. A control valve with "linear" trim will exhibit consistent responsiveness only with a constant pressure drop, while "equal percentage" trim is designed to counter-act the "droop" caused by changing pressure drop when installed in a process system.

[31] Even then, achieving the ideal maximum flow rate may be impossible. Our previous 100% flow rate for the valve was 80.5 gpm, but this goal has been rendered impossible by the narrow pipe, which according to the load line limits flow to an absolute maximum of 75 gpm (even with an infinitely large control valve).

Another common inherent valve characteristic available from manufacturers is *quick-opening*, where the valve's $C_v$ increases dramatically during the initial stages of opening, but then increases at a much slower rate for the rest of the travel. Quick-opening valves are often used in on/off or pressure-relief applications, where it is important to rapidly establish flow rate during the initial portions of valve stem travel.

The standard "textbook" comparison of quick-opening, linear, and equal-percentage valve characteristics usually looks something like the following graph:

Figure 16.82: Idealized quick-opening, linear, and equal percentage inherent characteristics

A graph showing valve characteristics taken from actual manufacturers' data on valve performance[32] shows a more moderate picture:

[32]Data for the three graphs were derived from actual $C_v$ factors published in Fisher's ED, EAD, and EDR sliding-stem control valve product bulletin (51.1:ED). The exact data was not copied, however; it was "normalized" so all three valves would have the exact same full-open $C_v$ rating of 50.

Figure 16.83: Quick-opening, linear, and equal percentage inherent characteristics that are more realistic

Different valve characterizations may be achieved by re-shaping the valve trim. For instance, the plug profiles of a single-ported, stem-guided globe valve may be modified to achieve the common quick-opening, linear, and equal-percentage characteristics:

Figure 16.84: Valve plug trims for quick-opening, linear, and equal-percentage inherent characteristics

Photographs of linear (left) and equal-percentage (right) globe valve plugs are shown side-by-side for comparison:

(a) Linear

(b) Equal %

Figure 16.85: Valve trims

Cage-guided globe valve trim characteristic is a function of port shape. As the plug rises up, the amount of port area uncovered determines the shape of the characteristic graph:

Figure 16.86: Valve cage trims for quick-opening, linear, and equal-percentage inherent characteristics

Ball valve trim characteristic is a function of notch shape. As the ball rotates, the amount of notch area opened to the fluid determines the shape of the characteristic graph. All valve trim in the following illustration is shown approximately half-open (50% stem rotation):

Figure 16.87: Valve ball trims for quick-opening, linear, and equal-percentage inherent characteristics

A different approach to valve characterization is to use a non-linear positioner function instead of a non-linear trim. That is, by "programming" a valve positioner to respond in a characterized fashion to command signals, it is possible to make an inherently linear valve behave as though it were quick-opening, equal-percentage, or anywhere in between. All the positioner does is modify the valve stem position as per the desired characteristic function instead of proportionally follow the signal as it normally would.

This approach has the distinct advantage of convenience (especially if the valve is already equipped with a positioner) over changing the actual valve trim. However, if valve stem friction ever becomes a problem, its effects will be disproportionate along the valve travel range, as the positioner must position the valve more precisely in some areas of travel than others when pressed into service as a characterizer.

### 16.3.12 Control valve problems

Control valves are subject to a number of common problems. This section is dedicated to an exploration of the more common control valve problems, and potential remedies.

#### Mechanical friction

Control valves are mechanical devices having moving parts, and as such they are subject to *friction*, primarily between the valve stem and the stem packing. Some degree of friction is inevitable in valve packing[33], and the goal is to minimize friction to a bare minimum while still maintaining a pressure-tight seal.

In physics, friction is classified as either *static* or *dynamic*. Static friction is defined as frictional force holding two stationary objects together. Dynamic friction is defined as frictional force impeding the motion of two objects sliding past each other. Static friction is almost always greater in magnitude than dynamic friction. Anyone who has ever pulled a sled through snow or ice knows that more force is required to "break" the sled loose from a stand-still (static friction) than is required to keep it moving (dynamic friction). The same holds true for packing friction in a control valve: the amount of force required to initially overcome static friction between the valve stem and the packing usually exceeds the amount of force required to maintain a constant speed between a moving valve stem and a stationary packing.

The presence of packing friction in a control valve increases the force necessary from the actuator to cause valve movement. If the actuator is electric or hydraulic, the only real problem with increased force is the additional energy required from the actuator to move the valve (recall that mechanical *work* is the product of force and parallel displacement). If the actuator is pneumatic, however, a more serious problem arises from the combined effects of static and dynamic friction.

A simple "thought experiment" illustrates the problem. Imagine an air-to-open, sliding-stem control valve with bench-set pressure applied to the pneumatic actuator. This should be the amount of pressure where the valve is just about to open from a fully-closed position. Now imagine slowly increasing the air pressure applied to the actuator. What should this valve do? If the spring tension is set properly, and there is negligible friction in the valve, the stem should smoothly rise from the fully-closed position as pressure increases beyond the bench-set pressure. However, what will this valve do if there is substantial friction present in the packing assembly? Instead of the stem smoothly lifting immediately as pressure exceeds the bench-set value, this valve will remain fully closed until

[33] Bellows seals are virtually frictionless, but this is arguably a special case.

enough *extra* pressure has accumulated in the actuator to generate a force large enough to overcome spring tension *plus* packing friction. Then, once the stem "breaks free" from static friction and begins to move, the stem will begin to accelerate because the actuator force now *exceeds* the sum of spring tension and friction, since dynamic friction is less than static friction. Compressed air trapped inside the actuator acts like a spring of its own, releasing stored energy. As the stem moves, however, the chamber volume in the diaphragm or piston actuator increases, causing pressure to drop, which causes the actuating force to decrease. When the force decreases sufficiently, the stem stops moving and static friction "grabs" it again. The stem will remain stationary until the applied pressure increases sufficiently again to overcome static friction, then the "stick-slip" cycle repeats.

A graph of the mechanical response of a pneumatic actuator with substantial stem friction looks something like this:

Figure 16.88: Valve showing stick-slip characteristics

What should be a straight, smooth line is reduced to a series of "stair-steps" as the combined effect of static and dynamic friction, plus the dynamic effects of a pneumatic actuator, conspire to make precise stem positioning nearly impossible. This effect is commonly referred to as *stiction*.

Even worse is the effect friction has on valve position when trying to *reverse* the direction of pressure change. Suppose that after some new valve position has been reached in the opening direction, the pneumatic pressure begins to ramp downward. Due to static friction (again), the valve will *not* immediately respond by moving in the closed direction. Instead, it will hold still until enough pressure has been released to diminish actuator force to the point where there is enough unbalanced spring force to overcome static friction in the downward direction. Once this static friction is overcome, the stem will begin to accelerate downward because (lesser) dynamic friction will have replaced (greater) static friction. As the stem moves, however, air volume inside the actuating diaphragm or piston chamber will decrease, causing the contained air pressure to rise.

Once this pressure rises enough that the stem stops moving downward, static friction will again "grab" the stem and hold it still until enough of a pressure change is applied to the actuator to overcome static friction.

What may not be immediately apparent in this second "thought experiment" is the amount of pressure change required to cause a reversal in stem motion compared to the amount of pressure change required to provoke continued stem motion in the same direction. In order to reverse the direction of stem motion, not only does the static friction have to be "relaxed" from the last movement, but additional static friction must be overcome in the opposite direction before the stem is able to move that way. To use numerical quantities, if pressure increments of 0.5 psig are required to repeatedly overcome static friction in the upward (opening) direction, a pressure decrement of approximately twice that (1.0 psig) will be required to make the stem go downward even just a bit. Pressure decrements of 0.5 psig should be sufficient to continue downward motion after the reversal if static friction is assumed to be symmetrical for the valve.

Thus, the effects of friction on a pneumatic control valve actuator are most severe (and most easily detected) by measuring the actuator pressure change required to reverse the direction of stem motion.

Short of performing a rebuild on a "sticky" control valve to replace a damaged stem and/or packing, there is not much that may be done to improve valve stiction than regular lubrication of the packing (if appropriate). Lubrication is applied to the packing by means of a special *lubricator* device threaded into the bonnet of the valve as shown in Figure 16.32.

As one of the common sources of excessive packing friction is over-tightening of the packing nuts by maintenance personnel eager to prevent process fluid leaks, a great deal of trouble may be avoided simply by educating the maintenance staff as to the "care and feeding" of control valve packing for long service life.

Many modern digital valve positioners have the ability to monitor the drive force applied by an actuator on a valve stem, and correlate that force against stem motion. Consequently, it is possible to perform highly informative diagnostic tests on a control valve's mechanical "health," at least with regard to friction[34]. For pneumatic and hydraulic actuators, actuator force is a simple and direct function of fluid pressure applied to the piston or diaphragm. For electric actuators, actuator force is an indirect function of electric motor current, or may be directly measured using load cells or springs and displacement sensors in the gear mechanism.

The following graph illustrates the kind of diagnostic "audit" that may be obtained from a digital control valve positioner based on actuator force (pneumatic air pressure) and stem motion:

[34] Other measures of a control valve's mechanical status, such as flow capacity, flow characterization, and seat shut-off, cannot be inferred from measurements of actuator force and stem position.

Figure 16.89: Valve positioner data showing hysteresis characteristics

This same diagnostic tool is useful for detecting trim seating problems in valve designs where there is sliding contact between the throttling element and the seat near the position of full closure (e.g., gate valves, ball valves, butterfly valves, plug valves, etc.). The force required to "seat" the valve into the fully-closed position will naturally be greater than the force required to move the throttling element during the rest of its travel, but this additional force should be smooth and consistent on the graph. A "jagged" force/travel graph near the fully-closed position indicates interference between the moving element and the stationary seat, providing information valuable for predicting the remaining service life of the valve before the next rebuild.

**Flashing**

When a fluid passes through the constrictive passageways of a control valve, its average velocity increases. This is predicted by the Law of Continuity, which states that the product of fluid density ($\rho$), cross-sectional area of flow ($A$), and velocity ($v$) must remain constant for any flowstream:

$$\rho_1 A_1 \overline{v_1} = \rho_2 A_2 \overline{v_2}$$

This holds true for the control valves as they throttle the flow rate of a fluid by forcing it to pass through a narrow constriction. As fluid velocity increases through the constrictive passages of a control valve, the fluid molecules' kinetic energy increases. In accordance with the Law of Energy

Conservation, potential energy in the form of fluid pressure must decrease correspondingly. Thus, fluid pressure decreases within the constriction of a control valve's trim as it throttles the flow, then increases (recovers) after leaving the constrictive passageways of the trim and entering the wider areas of the valve body:

Figure 16.90: Pressure profile as a function of position in a control valve

If the fluid being throttled by the valve is a liquid (as opposed to a gas or vapor), and its absolute pressure ever falls below the vapor pressure of that substance, the liquid will begin to boil. This phenomenon, when it happens inside a control valve, is called *flashing*. As the graph shows, the point of lowest pressure inside the valve (called the *vena contracta* pressure, or $P_{vc}$) is the location where flashing will first occur, if it occurs at all.

Flashing is almost universally undesirable in control valves. The effect of boiling liquid at the point of maximum constriction is that flow through the valve becomes "choked" by the rapid expansion of liquid to vapor as it boils, severely inhibiting the total flow rate allowed through the valve. Flashing is also destructive to the valve trim, as boiling action propels tiny droplets of liquid at extremely high velocities past the plug and seat faces, eroding the metal over time.

A photograph showing a badly eroded valve plug (from a cage-guided globe valve) reveals just how destructive flashing can be:

Figure 16.91: Erosion of the plug is evidence of cavitation in a control valve

A characteristic effect of flashing in a control valve is a "hissing" sound, reminiscent of what sand might sound like if it were flowing through the valve.

One of the most important performance parameters for a control valve with regard to flashing is its *pressure recovery factor*, $F_L$. This factor compares the valve's total pressure drop from inlet to outlet versus the pressure drop from inlet to the point of minimum pressure within the valve.

$$F_L = \sqrt{\frac{P_1 - P_2}{P_1 - P_{vc}}}$$

Where,
$F_L$ = Pressure recovery factor (unitless)
$P_1$ = Absolute fluid pressure upstream of the valve
$P_2$ = Absolute fluid pressure downstream of the valve
$P_{vc}$ = Absolute fluid pressure at the *vena contracta* (point of minimum fluid pressure within the valve)

The following set of illustrations shows three different control valves exhibiting the same permanent pressure drop ($P_1 - P_2$), but having different values of $F_L$:

Figure 16.92: Control valves with different pressure recovery factors

Valve #1 exhibits the greatest pressure recovery (the amount that fluid pressure *increases* from the minimum pressure at the vena contracta to the downstream pressure: $P_2 - P_{vc}$) and the lowest $F_L$ value. It is also the valve most prone to flashing in liquid service, because the vena contracta pressure is so much lower (all other factors being equal) than in the other two valves. If any valve is going to reduce fluid pressure to the point where it spontaneously flashes to vapor, it would be valve #1.

Valve #3, by contrast, has very little pressure recovery, and an $F_L$ value nearly equal to 1. From the perspective of avoiding flashing, it is the best of the three valves to use for liquid service.

The style of valve (ball, butterfly, globe, etc.) is very influential on pressure recovery factor. The more convoluted the path for fluid within a control valve, the more opportunities that fluid will have to dissipate energy in turbulent motion, resulting in the greatest permanent pressure drop for the least amount of restriction at any single point in the flow's path.

Compare these two styles of valve to see which will have lowest pressure recovery factor and therefore be most prone to flashing:

Figure 16.93: Comparison of pressure loss distribution

Clearly, the globe valve does a better job of evenly distributing pressure losses throughout the path of flow. By contrast, the butterfly valve can only drop pressure at the points of constriction between the disk and the valve body, because the rest of the valve body is a straight-through path for fluid offering little restriction at all. As a consequence, the butterfly valve experiences a much lower vena contracta pressure (i.e., greater pressure recovery, and a lower $F_L$ value) than the globe valve for any given amount of permanent pressure loss, making the butterfly valve more prone to flashing than the globe valve with all other factors being equal.

### Cavitation

Fluid passing through a control valve experiences changes in velocity as it enters the narrow constriction of the valve trim (increasing velocity) then enters the widening area of the valve body downstream of the trim (decreasing velocity). These changes in velocity result in the fluid molecules' kinetic energies changing as well, in accordance with the kinetic energy equation $E_k = \frac{1}{2}mv^2$. In order that energy be conserved in a moving fluid stream, any increase in kinetic energy due to increased velocity must be accompanied by a complementary decrease in potential energy, usually in the form of fluid pressure. This means the fluid's pressure will fall at the point of maximum constriction in the valve (the *vena contracta*, at the point where the trim throttles the flow) and rise again (or *recover*) downstream of the trim as shown in Figure 16.90.

If fluid being throttled is a liquid, and the pressure at the vena contracta is less than the vapor pressure of that liquid at the flowing temperature, the liquid will spontaneously boil. This is the phenomenon of *flashing* previously described. If, however, the pressure recovers to a point greater than the vapor pressure of the liquid, the vapor will re-condense back into liquid again. This is

called *cavitation*.

As destructive as flashing is to a control valve, cavitation is worse. When vapor bubbles recondense into liquid they often do so asymmetrically, one side of the bubble collapsing before the rest of the bubble. This has the effect of translating the kinetic energy of the bubble's collapse into a high-speed "jet" of liquid in the direction of the asymmetrical collapse. These liquid "microjets" have been experimentally measured at speeds up to 100 meters per second (over 320 feet per second). What is more, the pressure applied to the surface of control valve components in the path of these microjets is immense. Each microjet strikes the valve component surface over a very small surface area, resulting in a very high pressure ($P = \frac{F}{A}$) applied to that small area. Pressure estimates as high as 1500 newtons per square millimeter (1.5 *giga*-Pascals, or about 220,000 psig!) have been calculated for control valve applications involving water.

No substance known is able to continuously withstand this form of abuse, meaning that cavitation *will destroy* any control valve given enough time. The effect of each microjet impinging on a metal surface is to carve out a small pocket in that metal surface. Over time, the metal will begin to take on a "pock-marked" look over the area where cavitation occurs. This stands in stark contrast to the visual appearance of flashing damage, which is smooth and polished.

The sound made by substantial liquid cavitation also contrasts starkly against the sound made by flashing. Whereas flashing sounds as though sand were flowing through the valve, cavitation produces a much louder "crackling" sound comprised of distinct impact pulses, reminiscent of what gravel or rocks might sound like if they were somehow forced to flow through the valve.

Sustained cavitation also has the detrimental effect of accelerating corrosion in certain process services. Bare metal surfaces are highly reactive with many chemical fluids, but become more resistant to further attack when a thin layer of reacted metal on the surface (the so-called *passivation layer*) acts as a sort of chemical barrier. Rust on steel, or the powdery-white oxide of aluminum are good examples: the initially bare metal surfaces react with their surrounding environment to form a protective outer layer, impeding further degradation of the metal beneath that layer. Cavitation works to blast away any protective layer that might otherwise accumulate, allowing corrosion to work at full speed until the entire thickness of the metal is corroded through. The complementary destructive actions of cavitation and corrosion together is sometimes referred to as *cavitation corrosion*.

Several methods exist for abating cavitation in control valves:

1. Prevent flashing in the first place
2. Cushion with introduced gas
3. Sustain flashing action

Cavitation abatement method #1 is quite simple to understand: if flashing is prevented from ever happening in a control valve, cavitation cannot follow. The key to doing this is making sure the vena contracta pressure never falls below the vapor pressure for the liquid. Several techniques exist for doing this:

- Select a control valve type having less pressure recovery (i.e., greater $F_L$ value)
- Increase both upstream and downstream pressures by relocating the valve to a higher-pressure location in the process.
- Use multiple control valves in series to reduce the lowest pressure at either one
- Decrease the liquid's temperature (this decreases vapor pressure)
- Use cavitation-control valve trim

The last suggestion in this list deserves further exploration. Valve trim may be specially designed for cavitation abatement by providing multiple stages of pressure drop for the fluid as it passes through the trim. The following is a pressure versus location graph for a cavitating control valve. The liquid's vapor pressure is shown here as a dashed line marked $P_{vapor}$:

Figure 16.94: Pressure as a function of location in a cavitating valve

A valve equipped with cavitation-control trim will have a different pressure profile, with multiple *vena contracta* points where the fluid passes through a series of constrictions within the trim itself:

Figure 16.95: Pressure drops within the anti-cavitation trim of a valve

This way, the same final permanent pressure drop $(P_1 - P_2)$ may be achieved without the lowest pressure ever falling below the liquid's vapor pressure limit. An example of cavitation-control design applied to cage-guided globe valve trim is shown in the following illustration:

Figure 16.96: Cavitation-control design in cage-guided globe valve trim

Cavitation abatement method #2 is practical only in some process applications, where a non-reacting gas may be injected into the liquid stream to provide some "cushioning" within the cavitating region. The presence of non-condensible gas bubbles in the liquid stream disturbs the microjets' pathways, helping to dissipate their energy before striking the valve body walls.

Cavitation abatement method #3 involves a strategy opposite that of method #1. If, for whatever reason, there is no way to avoid the pressure falling below the vapor pressure of the liquid as the flow stream moves through the valve, the downstream liquid pressure may be able to be prevented from rising above the liquid's vapor pressure, at least until the fluid clears past the valuable control valve and into an area of the system where cavitation damage will not be so expensive. This avoids cavitation at the cost of guaranteed flashing within the control valve, which is generally not as destructive as cavitation. However, be prepared for some piping to shake!

A pressure diagram shows how this method works:

Figure 16.97: A diagram showing cavitation abatement

Of course, flashing is not good for a control valve either. Not only does it damage the valve over time, but it also causes problems with flow capacity, as will be explored next.

### Choked flow

Both gas and liquid control valves may experience what is generally known as *choked flow*. Simply put, "choked flow" is a condition where the rate of flow through a valve does not change substantially as downstream pressure is reduced.

Ideally, turbulent fluid flow rate through a control valve is a simple function of valve flow capacity ($C_v$) and differential pressure drop ($P_1 - P_2$), as described by the basic valve flow equation:

$$Q = C_v \sqrt{\frac{P_1 - P_2}{G_f}}$$

In a gas control valve, choking occurs when the velocity of the gas reaches the speed of sound for that gas. This is often referred to as *critical* or *sonic* flow. In a liquid control valve, choking occurs with the onset of flashing[35]. The reason sonic velocity is relevant to flow capacity for a control valve has to do with the propagation of pressure changes in fluids. Pascal's Law says that changes in pressure within a closed fluid system will manifest at all points in the fluid system,

[35]The *Control Valve Sourcebook – Power & Severe Service* on page 6-3 and the *ISA Handbook of Control Valves* on page 211 both suggest that the mechanism for choking in liquid service may be related to the speed of sound just as it is for choked flow in gas services. Normally, liquids have higher sonic velocities than gases due to their far greater densities. This makes choking due to sonic velocity very unlikely in liquid flowstreams. However, when a liquid flashes into vapor, the speed of sound *for that two-phase mixture of liquid and vapor* will be much less than it is for the liquid itself, opening up the possibility of sonic velocity choking.

but this never happens instantaneously. Instead, pressure changes propagate through any fluid at the speed of sound within that fluid. If a fluid stream happens to move at or above the speed of sound, pressure changes downstream are simply not able to overcome the stream's velocity to affect anything upstream, which explains why the flow rate through a control valve experiencing sonic (critical) flow velocities does not change with changes in downstream pressure: those downstream pressure changes cannot propagate upstream against the fast-moving flow, and so will have no effect on the flow as it accelerates to sonic velocity at the point(s) of constriction.

Choked flow conditions become readily apparent if the flow-versus-pressure function of a control valve at any fixed opening value is graphed. The basic valve flow equation predicts a perfectly straight line at constant slope with flow rate ($Q$) as the vertical variable and the square root of pressure drop ($\sqrt{P_1 - P_2}$) as the horizontal variable. However, if a control valve is actually tested by holding its upstream liquid pressure ($P_1$) constant and varying its downstream pressure ($P_2$) while maintaining a fixed stem position, there is a point where flow reaches a maximum limit value:

Figure 16.98: Flow vs. pressure showing choked flow in a control valve

In a choked flow condition, further reductions in downstream pressure achieve no greater flow of liquid through the valve. This is not to say that the valve has reached a maximum flow – the flow rate through a choked valve may still be increased by increasing its upstream pressure. However, more flow cannot be coaxed through a choked valve by decreasing its downstream pressure.

An approximate predictor of choked flow conditions for gas valve service is the upstream-to-minimum absolute pressure ratio. When the vena contracta pressure is less than one-half the upstream pressure, both measured in absolute pressure units, choked flow is virtually guaranteed. One should bear in mind that this is merely an approximation and not a precise prediction for

choked flow. A lot more information would have to be known about the valve design, the particular process gas, and other factors in order to more precisely predict the presence of choking.

Choked flow in liquid services is predicted when the vena contracta pressure equals the liquid's vapor pressure, since choking is a function of flashing for liquid flowstreams.

No attempt will be made in this book to explain sizing procedures for control valves in choked-flow service, due to the complexity of the subject.

An interesting and beneficial application of choked flow in gases is a device called a *critical velocity nozzle*. This is a nozzle designed to allow a fixed flow rate of gas through it given a known upstream pressure, and a downstream pressure that is sufficiently low to allow sonic velocities in the nozzle throat. One practical use for critical velocity nozzles is in the flow testing of compressed air systems. One or more of these nozzles are connected to the main header line of an air compressor system and allowed to vent to atmosphere. So long as the compressor(s) are able to maintain constant header pressure, the flow rate of air through the nozzles(s) is guaranteed to be fixed, allowing a technician to monitor compressor parameters under precisely known load conditions.

**Valve noise**

A troublesome phenomenon in severe services is audible noise produced by control valves. Noise output is worse for gas services experiencing sonic (critical) flow and for liquid services experiencing cavitation, although it is possible for a control valve to produce substantial noise even when avoiding these operating conditions.

### 16.3.13 Dampers and louvers

A *damper* (otherwise known as a *louver*) is a single- or multi-element flow control device generally used to isolate or to throttle large flows of air at low pressure. Dampers find common application in furnace and boiler draft control, and in HVAC (Heating, Ventilation, and Air Conditioning) systems.

Common damper designs include parallel-vane, opposed-vane, and radial vane. Parallel-vane dampers resemble a Venetian blind, with multiple rectangular vanes synchronously rotated to control flow through a rectangular opening. This type of damper is most suited in isolation applications (as opposed to throttling applications). A photograph of a parallel-vane damper is shown here, part of an induced-draft (suction) air fan system on a separator at a cement plant. The vanes are not visible in this photograph because they reside inside the metal air duct, but the actuator mechanism and linkages connecting seven vane shafts together are:

Figure 16.99: A parallel-vane damper actuator and mechanical linkage

Opposed vane dampers look a lot like their parallel-vane counterparts, but every other blade turns in the opposite direction when actuated. Opposed-vane dampers provide better throttling characteristics than parallel-vane dampers.

Radial-vane dampers use multiple vanes arranged like petals of a flower to throttle flow through a circular opening. A photograph of a radial-vane damper is shown here (note the levers and linkages on the periphery of the tube, synchronizing the motions of the eight vanes so they rotate at the same angle):

Figure 16.100: A radial-vane damper

Dampers find use in many non-industrial applications as well. Take for instance these greenhouse vents, actuated by pneumatic (air-powered) piston actuators:

Figure 16.101: A parallel-vane damper with actuator and mechanical linkage shown

## 16.4 On/off electric motor control circuits

An electric motor is often used as a discrete control element in a control system if driving a pump, conveyor belt, or other machine for the transportation of a process substance. As such, it is important to understand the functioning of on/off motor control circuits.

Of all the available electric motor types, the most common found in industrial applications (by far) is the three-phase AC induction motor. For this reason, this section of the book will focus exclusively on this type of motor as a final control element.

### 16.4.1 Two-wire motor control

A simple three-phase, 480 volt AC motor-control circuit is shown here, both in pictorial and schematic form. This entire assembly consisting of contactor, overload block, control power transformer, power fuses (or alternatively, a circuit breaker) and associated components will often be built into a motor control center cubicle that is informally referred to as a *bucket*:

Figure 16.102: A motor branch power and two-wire control circuit

Note how a *control power transformer* steps down the 480 volt AC to provide 120 volt AC power for the contactor coil to operate. Furthermore, note how the overload ("OL") contact is wired in series with the contactor coil so that a thermal overload event forces the contactor to de-energize and thus interrupt power to the motor even if the control switch is still in the "on" position. The overload heaters appear in the schematic diagram as pairs of back-to-back "hook" shapes, connected in series with the three "T" lines of the motor. Remember that these "OL" heater elements do not directly interrupt power to the motor in the event of an overload, but rather signal the "OL" contact to open up and de-energize the contactor.

In an automatic control system, the toggle switch would be replaced by another relay contact (that relay controlled by the status of a process), a process switch, or perhaps the discrete output channel of a programmable logic controller (PLC).

It should be noted that a maintained-style of switch is necessary in order for the motor to continue to run after a human operator actuates the switch. The motor runs with the switch is in the closed

state, and stops when the switch opens.

### 16.4.2 Three-wire motor control

An alternative to the two-wire design is to build a *latching* circuit allowing the use of momentary contact switches (one to start, and one to stop). A simple latching motor control circuit is shown here:

Figure 16.103: A motor branch power and three-wire control circuit

In this circuit, an *auxiliary contact* actuated by the motor contactor is wired in parallel with the "Start" pushbutton switch, so that the motor contactor continues to receive power after the operator releases the switch. This parallel contact – sometimes called a *seal-in contact* – latches the motor in an "on" state after a momentary closure of the "Start" pushbutton switch.

A normally-closed "Stop" switch provides a means to "un-latch" the motor circuit. Pressing this

pushbutton switch stops current in the coil of the contactor, causing it to de-energize, which then opens the three motor power contacts as well as the auxiliary contact that used to maintain the contactor's energized state.

A simple *ladder diagram* showing the interconnections of all components in this motor control circuit makes this system easier to understand:

Figure 16.104: A motor branch power and three-wire control circuit schematic

Most on/off motor control circuits in the United States are some variation on this wiring theme, if not identical to it. Once again, this system could be automated by replacing the "Start" and "Stop" pushbutton switches with process switches (e.g., pressure switches for an air compressor control system) to make a system that starts and stops automatically. A programmable logic controller (PLC) may also be used to provide the latching function rather than an auxiliary contact on the contactor. If a PLC is included in the motor control circuit, a great many automatic control features may be added to enhance the system's capabilities. Examples include timing functions, motor cycle count functions, and even remote start/stop capability via a digital network connecting to operator interface displays or other computers.

### 16.4.3 Manual/remote motor control

A maintained Hand-Off-Auto (HOA) switch can be used to select between local operation and remote operation of a motor. When in Hand, the motor starts and then runs continuously. When in Off, the motor shuts off. These two HOA switch positions allow manual control of the motor. When in Auto, one or more remote contacts are placed in series with the coil that allow remote operation. If the remote contact is closed, the motor will run. When the remote contact is opened or the HOA switch is moved to Off, the motor shuts off. Motor control center buckets are often provided with HOA switches.

### 16.4.4 Networked motor control

A modern trend in motor control is the use of digital networks to both command the contactor as well as monitor the motor's operating status remotely. This next photograph[36] shows a digitally monitored and controlled "bucket," using DeviceNet as the control network:

[36]This bucket was still under construction at the time the photograph was taken. As such, none of the motor leads have been connected, which is why there are no power conductors exiting the bottom of the bucket. Instead, only the three terminals ready to accept heavy-gauge motor leads are visible.

Figure 16.105: Internals of a "smart" motor bucket

Using a digital network standard such as Ethernet, DeviceNet, Modbus, Profibus, or any number of others to monitor and control a motor brings a host of benefits for maintenance and operations. Control wiring is vastly simplified with digital networks, as a single network cable is able to address multiple motor buckets. The "smart" network interface module installed in the bucket may be designed to monitor such parameters as line voltage, line current, phase imbalance, and power factor to report these values to the host control system via the network.

It is common for the network interface module inside the bucket to have its own digital display for local indication of these parameters as well. A close-up photograph of a Square-D "Motor Logic Plus" unit shows some of its locally-accessible features:

Figure 16.106: An industrial network interface module used for motor control

The PLC connected to the network is able to access all these values as well, reporting them to operations and/or maintenance personnel as desired. Instead of individual wires running between the PLC and the motor starter to command each motor to run and stop, the PLC simply transmits "start" and "stop" commands over the network to individually addressed digital starter modules. The network wiring may simply be paralleled ("daisy-chained") between units, such that several buckets reside on the same physical network, each one programmed with a unique address. A PLC connected to this same network is thus able to access and control all parameters for all motors on that network.

## 16.5 Variable-speed motor controls

An alternative method of flow control in lieu of control valves is to vary the speed of the machine(s) motivating fluid to flow. In the case of liquid flow control, this would take the form of variable-speed pumps. In the case of gas flow control, it would mean varying the speed of compressors or blowers.

Flow control by machine speed control makes a lot of sense for some process applications. It is certainly more energy-efficient to vary the speed of the machine pushing fluid to control flow, as opposed to letting the machine run at full speed all the time and adjusting flow rate by throttling the machine's discharge (outlet) or recycling fluid back to the machine's suction (inlet). The fact that the system has one less component in it (no control valve) also reduces capital investment and potentially increases system reliability:

Figure 16.107: Examples of flow control

Further advantages of machine speed control include the ability to "soft-start" the machine instead of always accelerating rapidly from a full stop to full speed, reduced wear on machines due to less motion over time, and reduced vibration. In applications such as conveyor belt control, robotic machine motion control, and electric vehicle propulsion, variable-speed technology makes perfect sense because the prime mover device is already (in most cases) an electric motor, with precise speed control of that motor providing many practical benefits. In some applications, *regenerative braking* may be of benefit, where the motor is used as an electrical generator to slow down the machine on command. Regenerative braking transfers kinetic energy within the machine to the power grid where it may be gainfully used in other processes, saving energy and reducing wear on any mechanical (friction) brakes already installed in the machine.

With all these advantages inherent to variable-speed pumps, fans, and compressors (as opposed to using dissipative control valves), one might wonder, "Why would anyone *ever* use a control valve to regulate flow? Why not just control all fluid flows using variable-speed pumping machines?" Several good answers exist to this question:

- Variable-speed machines often cannot respond as rapidly as control valves
- Control valves have the ability to positively halt flow; a stopped pump or blower will not necessarily prevent flow from going through
- Some process applications *must* contain a dissipative element in order for the system to function (e.g., let-down valves in closed refrigeration systems)
- Split-ranging may be difficult or impossible to achieve with multiple machine speed control
- Limited options for fail-safe status
- In many cases, there is no machine dedicated to a particular flow path (e.g., a pressure release valve)

### 16.5.1 DC motor speed control

DC electric motors generate torque by a reaction between two magnetic fields: one field established by stationary "field" windings (coils), and the other by windings in the rotating armature. Some DC motors lack field windings, substituting large permanent magnets in their place so that the stationary magnetic field is constant for all operating conditions.

In any case, the operating principle of a DC electric motor is that current passed through the armature creates a magnetic field that tries to align with the stationary magnetic field. This causes

the armature to rotate:

Figure 16.108: DC motor operation

However, a set of segmented copper strips called a *commutator* breaks electrical contact with the now-aligned coil and energizes another coil (or in the simple example shown above, it re-energizes the same loop of wire in the opposite direction) to create another out-of-alignment magnetic field that continues to rotate the armature. Electrical contact between the rotating commutator segments and the stationary power source is made through carbon *brushes*. These brushes wear over time (as does the commutator itself), and must be periodically replaced.

Most industrial DC motors are built with multiple armature coils, not just one as shown in the simplified illustration above. A photograph of a large (1250 horsepower) DC motor used to propel a ferry ship is shown here, with the field and armature poles clearly seen (appearing much like spokes in a wheel):

Figure 16.109: Internals of a large industrial DC motor

A close-up of one brush assembly on this large motor shows both the carbon brush, the brush's spring-loaded holder, and the myriad of commutator bars the brush makes contact with as the armature rotates:

Figure 16.110: The commutator of a DC motor is shown with one of the carbon brushes

DC motors exhibit the following relationships between mechanical and electrical quantities:

**Torque:**

- Torque is directly proportional to armature magnetic field strength, which in turn is directly proportional to current through the armature windings
- Torque is also directly proportional to the stationary pole magnetic field strength, which in turn is directly proportional to current through the field windings (in a motor with non-permanent field magnets)

**Speed:**

- Speed is limited by the counter-EMF generated by the armature as it spins through the stationary magnetic field. This counter-EMF is directly proportional to armature speed, and also directly proportional to stationary pole magnetic field strength (which is directly proportional to field winding current in a motor that is not permanent-magnet)
- Thus, speed is directly proportional to armature voltage
- Speed is also inversely proportional to stationary magnetic field strength, which is directly proportional to current through the field windings (in a motor with non-permanent field magnets)

A very simple method for controlling the speed and torque characteristics of a wound-field (non-permanent magnet) DC motor is to control the amount of current through the field winding:

Figure 16.111: Schematic of a shunt field DC motor

Decreasing the field control resistor's resistance allows more current through the field winding, strengthening its magnetic field. This will have two effects on the motor's operation: first, the motor will generate more torque than it did before (for the same amount of armature current) because there is now a stronger magnetic field for the armature to react against; second, the motor's speed will decrease because more counter-EMF will be generated by the spinning armature for the same rotational speed, and this counter-EMF naturally attempts to equalize with the applied DC source voltage. Conversely, a DC motor's speed may be increased (and its torque output reduced) by increasing the field control resistor's resistance, weakening the stationary magnetic field through which the armature spins.

Regulating field current may alter the balance between speed and torque, but it does little to control total motor *power*. In order to control the power output of a DC motor, the armature voltage and current must also be regulated. Variable resistors may also be used for this task, but this is generally frowned upon in modern times because of the wasted power.

A better solution is to have an electronic power control circuit very rapidly switch transistors on and off, switching voltage to the motor armature. This is called *pulse-width modulation*, or *PWM*.

Figure 16.112: Schematic of a shunt field DC motor with armature control

The *duty cycle* (on time versus on+off time) of the pulse waveform will determine the fraction of total power delivered to the motor:

Figure 16.113: Pulse-width modulation examples

Such an electronic power-control circuit is generally referred to as a *drive*. Thus, a *variable-speed drive* or *VSD* is a high-power circuit used to control the speed of a DC motor. Motor drives may be manually set to run a motor at a set speed, or accept an electronic control signal to vary the motor speed in the same manner an electronic signal commands a control valve to move. When equipped with remote control signaling, a motor drive functions just like any other final control element: following the command of a process controller in order to stabilize some process variable at setpoint.

An older technology for pulsing power to a DC motor is to use a *controlled rectifier* circuit, using thyristors instead of regular rectifying diodes to convert AC to DC. Since the main power source of

most industrial DC motors is AC anyway, and that AC must be converted into DC at some point in the system, it makes sense to integrate control right at the point of rectification:

Figure 16.114: Phase control scheme for a DC motor

Controlled rectifier circuits work on the principle of varying the "trigger" pulse times relative to the AC waveform pulses. The earlier the AC cycle each thyristor is triggered on, the longer it will be on to pass current to the motor. The "phase control" circuitry handles all this pulse timing and generation.

A DC motor drive that simply varied power to the motor according to a control signal would be crude and difficult to apply to the control of most processes. What is ideally desired from a variable-speed drive is precise command over the motor's *speed*. For this reason, most VSDs are designed to receive feedback from a tachometer mechanically connected to the motor shaft, so the VSD "knows" how fast the motor is turning. The tachometer is typically a small DC generator sometimes known as a *tachogenerator*, producing a DC voltage directly proportional to its shaft speed (0 to 10 volts is a common scale). With this information, the VSD may throttle electrical power to the motor as necessary to achieve whatever speed is being commanded by the control signal. Having a speed-control feedback loop built into the drive makes the VSD a "slave controller" in a cascade control system, the drive receiving a speed setpoint signal from whatever process controller is sending an output signal to it:

Figure 16.115: Diagram of a VSD controlling the speed of a DC motor

A photograph of the tachogenerators (dual, for redundancy) mechanically coupled to that large 1250 horsepower ferry ship propulsion motor appears here:

Figure 16.116: A redundant pair of tachogenerators

The thyristors switching power to this motor may be seen here, connected via twisted-pair wires to control boards issuing "firing" pulses to each thyristor at the appropriate times:

Figure 16.117: The internals of a DC VSD showing the thyristors

The integrity of the tachogenerator feedback signal to the VSD is extremely important for safety reasons. If the tachogenerator becomes disconnected – whether mechanically or electrically (it doesn't matter) – from the drive, the drive will "think" the motor is not turning. In its capacity as a speed *controller*, the drive will then send full power to the DC motor in an attempt to get it up to speed. Thus, loss of tachogenerator feedback causes the motor to immediately "run away" to full speed. This is undesirable at best, and likely dangerous in the case of motors as large as the one powering this ship.

As with all forms of electric power control based on pulse durations and duty cycles, there is a lot of electrical "noise" cast by VSD circuits. Square-edged periodic pulse waveforms created

by the rapid on-and-off switching of the semiconductor power devices can be approximated by an infinite sum of sine waves of various amplitudes and frequencies[37], some of which may be of high enough frequency to self-propagate through space as electromagnetic waves. This *radio-frequency interference* or *RFI* may be quite severe given the high power levels of industrial motor drive circuits. For this reason, it is *imperative* that neither the motor power conductors nor the conductors feeding AC power to the drive circuit be routed anywhere near small-signal or control wiring, because the induced noise *will* wreak havoc with whatever systems utilize those low-level signals.

RFI noise on the AC power conductors may be mitigated by routing the AC power through *filter* circuits placed near the drive. The filter circuits block high-frequency noise from propagating back to the rest of the AC power distribution wiring where it may influence other electronic equipment. However, there is little that may be done about the RFI noise between the drive and the motor other than to shield the conductors in well-grounded metallic conduit.

### 16.5.2 AC motor speed control

AC induction motors are based on the principle of a *rotating magnetic field* produced by a set of stationary windings (called *stator* windings) energized by AC power of different phases. The effect is not unlike a series of blinking light bulbs which appear to "move" in one direction due to intentional sequencing of the blinking. If sets of wire coils (windings) are energized in a like manner – each coil reaching its peak field strength at a different time from its adjacent neighbor – the effect will be a magnetic field that "appears" to move in one direction. If these windings are oriented around the circumference of a circle, the moving magnetic field rotates about the center of the circle, as illustrated by this sequence of images (read left-to-right, top-to-bottom, as if reading words in a sentence):

[37] This theory was put forth by Jean Baptiste Joseph Fourier (1768-1830), and is known as a *Fourier series*.

Figure 16.118: A rotating magnetic field

Any magnetized object placed in the center of this circle will attempt to spin at the same rotational speed as the rotating magnetic field. *Synchronous* AC motors use this principle, where a magnetized rotor precisely follows the magnetic field's speed.

Any electrically conductive object placed in the center of the circle will experience *induction* as the magnetic field direction changes around the conductor. This will induce electric currents within the conductive object, which in turn will react against the rotating magnetic field in such a way that the object will be "dragged along" by the field, always lagging a bit in speed. *Induction* AC motors use this principle, where a non-magnetized (but electrically conductive) rotor rotates at a speed slightly less[38] than the synchronous speed of the rotating magnetic field.

The rotational speed of this magnetic field is directly proportional to the frequency of the AC power, and inversely proportional to the number of poles in the stator:

$$S = \frac{120f}{n}$$

Where,

$S$ = Synchronous speed of rotating magnetic field, in revolutions per minute (RPM)

$f$ = Frequency, in cycles per second (Hz)

$n$ = Total number of stator poles per phase (the simplest possible AC motor design will have 2 poles)

While the number of poles in the motor's stator is a quantity fixed at the time of the motor's manufacture, the frequency of power applied may be adjusted with the proper electronic circuitry. A high-power circuit designed to produce varying frequencies for an AC motor to run on is called a *variable-frequency drive*, or *VFD*.

A simplified schematic diagram for a VFD is shown here, with a rectifier section on the left (to convert AC input power into DC), a filter capacitor to "smooth" the rectified DC power, and a transistor "bridge" to switch DC into AC at whatever frequency is desired to power the motor. The transistor control circuitry has been omitted from this diagram for the sake of simplicity:

[38] The difference between the synchronous speed and the rotor's actual speed is called the motor's *slip speed*.

Figure 16.119: Schematic of a VFD

As with DC motor drives (VSDs), the power transistors in an AC drive (VFD) switch on and off very rapidly with a varying duty cycle. Unlike DC drives, however, the duty cycle of an AC drive's power transistors must vary rapidly in order to synthesize an AC waveform from the DC "bus" voltage following the rectifier. A DC drive circuit's PWM duty cycle controls motor power, and so it will remain at a constant value when the desired motor power is constant. Not so for an AC motor drive circuit: its duty cycle must vary from zero to maximum and back to zero repeatedly in order to generate an *AC* waveform for the motor to run on.

The equivalence between a rapidly-varied pulse-width modulation (PWM) waveform and a sine wave is shown in the following illustration:

Figure 16.120: A PWM waveform overlayed on its sine wave average

This concept of rapid PWM transistor switching allows the drive to "carve" any arbitrary waveform out of the filtered DC voltage it receives from the rectifier. Virtually any frequency may be synthesized (up to a maximum limited by the frequency of the PWM pulsing), and any voltage (up to a maximum peak established by the DC bus voltage), giving the VFD the ability to power an induction motor over a wide range of speeds.

While frequency control is the key to synchronous and induction AC motor speed control, it is generally not enough on its own. While the speed of an AC motor is a direct function of frequency (controlling how fast the rotating magnetic field rotates around the circumference of the stator), torque is approximately proportional to stator winding current. Since the stator windings are inductors by nature, their reactance varies with frequency as described by the formula $X_L = 2\pi f L$. Thus, as frequency is increased, winding reactance increases right along with it. This increase in reactance results in a decreased stator current (assuming the AC voltage is held constant as frequency is increased). This can cause undue torque loss at high speeds, and excessive torque (as well as excessive stator heat!) at low speeds. For this reason, the AC voltage applied to a motor by a VFD is usually made to vary in direct proportion to the applied frequency, so that the stator current will remain within good operating limits throughout the speed range of the VFD. This correspondence is called the *voltage-to-frequency ratio*, or V/F ratio. It may also be called the *volts-per-Herz ratio*, or V/Hz ratio.

Variable-frequency motor drives are manufactured for industrial motor control in a wide range of sizes and horsepower capabilities. Some VFDs are small enough to hold in one hand, while others are large enough to require a freight train for transport. The following photograph shows a pair of moderately-sized Allen-Bradley VFDs (about 100 horsepower each, standing about 4 feet high), used to control pumps at a wastewater treatment plant:

Figure 16.121: A pair of industrial VFDs

Variable-frequency AC motor drives do not require motor speed feedback the way variable-speed DC motor drives do. The reason for this is quite simple: the controlled variable in an AC drive

is the frequency of power sent to the motor, and rotating-magnetic-field AC motors are *frequency-controlled* machines by their very nature. For example, a 4-pole AC induction motor powered by 60 Hz has a base speed of 1728 RPM (assuming 4% slip). If a VFD sends 30 Hz AC power to this motor, its speed will be approximately half its base-speed value, or 864 RPM. There is really no need for speed-sensing feedback in an AC drive, because the motor's real speed will always be limited by the drive's output frequency. To control frequency *is* to control motor speed for AC synchronous and induction motors, so no tachogenerator feedback is necessary for an AC drive to "know" approximately[39] how fast the motor is turning. The non-necessity of speed feedback for AC drives eliminates a potential safety hazard common to DC drives: the possibility of a "runaway" event where the drive loses its speed feedback signal and sends full power to the motor.

As with DC motor drives, there is a lot of electrical "noise" cast by VFD circuits. Square-edged pulse waveforms created by the rapid on-and-off switching of the power transistors can be approximated by an infinite sum of sine waves of various amplitudes and frequencies[40], some of which may be of high enough frequency to self-propagate through space as electromagnetic waves. This *radio-frequency interference* or *RFI* may be quite severe given the high power levels of industrial motor drive circuits. For this reason, it is *imperative* that neither the motor power conductors nor the conductors feeding AC power to the drive circuit be routed anywhere near small-signal or control wiring, because the induced noise *will* wreak havoc with whatever systems utilize those low-level signals.

RFI noise on the AC power conductors may be mitigated by routing the AC power through *filter* circuits placed near the drive. The filter circuits block high-frequency noise from propagating back to the rest of the AC power distribution wiring where it may influence other electronic equipment. However, there is little that may be done about the RFI noise between the drive and the motor other than to shield the conductors in well-grounded metallic conduit.

### 16.5.3 Motor drive features

Modern DC and AC motor drives provide features useful when using electric motors as final control elements. Some common features seen in both VSDs and VFDs are listed here:

- Speed limiting
- Torque limiting
- Torque profile curves (used to regulate the amount of torque available at different motor speeds)
- Acceleration (speed rate-of-change) limiting
- Deceleration (speed rate-of-change) limiting
- Dynamic braking (turning the motor into an electromagnetic brake[41])
- Plugging (applying reverse-direction power to a motor to *quickly* stop it)
- Regenerative braking (turning the motor into a generator to recover kinetic energy)

[39] For more precise control of AC motor speed (especially at low speeds where slip speed becomes a greater percentage of actual speed), speed sensors may indeed be necessary.

[40] This theory was put forth by Jean Baptiste Joseph Fourier (1768-1830), and is known as a *Fourier series*.

[41] This is accomplished in very different ways for DC versus AC motors. To dynamically brake a DC motor, the field winding must be kept energized while a high-power load resistor is connected to the armature. As the motor turns, the armature will push current through the resistor, generating a braking torque as it does. To dynamically brake an AC motor, a relatively small DC current is passed through the stator windings, causing large braking currents to be induced in the rotor.

- Overcurrent monitoring and automatic shut-down
- Overvoltage monitoring and automatic shut-down
- PWM frequency adjustment (may be helpful in reducing electromagnetic interference with some equipment)

Not only are some of these limiting parameters useful in extending the life of the motor, but they may also help extend the operating life of the mechanical equipment powered by the motor. It is certainly advantageous, for example, to have torque limiting on a conveyor belt motor, so that the motor does not apply full rated torque (i.e., stretching force) to the belt during start-up.

If a motor drive is equipped with digital network communication capability (e.g., Modbus), it is usually possible for a host system such as a PLC or DCS to update these control parameters as the motor is running.

### 16.5.4 Metering pumps

A very common method for directly controlling low flow rates of fluids is to use a device known as a *metering pump*. A "metering pump" is a pump mechanism, motor, and drive electronics contained in a monolithic package. Simply supply 120 VAC power and a control signal to a metering pump, and it is ready to use.

Metering pumps are commonly used in water treatment processes to inject small quantities of treatment chemicals (e.g., coagulants, disinfectants, acid or caustic liquids for pH neutralization, corrosion-control chemicals) into the water flowstream, as is the Milton-Roy unit shown in this photograph:

Figure 16.122: A chemical metering pump

Adjustment knobs that allow the user to vary the speed at which the pump cycles and the length of the pump's stroke are placed on the front of the pump to allow the flow rate to be varied from 0% to 100%:

Figure 16.123: The controls for a chemical metering pump

While some metering pumps use rotary motor and pump mechanisms, many use a "plunger" style mechanism operated by a solenoid at variable intervals. Thus, the latter type of metering pump does not provide continuous flow control, but rather a flow consisting of discrete pulse events distributed over a period of time. The "plunger" metering pumps are quite simple and reliable, and are entirely appropriate if non-continuous flow is permissible for the process.

# References


[1] H. D. Baumann, Control Valve Primer, A User's Guide, 2nd ed. Research Triangle Park, NC: Instrument Society of America, 1994.

[2] Compatability of Analog Signals for Electronic Industrial Process Instruments, International Society of Automation Std. 50.00.01, 2010.

[3] Cavitation in Control Valves, L351 EN, Samson AG, Frankfurt, DE, 2003.

[4] Control Valve Sourcebook – Power & Severe Service, Fisher Controls International, Marshalltown, IA, 1993.

[5] B. Grumstrup, Design ED, EAD, and EDR Sliding-Stem Control Valves, 51.1:ED, Fisher Controls International, Marshalltown, IA, 2006.

[6] ——, Considerations in the Design and Selection of Bellows Seal Equipment Valves, Technical Monograph 37, Fisher Controls International, Marshalltown, IA, 1991.

[7] J. W. Hutchison, ISA Handbook of Control Valves, 2nd ed. Research Triangle Park, NC: Instrument Society of America, 1976.

[8] J. D. Irwin, Ed., The Industrial Electronics Handbook. Boca Raton, FL: CRC Press, 1997.

[9] F. D. Jury, Fundamentals of Aerodynamic Noise in Control Valves, Technical Monograph 43, Fisher Controls International, Marshalltown, IA, 1999.

[10] B. Lipták, Ed., Instrument Engineers Handbook: Process Control and Optimization, 4th ed. Boca Raton, FL: CRC Press, 2006, vol. 2.

[11] Packing Selection Guidelines for Sliding-Stem Valves, 59.1:062, Emerson Process Management, Marshalltown, IA, 2007.

[12] J. W. Richardson, Primary Seat Shutoff, Technical Monograph 47, Fisher Controls International, Marshalltown, IA, 2005.

[13] M. Riveland, Fundamentals of Valve Sizing for Liquids, Technical Monograph 30, Fisher Controls International, Marshalltown, IA, 1985.

[14] P. Schafbuch, Fundamentals of Flow Characterization, Technical Monograph 29, Fisher Controls International, Marshalltown, IA, 1985.

[15] C. Warnett, Using Valve Actuators as Predictive Maintenance Tools for MOVs, Rotork Controls, Inc., Rochester, NY, 2000.

[16] Valve Sizing Technical Bulletin, MS-06-84-E Rev. 3, Swagelok Company, Solon, OH, 2002.

# Part III

# Control Systems

# Chapter 17

# Introduction to Control Systems

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: system, state, input, output, disturbance, controller, system (process, plant), feedback, linearity, time invariance, transfer function, open-loop control, closed-loop control, tracking, regulation
2. Define the following system problems: system dynamics, system identification, system control
3. State the differences between open-loop and closed-loop control
4. List and explain four advantages of closed-loop control
5. Determine whether or not a system is linear or time invariant
6. Convert transfer function models between Evans form and Bode form
7. Draw a block diagram of a basic feedback control system and state the functions of each block
8. Reduce complex block diagrams to determine overall transfer functions for multi-input, multi-output systems

A system is a bounded collection of components that interact to execute a process that performs a task. In control engineering, the words system, plant, and process are synonymous. The state of a system is affected by its internal stored energy and by energy crossing into and out of the system boundary. The state is defined as the current internal condition of the system. Input variables stimulate the system to change its state, and can be classified into two types: manipulated variables and disturbance variables. Output variables are variables that are observed by measurement for information or control purposes.

## 17.1 System structure

### 17.1.1 State

State refers to the energy conditions internal to the system. A state variable will be indicated in this text (and many others) by $x(t)$ in the time domain and $X(s)$ in the complex frequency domain. The concept of state will be addressed later in the text in regard to state space modeling.

### 17.1.2 Inputs

#### Control inputs

Control inputs are input signals to the system that can be manipulated. A control input variable will be indicated in this text (and many others) by $u(t)$ in the time domain and $U(s)$ in the complex frequency domain. Controller manufacturers (and some textbooks) may refer to these signals as Manipulated Variables (MV), Control Variables (CV), or Controller Outputs (CO).

#### Disturbance inputs

Disturbance inputs are input signals to a system that cannot be manipulated. Disturbance inputs will be classified into three different categories in this text: disturbances that affect the systems inputs or outputs directly (typically at low frequencies), noise that affects measurements (typically at high frequencies), and references. A disturbance variable will be indicated in this text (and many others) by $d(t)$ in the time domain and $D(s)$ in the complex frequency domain. A noise variable will be indicated by $n(t)$ in the time domain and $N(s)$ in the complex frequency domain. A reference variable will be indicated by $r(t)$ in the time domain and $R(s)$ in the complex frequency domain. Controller manufacturers (and some textbooks) may refer to reference variables as setpoints (SP).

### 17.1.3 Outputs

#### Process variables

## 17.2 System properties

This book will focus on using linear time-invariant (LTI) transfer function or state-space models (though linearization may be required to arrive at these models). Thus, a brief description of linearity and time-invariance are described below.

### 17.2.1 Linearity

Systems that are linear are both additive and homogeneous. Additivity means that if a system $G(s)$ gives output $y(t)$ when input $u(t)$ is applied, when the sum of two inputs $u_1(t)$ and $u_2(t)$ is applied, the output will be the sum of the individual responses to the inputs, i.e., $y_1(t)+y_2(t)$. Homogeneity means that if a system $G(s)$ gives output $y(t)$ when input $u(t)$ is applied, when a scaled version of the input is applied $au(t)$, the output will be a scaled version of the inputs response, i.e., $ay(t)$.

In other words, a system exhibits linear operation will produce an output that is made up of the individual scaled and summed output responses to its scaled and summed inputs. This allows the property of superposition to be used in analysis. Thus, if a linear system has multiple inputs, its output will be the sum of each of the responses to each input considered independently.

### 17.2.2 Time-invariance

Assume a system $G(s)$ gives output $y(t)$ when input $u(t)$ is applied. Now, let the input to the system be shifted in time by $\tau$ units, i.e., $u(t)$ becomes $u(t-\tau)$. A time-invariant system will produce an output that will be the time-shifted version of the original response, i.e., $y(t-\tau)$.

## 17.3 Transfer functions

A *transfer function* is the ratio of a linear time-invariant system's output signal to its input signal in the complex frequency domain when no initial conditions exist. This definition is important, as transfer functions are used extensively later in the text. For the purposes of this text, all transfer functions will be written using the ratio of two polynomials. They will also be causal, meaning that the order of the denominator polynomial will always be equal to or greater than the order of the numerator. There are two main ways of writing a transfer function: Evans form and Bode form.

### 17.3.1 Evans form

In Evans form, the coefficient of the highest power of $s$ is equal to one for every term. Given a system with $m$ zeros and $n$ poles ($m \leq n$), the Evans form of the transfer function may be written:

$$G(s) = \frac{K\prod_i^m (s+z_i)}{\sum_j^n (s+p_j)}$$

where $K$ is a constant multiplier and $z_i$ and $p_j$ are the ith zero and jth pole, respectively. As an example, consider the following transfer function written in Evans form (one in expanded form and one in factored form):

$$T(s) = \frac{1}{s^2+4s+3} T(s) = \frac{1}{(s+1)(s+3)}$$

Evans form offers the advantage that the poles are easily identified, as they are the constants of the factored terms. This makes it easy to draw a pole-zero map.

### 17.3.2 Bode form

In Bode form, the constant term is equal to one for every term. Given a system with $m$ zeros and $n$ poles ($m \leq n$), the Bode form of the transfer function may be written:

$$G(s) = \frac{K_{dc} \prod_i^m (b_i s + 1)}{\sum_j^n (a_j s + 1)}$$

where $K_{dc}$ is the dc gain of the system (the steady-state gain occurs when $s = 0$) and $b_i$ and $a_j$ are the inverse of the zeros and poles, respectively.

As an example, consider the above transfer function written in Bode form (one in expanded form and one in factored form):

$$T(s) = \frac{\frac{1}{3}}{\frac{1}{3}s^2 + \frac{4}{3}s + 1} T(s) = \frac{\frac{1}{3}}{(s+1)(\frac{1}{3}s + 1)}$$

Bode form offers the advantage that the time constants are easily identified, as they are the coefficients of the factored $s$ terms. Also, the constant in the numerator is the steady-state gain of the system if there are no independent $s$ terms in the denominator.

## 17.4 System problems

A system may be represented in a block diagram consisting of three main parts: input signals, the system, and output signals. The system is typically shown as a box on a diagram with arrows for input signals entering it and output signals leaving it as shown in Figure 17.1.

Figure 17.1: A generic system block diagram

When a mathematical representation of any two of the three main parts is known, the third may be found. This concept leads into several problems that can then be solved.

### 17.4.1 System dynamics

The system dynamics problem solves for the outputs of a system given a mathematical representation of both the inputs and the system, as shown in Figure 17.2.

Figure 17.2: The system dynamics problem

This type of problem is often introduced in a class that teaches convolution, Laplace transforms, and transfer functions of a simple linear time-invariant lumped parameter system which are represented by ordinary differential equations. Typically, the Laplace transform is taken of the differential equation $g(t)$ to find a transfer function $G(s)$, the Laplace transform of the input signal $u(t)$ is found to be $U(s)$, the two are multiplied to find $Y(s) = G(s)U(s)$, then the inverse Laplace transform is taken of $Y(s)$ to find $y(t)$.

### 17.4.2 System design

The system design problem requires the designation of a system that receives a set of input signals and produces a desired set of output signals. Filter design is one example of this problem where it is easy to visualize a noisy signal being turned into a smooth signal with the dynamic response of a properly-designed filter[1].

[1] Technically, all engineering design problems fall into this category, since the goal is to produce a system that converts inputs into the proper outputs

### 17.4.3 System identification

The system identification problem solves for a mathematical representation of a system given a mathematical representation of both the input signals and the output signals, as shown in Figure 17.3.

Figure 17.3: The system identification problem

This problem is typically not introduced in the first course a student takes in system theory, however, it is much more likely to be encountered by a control engineer in the real world than is the dynamics problem. Simply put, unless a system is really simple, it is nearly impossible to represent it as a linear time-invariant ordinary differential equation. Systems are usually nonlinear, and sometimes time-varying, and often require distributed parameters so that ordinary differential equations are not sufficient. Thus, experiments are conducted where the inputs are manipulated and the outputs are recorded. The results of these experiments are used to determine the mathematical model of the system. System identification is also referred to as process characterization.

### 17.4.4 System control

The system control problem determines the required input signals that must be put into a known system to achieve desired output signals, as shown in Figure 17.4.

Figure 17.4: The system control problem

The required input signals are developed by a controller. Since a mathematical model of the system must be known, the system identification problem must usually be solved first before tackling the system control problem.

## 17.5 Types of control

Control involves placing a controller in the system in such a way that it affects the outputs of a system.

### 17.5.1 Open-loop control

If the system model is known, the simplest form of control is to use a controller that is the inverse of the system connected in series with the system. In this configuration, since the controller has no knowledge of the output of the system, this is termed *open-loop control*. Open-loop control is indicated in Figure 17.5.

Figure 17.5: Open-loop control

For example, assume the desired output from a system $G(s)$ is a constant value of 1 at all times and the system performs the mathematical operation of multiplying the input signal by 5, i.e., $G(s) = 5$. Logically, the input signal to the system should be held at $\frac{1}{5}$ at all times to achieve the desired output. If a controller is connected in series with the system and is provided with the setpoint set at the desired output of 1, what mathematical operation should the controller perform? Well, it obviously should perform multiplication by $\frac{1}{5}$.

Open-loop control can be used, but it suffers from some problems. In order to get the desired output, the model of the system must be known perfectly, the system must be stable, and it must be invertible. Also, no noise or other disturbances can affect the system. In practice, this is impossible to achieve, and the output of a system under open-loop control will vary from the desired output. This variance may be small enough to be acceptable, but more often than not, open loop control will not perform to desired specifications.

### 17.5.2 Closed-loop control

Closed-loop control involves feeding back some or all of the output of the system to the controller and comparing it to the reference signal. See Figure 17.6.

Figure 17.6: Closed-loop control

Note: this figure shows the transfer functions as upper case letters and signals as lower case letters to differentiate between transfer functions and signals. Since no variables indicate whether or not these are in the time domain or the frequency domain, either may be assumed as long as all signals and transfer functions are in the same domain.

Feedback can be used to:

- reduce the sensitivity to uncertainties in the system model or system parameter shifts
- stabilize unstable systems
- cause the system behavior to follow a desired setpoint (tracking)
- reject disturbances and noise in a system with a constant setpoint (regulation)

## 17.6 Block diagram mathematics

Other controls textbooks attempt to teach shortcuts for reducing complex system block diagrams to simpler block diagrams or determining transfer functions from one signal to another. These methods, while meant to save students time, seem to be difficult for students to apply. Thus, this text focuses on methods of simplification that are very straightforward and have been developed from simple algebra. Since the focus of this text is on block diagrams of LTI systems, superposition applies.

The procedure to find the transfer function from one signal to another is fairly simple:

1. Choose the desired input signal and set all other inputs to zero.
2. Move from the desired input signal toward the output signal in the direction of the arrows, writing down all signs and transfer functions along the way until arriving at the output signal. This is the numerator of the transfer function.

3. Determine the product of all transfer functions in the feedback loop, then add one to it. This is the denominator of the transfer function.

**Example 17.15** Refer to Figure 17.7, and find the transfer function from $n$ to $y$.

Figure 17.7: An example system

**Answer to Example 17.15** After executing Step 1, the diagram looks like Figure 17.8

Figure 17.8: An example system solution - step 1

To execute Step 2, start at $n$ and move toward $y$. A positive sign is encountered, then a negative sign, followed by $C$, then $G$, and finally, $y$ is reached without encountering any more signs or transfer functions, so the numerator of the transfer function from $n$ to $y$ is $-CG$.

Step 3 requires the product of all transfer functions in the loop. The only transfer functions present in the feedback loop are $C$ and $G$, so the denominator of the transfer function from $n$ to $y$ is $CG + 1$. Therefore, the transfer function from $n$ to $y$ is:

$$T_{yn} = \frac{-CG}{CG + 1}$$

An alternative method starts with the signal $y$ and then works all the way around the loop in the direction of the arrows, picking up only the input signal of interest along the way. For example, beginning with $y$ and moving in the direction of the arrows, the output of the summing block near nonzero input signal $n$ is $y + n$. Since input $r$ is set to zero, the output of the summing block near it is $-(y + n)$, which enters controller transfer function $C$. The output of $C$ is then $-C(y + n)$, which enters plant transfer function $G$. The output of $G$ is then $-GC(y + n)$, which is equal to $Y$ since input $d$ is set to zero. Solving $y = -GC(y + n)$ for the ratio of $y$ to $n$ gives the transfer function:

$$T_{yn} = \frac{-CG}{CG + 1}$$

which is the same as found earlier.

Since the example has three inputs ($r$, $d$, and $n$) and one output ($y$) and the principle of superposition applies, $y = T_{yr}r + T_{yd}d + T_{yn}n$. Thus, there are three transfer functions that can

be written, and linear algebra representations can be used to create an output vector that is the product of a transfer function matrix and an input vector:

$$[y] = \begin{bmatrix} T_{yr} & T_{yd} & T_{yn} \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix}$$

This notation makes it easier to handle systems with multiple inputs and outputs. For example, to measure the error signal $e$ and the controller effort $u$ in addition to the plant output $y$, there would be three outputs. This would then be written as a 3x3 transfer function matrix with a 3x1 output vector and a 3x1 input vector:

$$\begin{bmatrix} y \\ e \\ u \end{bmatrix} = \begin{bmatrix} T_{yr} & T_{yd} & T_{yn} \\ T_{er} & T_{ed} & T_{en} \\ T_{ur} & T_{ud} & T_{un} \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix}$$

As an exercise, determine the remaining two entries in the transfer function matrix for the case with inputs ($r$, $d$, and $n$) and one output ($y$).

# References


[1] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, Design of Feedback Control Systems, 4th ed. New York, NY: Oxford University Press, 2002.

[2] C. L. Phillips and R. D. Harbor, Feedback Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[3] J. R. Rowland, Linear Control Systems: Modeling, Analysis, and Design. New York, NY: John Wiley & Sons, 1986.

[4] R. C. Dorf and R. H. Bishop, Modern Control Systems, 10th ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2005.

[5] F. J. Hale, Introduction to Control System Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[6] A. P. Sage, Linear Systems Control. Champaign, IL: Matrix Publishers, 1978.

[7] M. Driels, Linear Control Systems Engineering. New York, NY: McGraw-Hill, 1995.

[8] W. J. Palm, Control Systems Engineering. New York, NY: John Wiley and Sons, 1986.

[9] S. C. Gupta and L. Hasdorff, Fundamentals of Automatic Control. New York, NY: John Wiley and Sons, 1970.

[10] A. F. D'Souza, Design of Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[11] T. E. Marlin, Process Control, 2nd ed. New York, NY: McGraw-Hill, 2000.

[12] B. A. Ogunnaike and W. H. Ray, Process Dynamics, Modeling, and Control. New York, NY: Oxford University Press, 1994.

[13] A. Tewari, Modern Control Design with MATLAB and Simulink. New York, NY: John Wiley and Sons, 2002.

[14] C. A. Smith and A. Corripio, Principles and Practice of Automatic Process Control, 3rd ed. New York, NY: John Wiley and Sons, 2006.

[15] H. V. Vu and R. S. Esfandiari, Dynamic Systems: Modeling and Analysis. Boston, MA: McGraw-Hill, 1997.

[16] M. Gopal, Control Systems: Principles and Design. Boston, MA: McGraw-Hill, 2008.

# Chapter 18

# Process Dynamics

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: time constant, rise time, peak time, overshoot, settling time, integrating process, open-loop unstable process, inverse response (nonminimum phase) process, dead time
2. State and apply the following Laplace transforms for signals or operations: unit impulse, unit step, unit ramp, exponential, decaying sine (cosine), derivative, integral, time delay, final value theorem
3. Solve the time domain and frequency domain equations relating output to input for a first- or second-order system by applying Laplace transforms
4. Qualitatively determine the type of time-domain impulse response indicated by the locations of simple (or complex) poles and zeros in the complex plane
5. Classify a second-order system differential equation or transfer function as underdamped, critically damped, or overdamped using the system characteristic equation
6. Given an underdamped second-order system differential equation or transfer function, determine the undamped natural frequency, the damped response frequency, and the damping ratio
7. Describe the effect of dead time (process delay) on process dynamics

The purpose of this chapter is to review some basic systems theory, especially to review the dynamics of simple processes. It is not meant to be a primer on linear systems.

## 18.1 Laplace transform review

### 18.1.1 Definition

The Laplace transform of a function $f(t)$ for $t > 0$ is defined by the following integral:

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty e^{-st} f(t)dt$$

### 18.1.2 Laplace Transforms

Table 18.1 contains Laplace transforms that will be commonly used when discussing control systems and their dynamic responses.

| **Time Function** | **Laplace Transform** |
|---|---|
| $f(t) = \mathcal{L}^{-1}\{F(s)\}$ | $F(s) = \mathcal{L}\{f(t)\}$ |
| $\delta(t)$ (unit-impulse function) | $1 \quad s > 0$ |
| $u(t)$ (unit-step function) | $\frac{1}{s} \quad s > 0$ |
| $t$ (unit-ramp function) | $\frac{1}{s^2} \quad s > 0$ |
| $e^{-at}$ | $\frac{1}{s+a} \quad s > a$ |
| $\sin \omega t$ | $\frac{\omega}{s^2+\omega^2} \quad s > 0$ |
| $\cos \omega t$ | $\frac{s}{s^2+\omega^2} \quad s > 0$ |
| $e^{-at} \sin \omega t$ | $\frac{\omega}{(s+a)^2+\omega^2} \quad s > a$ |
| $e^{-at} \cos \omega t$ | $\frac{s+a}{(s+a)^2+\omega^2} \quad s > a$ |
| $f(t-a)$ | $e^{-as}F(s)$ Time-delay theorem |
| $\lim_{t\to\infty} f(t)$ | $\lim_{s\to 0} sF(s)$ Final value theorem (only applies to stable systems) |
| $f'(t)$ | $sF(s) - f(0)$ |
| $f''(t)$ | $s^2F(s) - sf(0) - f'(0)$ |
| $f^{(n)}(t)$ | $s^nF(s) - s^{n-1}f(0) - s^{n-2}f'(0) ... - f^{(n-1)}(0)$ |
| $\int_0^t f(t)dt$ | $\frac{F(s)}{s}$ |

Table 18.1: A table of commonly-used Laplace transforms

## 18.2 Stable (self-regulating) processes

### 18.2.1 Examples

A "self-regulating" process responds to a step-change in the final control element's status by settling to a new, stable value. These processes are inherently stable, and the poles of the characteristic equation are in the open left half of the complex frequency.

A good example of a self-regulating process is liquid flow control. If a control valve is opened in a step-change fashion, liquid flow through the pipe generally stabilizes at a new rate rather quickly. Figure 18.1 shows a typical liquid flow-control installation, with a process trend showing the flow response to a step-change in valve position (with the controller in manual mode).

Figure 18.1: A self-regulating flow control process response

The defining characteristic of a self-regulating process is its inherent ability to settle at a new process variable value without any corrective action on the part of the controller. In other words, there is a unique process variable value for each possible output (valve) value.

A corollary to the above statement is that a unique output value *will be required* to achieve a new process variable value. For example, to achieve a greater flow rate, the control valve must be opened further and held at that further-open position for as long as the greater flow rate is desired.

### 18.2.2 Mathematical descriptions

#### First-order systems

First-order systems have a transfer function with only one real pole. The time constant is the inverse of the pole frequency. If the real part of the pole is negative, the system will be stable. If the real part of the pole is positive, the system will be unstable. Only the stable case shall be considered. The pole-zero map, the step response, and the Bode plot of the system with transfer function of $G(s) = \frac{1}{s+1}$ are shown in Figure 18.2, Figure 18.3, and Figure 18.4, respectively.

Figure 18.2: The pole-zero map of a first-order self-regulating process

Figure 18.3: The unit step response of a first-order self-regulating process

Figure 18.4: The Bode plot of a first-order self-regulating process

### Second-order systems

Second order systems have a transfer function with two poles, therefore they have a quadratic characteristic equation. There are two general forms used for second-order systems. The first to

consider is:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\omega_n$ is the natural frequency and $\zeta$ is the damping factor. The damping factor is 0 for completely undamped systems, $0 < \zeta < 1$ for underdamped systems, 1 for critically damped, and greater than 1 for overdamped systems.

The poles may be real and distinct (overdamped), real and identical (critically-damped), or complex (underdamped). If the real parts of the poles are negative, the system will be stable. If the real part of the poles are positive, the system will be unstable. Only the stable case is considered below.

**Overdamped** The pole-zero map, the step response, and the Bode plot of the system with transfer function of $G(s) = \frac{2}{s^2+3s+2}$ are shown in Figure 18.5, Figure 18.6, and Figure 18.7, respectively.

Figure 18.5: The pole-zero plot of a second-order overdamped self-regulating process

Figure 18.6: The unit step response plot of a second-order overdamped self-regulating process

Figure 18.7: The Bode plot of a second-order overdamped self-regulating process

**Critically-damped** The pole-zero map, the step response, and the Bode plot of the system with transfer function of $G(s) = \frac{1}{s^2+2s+1}$ are shown in Figure 18.8, Figure 18.9, and Figure 18.10, respectively.

Figure 18.8: The pole-zero plot of a second-order critically-damped self-regulating process

Figure 18.9: The unit step response plot of a second-order critically-damped self-regulating process

Figure 18.10: The Bode plot of a second-order critically-damped self-regulating process

**Underdamped** The pole-zero map, the step response, and the Bode plot of the system with transfer function of $G(s) = \frac{100}{s^2+2s+100}$ are shown in Figure 18.11, Figure 18.12, and Figure 18.13, respectively.

Figure 18.11: The pole-zero plot of a second-order underdamped self-regulating process

Figure 18.12: The unit step response plot of a second-order underdamped self-regulating process

Figure 18.13: The Bode plot of a second-order underdamped self-regulating process

### Higher-order processes

Simple, self-regulating processes tend to be first-order: that is, they have only one mechanism of lag. More complicated processes often consist of multiple sub-processes, each one with its own lag time. A convection oven heating a potato can be used as an example. Both the oven temperature and the potato temperature are monitored using thermocouples and remote temperature indicators as shown in Figure 18.14.

Figure 18.14: A convection oven cooking a potato

The oven itself is a first-order process. Given enough time and sufficiently thorough air circulation, the oven's air temperature will eventually self-stabilize at the heating element's temperature. The graph of its temperature over time with the heater power fixed in "manual" mode (no thermostat to control it), shows a classic first-order function as shown in Figure 18.15.

Figure 18.15: A first-order response

The potato forms another first-order process, absorbing heat from the air within the oven (heat transfer by convection), gradually warming up until its temperature (eventually) reaches that of the

oven[1]. From the perspective of the heating element to the oven air temperature, a first-order process exists. From the perspective of the heating element to the potato, however, we have a *second*-order process.

Intuition might lead you to believe that a second-order process is just like a first-order process – except slower – but that intuition would be wrong. Cascading two first-order lags creates a fundamentally different time dynamic. In other words, two first-order lags do not simply result in a *longer* first-order lag, but rather a second-order lag with its own unique characteristics as discussed in the section on second-order process dynamics above.

By superimposing a graph of the potato temperature with a graph of the oven temperature (once again assuming constant power output from the heating element, with no thermostatic control), it can be seen that the *shape* of this second-order lag is different. The curve now has an "S" shape, rather than a consistent downward concavity, as seen in Figure 18.16.

Figure 18.16: A second-order overdamped response

This, in fact, is the tell-tale signature of multiple lags in a process: an "S"-shaped curve rather than the characteristically abrupt initial rise of a first-order curve.

By ramping the heater power at a constant rate and graph the heater element, air, and potato temperatures, the separate lag times of the oven and the potato can be clearly seen in Figure 18.17 as offsets in time at any given temperature.

[1]Given the presence of water in the potato which turns to steam at 212 $^{o}$F, things are just a bit more complicated than this, but assume a completely dry potato for now!

Figure 18.17: Lag times illustrated as offsets

As another example, consider the control of level in three cascaded, gravity-drained vessels as shown in Figure 18.18.

Figure 18.18: Gravity-drained tanks in series

From the perspective of the level transmitter on the last vessel, the control valve is driving a *third-order* process, with three distinct lags cascaded in series. This would be a challenging process to control, and not just because of the possibility of the intermediate vessels overflowing (since their levels are not being measured)!

When considering the dynamic response of a process, the primary concern is with the physical process itself. However, the instruments attached to that process also influence lag orders and lag times. As discussed previously, almost every physical function exhibits some form of lag. Even the instruments used to measure process variables have their own (usually very short) lag times. Control valves may have substantial lag times, measured in the tens of seconds for some large valves. Thus, a "slow" control valve exerting control over a first-order process effectively creates a second-order loop response. Thermowells used with temperature sensors such as thermocouples and RTDs can also introduce lag times into a loop (especially if the sensing element is not fully contacting the bottom of the well!).

This means it is nearly impossible to have a control loop with a purely first-order response. Many higher-order loops come close to being first-order, but only because the lag time of the physical process swamps (dominates) the relatively tiny lag times of the instruments. For inherently fast processes such as liquid flow and liquid pressure, however, the process response is so fast that even short time lags in valve positioners, transmitters, and other loop instruments significantly alter the loop's dynamic character.

### 18.2.3 Inverse-response processes

#### Example

One example of a system that exhibits this inverse response is boiler water level. When steam flow increases, it would be normal to presume that since more water is leaving the boiler, the level should drop. However, the pressure (saturation) inside the boiler drops, which causes larger vapor bubbles to form in the water. Since the vapor bubbles take up more volume, they displace more water, and the only direction for the water to move is up since it is constrained by the sides and bottom of the tank. Thus, the level rises (this is termed swell[2]). If the higher level of steam demand continues without and increase in flow, the water level then eventually drops.

#### Mathematical description

One or more zeros in the right side of the complex plane (i.e., $Re\{z\} > 0$) causes an inverse response to a step input. The process response starts out going in one direction, then reverses direction and ends up going in the opposite direction and finally coming to a final value (presuming the system is stable). The pole-zero map, the step response plot, and the Bode plot of the system with transfer function of $G(s) = \frac{-5(s-2)}{s^2+2s+10}$ are shown in Figure 18.19, Figure 18.20, and Figure 18.21, respectively.

[2]The opposite effect, shrink, occurs when a drop in steam demand occurs.

Figure 18.19: The pole-zero plot of an inverse response self-regulating process

Figure 18.20: The unit step response plot of an inverse response self-regulating process

Figure 18.21: The Bode plot of an inverse response self-regulating process

## 18.3 Unstable processes

Two types of unstable processes are normally encountered: integrating and runaway. An "integrating" process responds by ramping either up or down at a rate proportional to the magnitude of the final control element's step-change. Finally, a "runaway" process responds by heading toward complete instability without some form of corrective action from the controller.

### 18.3.1 Integrating processes

#### Examples

A good example of an integrating process is liquid level control, where either the flow rate of liquid into or out of a vessel is constant and the other flow rate is controlled. If a control valve is opened in a step-change fashion, liquid level in the vessel ramps at a rate proportional to the difference in flow rates in and out of the vessel. Figure 18.22 shows a typical liquid level-control installation, with a process trend showing the level response to a step-change in valve position (with the controller in manual mode).

Figure 18.22: The step response of an integrating process

It is critically important to realize that this ramping action of the process variable over time is a characteristic of the process itself, not the controller. When liquid flow rates in and out of a vessel are mis-matched, the liquid level within that vessel will change at a rate proportional to the difference in flow rates. The trend shown here reveals a fundamental characteristic of the process.

Mathematically, the integrating nature of this process can be expressed using calculus notation. First, express the *rate of change* of volume in the tank over time ($\frac{dV}{dt}$) in terms of the flow rates in and out of the vessel:

$$\frac{dV}{dt} = Q_{in} - Q_{out}$$

For example, if the flow rate of liquid going into the vessel was 450 gallons per minute, and the constant flow rate drawn out of the vessel was 380 gallons per minute, the volume of liquid contained within the vessel would increase over time at a rate equal to 70 gallons per minute: the difference between the in-flow and the out-flow rates.

The solution to the differential equation do determine the liquid volume in the vessel as a function of time is found by integration:

$$V = \int_0^T (Q_{in} - Q_{out})\, dt$$

In the given scenario of a liquid level control system where the out-going flow is held constant, this means the level will be stable only at one in-coming flow rate (where $Q_{in} = Q_{out}$). At any other controlled flow rate, the level will either be increasing over time or decreasing over time.

The example of an integrating process used here is but one of many possible processes that deal with either a *mass balance* or an *energy balance* problem. "Mass balance" is the accounting of all mass into and out of a process. Since the Law of Mass Conservation states the impossibility of mass creation or destruction, all mass into and out of a process must be accounted for. If the mass flow rate into a process does not equal the mass flow rate out of a process, the process must be either gaining or losing an internal store of mass. The same may be said for energy: all energy flowing into and out of a process must be accounted for, since the Law of Energy Conservation states the impossibility of energy creation or destruction. If the energy flow rate (input power) into a process does not equal the energy flow rate (output power) out of a process, the process must be either gaining or losing an internal store of energy.

Other examples of integrating processes include the following:

- Gas pressure control – *mass balance* – when the flow of gas either into or out of a vessel is manipulated, and the other flows in or out of the vessel are constant
- Storage bin level control – *mass balance* – when the conveyor feed rate into the bin is manipulated, and the draw from the bin is constant
- Temperature control – *energy balance* – when the flow of heat into or out of a process is manipulated, and all other heat flows are constant
- Speed control – *energy balance* – when the force (linear) or torque (angular) applied to a mass is manipulated, and all other loads are constant in force or torque

In a self-regulating process, the control element (valve) exerts control over *both* the in-flow and the out-flow of either mass or energy. In the previous subsection, where liquid flow control was the process example, the mass balance consisted of liquid flow into the valve and liquid flow out of the valve. Since the piping was essentially a "series" path for an incompressible fluid, where input flow must equal output flow at any given time, mass in and mass out were *guaranteed* to be in a state of balance, with one valve controlling both. This is why a change in valve position resulted in an almost immediate change and re-stabilization of flow rate: the valve exerts immediate control over both the incoming and the outgoing flow rates, with both in perfect balance. Therefore, nothing "integrates" over time in a liquid flow control process because there can never be an imbalance between in-flow and out-flow.

In an integrating process, the control element (valve) exerts control over *either* the in-flow *or* the out-flow of mass or energy, but never both. Thus, changing valve position in an integrating process causes an imbalance of mass flow and/or energy flow, resulting in the process variable ramping over time as either mass or energy accumulates in (or depletes from) the process.

The "simple" example of an integrating (level-control) process becomes a bit more complicated if the outgoing flow is dependent on level, as is the case with the gravity-drained vessel seen in Figure 18.23

Figure 18.23: The step response of an integrating process with a load dependent on the process variable

If the control valve is subjected to a manual step-change increase, the flow rate of liquid into the vessel immediately increases. This causes an imbalance of incoming and outgoing flow, resulting in the liquid level rising over time. As level rises, however, increasing hydrostatic pressure across the manual valve at the vessel outlet causes the outgoing flow rate to increase. This causes the mass imbalance rate to be less than it was before, resulting in a decreased integration rate (rate of level

rise). Thus, the liquid level still rises, but at a slower and slower rate as time goes on. Eventually, the liquid level will become high enough that the pressure across the manual valve forces a flow rate out of the vessel equal to the flow rate into the vessel. At this point, with matched flow rates, the liquid level stabilizes with no corrective action from the controller (remember, the step-change in output was made in manual mode!). Note the final result of letting the outgoing flow be a function of liquid level: *what used to be an integrating process has now become a self-regulating process,* albeit one with a substantial lag time.

Many processes ideally categorized as integrating actually behave in this manner. Although the manipulated variable may control the flow rate into or out of a process, the other flow rates often change with the process variable. Returning to the list of integrating process examples, it is easy to see how a process-variable load in each case can make the process self-regulate:

- Gas pressure control – *mass balance* – if uncontrolled flow rates into the vessel are allowed to naturally decrease with increasing vessel pressure, and uncontrolled flow rates out of the vessel are allowed to naturally increase with increasing vessel pressure, the vessel's pressure will tend to self-regulate instead of integrate
- Storage bin level control – *mass balance* – if the draw from the bin increases with bin level (greater weight pushing material out at a faster rate), the bin's level will tend to self-regulate instead of integrate
- Temperature control – *energy balance* – if the process loses heat at a faster rate as temperature increases, the process temperature will tend to self-regulate instead of integrate
- Speed control – *energy balance* – if drag forces on the object increase with speed (as they usually do for any fast-moving object, often proportional to speed squared), the speed will tend to self-regulate instead of integrate

This one detail completely alters the fundamental characteristic of a process from integrating to self-regulating.

### Mathematical description

An integrating process has a single pole at the origin of the complex plane. This type of process is also known as a pure capacity process. The transfer function of the simplest integrating process is:

$$G(s) = \frac{K}{s}$$

The pole-zero map, the step response plot, and the Bode plot of the system with transfer function of $G(s) = \frac{1}{s}$ are shown in Figure 18.24, Figure 18.25, and Figure 18.26, respectively.

Figure 18.24: The pole-zero plot of an integrating process

Figure 18.25: The unit step response plot of an integrating process

Figure 18.26: The Bode plot of an integrating process

Obviously, this system is not inherently stable, as the input continues to grow linearly without bound. Examples of pure capacity processes include capacitors that integrate the current flow to them and store it as charge and tanks or vessels that integrate the difference between inlet and outlet flows and store it as mass.

### 18.3.2 Runaway processes

#### Examples

A good example of an runaway process is the RBMK-1000 nuclear reactor design that was installed at Chernobyl. In that type of reactor under certain conditions, an increase in reactor power results in an increase in temperature that leads to an increase in fission neutrons and therefore a further increase in reactor power. This led to an explosion, a loss of coolant, and a meltdown of the core. This type of positive feedback results in instability. Another example of a runaway process is an electronic comparator. A very small voltage difference causes the device to run away toward one rail or the other. If the noninverting input has a larger voltage on it than the inverting input, the output voltage of the comparator will increase to just below the positive rail. If the inverting input has a larger voltage on it than the noninverting input, the output voltage of the comparator will increase to just above the negative rail.

#### Mathematical description

A runaway process has more than a single pole pair on the $j\omega$ axis of the complex plane (including at the origin), or one or more poles in the right half of the complex plane. These systems will grow without bound for any amount of stored energy or energy input.

## 18.4 Delay processes

*Lag time* refers to a damped response from a process, from a change in manipulated variable (e.g., control valve position) to a measured change in process variable: the initial effect of a change in controller output is immediately seen, but the final effect takes time to develop. *Process delay*, also known as *dead time*, by contrast, refers to a period of time during which a change in manipulated variable produces *no effect whatsoever* in the process variable: the process appears "dead" for some amount of time before showing a response. Figure 18.27 contrasts first-order and multiple-order lag times against pure dead time, as revealed in response to a manual step-change in the controller's output (an "open-loop" test of the process characteristics).

Figure 18.27: Comparison of first-order and multiple-order lag unit step responses with pure dead time response

Although the first-order response does takes some time to settle at a stable value, there is no time delay between when the output steps up and the first-order response *begins* to rise. The same may be said for the multiple-order response, albeit with a slower rate of initial rise. The dead-time response, however, is actually delayed some time after the output makes its step-change. There is a period of time where the dead-time response does *absolutely nothing* following the output step-change.

Dead time is also referred to as *transport delay*, because the mechanism of dead time is often a time delay caused by the transportation of material at finite speed across some distance. The cookie-baking process shown in Figure 18.28 has dead time by virtue of the time delay inherent to the cookies' journey from the oven to the temperature sensor.

Figure 18.28: A cookie-baking process

Dead time is a far worse problem for feedback control systems than lag time, as will be discussed in Chapter 25.

A visual comparison between the phase shifts and gains exhibited by lag versus dead time functions may be seen in Figure 18.29, the respective functions modeled by the electrical entities of a simple RC network (lag) and a transmission line (dead time).

Figure 18.29: Comparison of phase shifts and gains produced by lags and dead time

As frequency increases, the lag function's phase shift asymptotically approaches -90$^o$ while its attenuation becomes more severe. Ultimately, when the phase shift reaches its maximum of -90$^o$, the output signal amplitude is reduced to nothing. By contrast, the dead time function's phase shift grows linearly with frequency (to -180$^o$ and beyond!) while its attenuation remains unchanged.

Pure dead-time processes almost always involve the transportation of matter or energy from one

point in space to another. Usually, an industrial process will exhibit at least some degree of lag time in addition to dead time. As strange as it may sound, this is a fortunate for the purpose of feedback control, and will be discussed further in Chapter 23.

The appearance of dead time may be created in a process by the cascaded effect of multiple lags. Multiple lags create a process response to step-changes that is "S"-shaped, responding gradually at first instead of immediately following the step-change. Given enough lags acting in series, the beginning of this "S" curve may be so flat that it appears "dead," as shown in Figure 18.30.

Figure 18.30: Multiple lags can produce the appearance of dead time

While dead time may be impossible to eliminate in some processes, it should be minimized wherever possible due to its detrimental impact on feedback control. Once an open-loop (manual-mode step-change) test on a process confirms the existence of dead time, the source of dead time should be identified and eliminated if at all possible.

# References


[1] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, Design of Feedback Control Systems, 4th ed. New York, NY: Oxford University Press, 2002.

[2] C. L. Phillips and R. D. Harbor, Feedback Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[3] J. R. Rowland, Linear Control Systems: Modeling, Analysis, and Design. New York, NY: John Wiley & Sons, 1986.

[4] R. C. Dorf and R. H. Bishop, Modern Control Systems, 10th ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2005.

[5] F. J. Hale, Introduction to Control System Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[6] A. P. Sage, Linear Systems Control. Champaign, IL: Matrix Publishers, 1978.

[7] M. Driels, Linear Control Systems Engineering. New York, NY: McGraw-Hill, 1995.

[8] W. J. Palm, Control Systems Engineering. New York, NY: John Wiley and Sons, 1986.

[9] S. C. Gupta and L. Hasdorff, Fundamentals of Automatic Control. New York, NY: John Wiley and Sons, 1970.

[10] A. F. D'Souza, Design of Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[11] T. E. Marlin, Process Control, 2nd ed. New York, NY: McGraw-Hill, 2000.

[12] B. A. Ogunnaike and W. H. Ray, Process Dynamics, Modeling, and Control. New York, NY: Oxford University Press, 1994.

[13] A. Tewari, Modern Control Design with MATLAB and Simulink. New York, NY: John Wiley and Sons, 2002.

[14] C. A. Smith and A. Corripio, Principles and Practice of Automatic Process Control, 3rd ed. New York, NY: John Wiley and Sons, 2006.

[15] H. V. Vu and R. S. Esfandiari, Dynamic Systems: Modeling and Analysis. Boston, MA: McGraw-Hill, 1997.

[16] M. Gopal, Control Systems: Principles and Design. Boston, MA: McGraw-Hill, 2008.

# Chapter 19

# System Modeling

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: model, simulation, static model, dynamic model, steady state condition, deviation variable, state
2. Use MATLAB to define the following types of models and simulate time or frequency domain responses: transfer function (tf, zpk), state space (ss)
3. Derive mathematical models of the following types of simple lumped parameter systems given a problem statement, drawing, or description: electrical, mechanical (translational and rotational), electromechanical, fluid, and thermodynamic
4. Apply conservation laws to develop models
5. Use the Padé approximation to approximate dead time in a control system transfer function
6. Use deviation variables in both state space and transfer function models
7. Linearize a system of differential equations that has either a functional nonlinearity or a switching nonlinearity
8. State and apply the steps used to create a linearized mathematical model of a system

Modeling is the process of building a mathematical representation of a system that adequately mimics the behavior of the system to certain stimuli over a range of interest (a vague statement, to be true). Some mathematical models represent only behavior observed from outside a system; others also represent the internal behavior of a system.

This chapter focuses on lumped parameter modeling of systems using differential equations of components. Mathematical models that can be easily implemented in computer simulations are of interest.

## 19.1 Component analogies

The dynamics of a system are governed by how its components act upon the energy in a system. There are generally two types of components: components that store energy and components that dissipate energy. Since energy is the time integral of power in a system, models that relate two variables that, multiplied together, produce power, are often useful.

### 19.1.1 Variables

The variables that determine the power in a component are often categorized into across variables and through variables. *Across* variables develop across a component; *through* variables pass through a component[1]. For example, in an electrical component, power is the product of voltage and current. The voltage is considered to be an across variable, since it appears across resistors and energy sources (it also supplies the effort, or motive force, for the circuit). The current that results due to the applied voltage is considered to be a through variable since it moves through resistors and energy sources (it flows through them). The following sections provide a single dissipative equation for each system discussed, and two storage equations hereafter known as Category 1 and 2. Category 1 storage components will define the flow variable in terms of the rate of change of the effort variable; Category 2 storage components will define the effort variable in terms of the rate of change of the flow variable.

### 19.1.2 Electrical

The across variable is voltage ($v$) and the through variable is current ($i$), as previously discussed.

#### Dissipation

A resistor dissipates energy in an electrical system. The equation relating the effort and through variables is:

$$v = Ri$$

#### Storage

The Category 1 storage component in an electrical system is the capacitor, which stores energy in the electrical polarization of its dielectric. The equation relating the across and through variables

[1] Across varaibles also may be called *effort* variables; through variables also may be called *flow* variables since they pass through a component.

in a capacitor is:

$$i = C\frac{dv}{dt}$$

where C is the capacitance. The Category 2 storage component in an electrical system is the inductor, which stores energy in its magnetic field. The equation relating the across and through variables in an inductor is:

$$v = L\frac{di}{dt}$$

where L is the inductance.

### 19.1.3 Mechanical - Translational Motion

The across variable is velocity ($v$) and the through variable is force ($F$).

#### Dissipation

A damper dissipates energy in a mechanical system. One equation relating the across and through variables is:

$$F = Dv$$

where D is the damping factor. Note: this is a linear relationship between the across and through variables. Not all mechanical components are governed by this relationship. Some damping forces are a function of the velocity squared ($v^2$).

#### Storage

The Category 1 storage component in a mechanical system is the mass, which stores energy in its inertia. The equation relating the across and through variables in a mass is:

$$F = m\frac{dv}{dt}$$

where m is the mass. The Category 2 storage component in a mechanical system is the spring, which stores energy in its compression or tension. The equation relating the across and through variables in a spring is:

$$v = c\frac{dF}{dt}$$

where c is the compliance of the spring (the inverse of the spring constant).

### 19.1.4 Mechanical - Rotational Motion

The across variable is angular velocity ($\omega$) and the through variable is torque ($\tau$).

#### Dissipation

A rotational damper dissipates energy in a rotational mechanical system. One equation relating the across and through variables is:

$$\tau = D\omega$$

where D is the damping factor. Again, this is a linear relationship between the across and through variables. Not all rotational mechanical components are governed by this relationship. Some damping forces are a function of the angular velocity squared ($\omega^2$).

#### Storage

The Category 1 storage component in a rotational mechanical system is the mass, which stores energy in its mass moment of inertia. The equation relating the across and through variables in a mass is:

$$\tau = J\frac{d\omega}{dt}$$

where J is the moment of inertia. The Category 2 storage component in a rotational mechanical system is the rotary spring, which stores energy in its compression or tension. The equation relating the across and through variables in a spring is:

$$\omega = c\frac{d\tau}{dt}$$

where c is the compliance of the spring (the inverse of the spring constant).

### 19.1.5 Thermal

The across variable is temperature difference ($T$) and the through variable is heat flow rate ($\dot{Q}$).

#### Dissipation

Thermal resistance dissipates energy in a thermal system. One equation relating the across and through variables is:

$$\dot{Q} = kT$$

where k is the thermal conductivity.

#### Storage

The Category 1 storage component in a in a thermal system is the mass. The equation relating the across and through variables is:

$$\dot{Q} = C_t\frac{dT}{dt}$$

where $C_t$ is the thermal capacitance of the material. There is no Category 2 storage component in a thermal system.

### 19.1.6 Fluid

The across variable is pressure ($p$) and the through variable is volumetric flow ($q$).

**Dissipation**

Fluid friction dissipates energy in a fluid system. One equation relating the across and through variables is:

$$p = fq$$

where f is the friction factor. Again, this is a linear relationship between the across and through variables. This relationship only applies to laminar flow conditions. More often than not, the fluid flow will be turbulent. The pressure drop in turbulent flow conditions will be proportional to flow squared ($q^2$).

**Storage**

The Category 1 storage component in a fluid system is the fluid capacitance, which stores volume of fluid as a function of pressure. A tank is an example of a fluid capacitor. The equation relating the across and through variables in a fluid capacitor is:

$$q = C_f \frac{dp}{dt}$$

where $C_f$ is the fluid capacitance. The Category 2 storage component in a fluid system is the fluid inertance, which stores energy in the inertia of the mass of the fluid. The equation relating the across and through variables in a spring is:

$$p = I \frac{dq}{dt}$$

where I is the fluid inertance.

## 19.2 Conservation laws

Conservation laws are fundamentally balance equations that take the following form:

$$Rate\ of\ Input\ +\ Rate\ of\ Generation\ =$$

$$Rate\ of\ Output\ +\ Rate\ of\ Consumption\ +\ Rate\ of\ Accumulation$$

They generally fall into two classes based on whether or not a variable is a through variable or an across variable. For through variables, the algebraic sum of the flows entering a junction or node must be zero, and therefore may be termed nodal conservation laws. For across variables, the algebraic sum of the effort around a closed loop must be zero, and therefore may be termed loop conservation laws.

### 19.2.1 Electrical conservation laws

The nodal conservation law is Kirchoff's current law, which states that the algebraic sum of currents entering or leaving a node is zero:

$$\sum i = 0$$

The loop conservation law is Kirchoff's voltage law, which states that the algebraic sum of the voltages around a closed loop is zero:

$$\sum v = 0$$

### 19.2.2 Mechanical translational conservation laws

The nodal conservation law is d'Alembert's principle, which states that the algebraic sum of forces on an object is zero:

$$\sum F = 0$$

The loop conservation law is the velocity principle, which states that the algebraic sum of the velocities around a closed loop is zero:

$$\sum v = 0$$

### 19.2.3 Mechanical rotational conservation laws

The nodal conservation law is the rotational version of d'Alembert's principle, which states that the algebraic sum of torques on an object is zero:

$$\sum \tau = 0$$

The loop conservation law is the angular velocity principle, which states that the algebraic sum of the angular velocities around a closed loop is zero:

$$\sum \omega = 0$$

### 19.2.4 Thermal conservation laws

The nodal conservation law is the algebraic sum of heat flow rates is zero:

$$\sum \dot{Q} = 0$$

The loop conservation law is the algebraic sum of the temperature differences around a closed loop is zero:

$$\sum T = 0$$

### 19.2.5 Fluid conservation laws

The nodal conservation law is the algebraic sum of fluid flow rates entering or leaving a node is zero:

$$\sum q = 0$$

The loop conservation law is the algebraic sum of the pressure differentials around a closed loop is zero:

$$\sum p = 0$$

## 19.3 Transfer functions

Transfer functions (TFs) are defined only for linear time invariant (LTI) systems. A TF is the ratio of an output signal to an input signal in the complex frequency domain with no initial conditions. There are two general forms of transfer functions: Evans form and Bode form.

### 19.3.1 Evans form

The Evans[2] form of the transfer function is often used with root locus analysis and control design techniques because the poles and zeros are given explicitly. In the Evans form, each term in the transfer function is written as the sum of the complex variable $s$ and either a real, imaginary, or complex number $z$ or $p$ representing a zero or pole, respectively. Given a system with $m$ zeros and $n$ poles ($m \leq n$), the Evans form of the transfer function may be written:

$$G(s) = \frac{K \prod_i^m (s + z_i)}{\sum_j^n (s + p_j)}$$

where $K$ is a constant multiplier and $z_i$ and $p_j$ are the ith zero and jth pole, respectively.

### 19.3.2 Bode form

The Bode[3] form of the transfer function is often used with Bode frequency response analysis and control design techniques because the steady-state gain is given explicitly. In the Bode form, each term in the transfer function is written as the sum of a product of the complex variable $s$ with a constant (the inverse pole or zero) and 1. Given a system with $m$ zeros and $n$ poles ($m \leq n$), the Bode form of the transfer function may be written:

$$G(s) = \frac{K_{dc} \prod_i^m (b_i s + 1)}{\sum_j^n (a_j s + 1)}$$

where $K_{dc}$ is the dc gain of the system (the steady-state gain occurs when $s = 0$) and $b_i$ and $a_j$ are the inverse of the zeros and poles, respectively. This means that $a_j$ are the time constants of the system.

### 19.3.3 Transfer function model limitations

Transfer function models possess limitations. A few of these limitations are described below.

#### Dead time

Time delays, also known as dead time, result most often from transport operations. For example, if a step change in temperature occurs at the outlet of a heat exchanger in water with velocity 10 ft/s but the temperature sensor is located 20 ft downstream of the outlet, the sensor will not sense the change in temperature until 2 s after it occurred. A time delay in a function, e.g., f(x-a), which is a function f(x) that has a dead time of $a$ units, is modeled in the complex frequency domain using an exponential $e^{-as}F(s)$. However, since the exponential is not a polynomial, this representation is impossible to show on a pole-zero plot. For a transfer function to be useful for control design methods that cannot properly incorporate time delays, the delays must be approximated by pole-zero models. As a time delay is essentially an operation with no magnitude effect, only a frequency-dependent phase shift, it can be approximated by an all-pass filter. A Padé approximation replaces the exponential by a proper rational polynomial expansion that is an approximation to the exponential term, i.e., it is a

[2] Named for Walter R. Evans
[3] Named for Hendrik W. Bode

transfer function that will approximate the time delay behavior. Since the roots of the numerator and denominator of a rational polynomial can be shown on a pole-zero plot, any controls analysis or design technique (e.g., root locus) that uses pole-zero plots can use a Padé approximation. For example, the first-order Padé approximation uses the constant and the linear term from the infinite series expansion of the function, as shown below. The second-order Padé approximation uses the constant, the linear, and the quadratic terms. The *nth*-order approximation uses up through the *nth* order term. Obviously, the more terms that are included, the more accurate the approximation, but the complexity of the approximation increases. The first-order approximation follows:

$$e^{-as} \approx \frac{1 - \frac{a}{2}s}{1 + \frac{a}{2}s}$$

Higher-order Padé approximations may provide more accurate transfer functions at the expense of computation, but computer programs such as MATLAB can easily handle the task.

### Deviation variables

Systems almost invariably have initial conditions, matter or energy that is in the system at the start of a given operation. However, all initial conditions must be zero in order to form a transfer function. *Deviation variables* eliminate initial conditions and allow the development of transfer functions. Deviation variables, identified in this text by a tilde over the variable symbol, are small-signal variables defined as the difference between the value at time $t$ and the value at time 0, e.g., $\tilde{y}(t) = y(t) - y_0$.

Deviation variables are also known as *small-signal variables* because they provide the dynamics of the small portion of a signal without regard for the large part (which is subtracted out when the deviation variables are created). For example, in electronics, signals are often composed of two parts:

$$i_{AC} = I_{DC} + i_{ac}$$

where $i_{AC}$ is the total signal, $I_{DC}$ is the large-signal component, and $i_{ac}$ is the small-signal component. Many circuit models have been created to evaluate the behavior when only one of these components are present, so the technique of small-signal analysis is widely used.

To form deviation variable models from a differential equation, assume the system is at steady-state at $t = 0$ and write the differential equation again with initial variables to form the initial steady-state differential equation. Subtract the initial steady-state differential equation (which technically equals zero since the derivative of a steady-state constant is zero) from the differential equation and collect like variables to form a differential equation based on deviation variables.

**Example 19.16** Given the following differential equation with input u(t), output y(t), and initial condition $y_0$, develop a transfer function based on deviation variables.

$$\frac{dy(t)}{dt} + 4y(t) = u(t),\ y_0 = 4$$

**Answer to Example 19.16** Write the initial steady-state differential equation:

$$\frac{dy_0}{dt} + 4y_0 = u_0$$

Subtract the initial steady-state differential equation from the differential equation and collect like terms:

$$\frac{dy(t)}{dt} - \frac{dy_0}{dt} + 4y(t) - 4y_0 = u(t) - u_0 \frac{d\left[y(t) - y_0\right]}{dt} + 4\left[y(t) - y_0\right] = \left[u(t) - u_0\right]$$

Substitute deviation variables $\tilde{y}(t) = y(t) - y_0$ and $\tilde{u}(t) = u(t) - u_0$:

$$\frac{d\tilde{y}}{dt} + 4\tilde{y} = \tilde{u}$$

Taking the Laplace transform of both sides, and remembering that all initial conditions are now zero:

$$s\tilde{Y}(s) + 4\tilde{Y}(s) = \tilde{U}(s)$$

Rearrange to find $G(s)$, the ratio of the output over the input:

$$G(s) = \frac{\tilde{Y}(s)}{\tilde{U}(s)} = \frac{1}{s+4}$$

To

## 19.4 State space models

> *The state of a physical system is a set of physical quantities, the specification of which (in the absence of external excitation) completely determines the evolution of a system.* [1]

The state of a system is the minimum number of variables that is needed, given the dynamic equations of the system, to determine any future state of the system. That definition is extremely formal and does little to aid the reader in applying the concept of state. The dynamics of physical systems may be described by one or more differential equations. The specific solution to a problem defined with a differential equation requires a number of initial conditions equal to the order of the characteristic differential equation. For example, a second-order differential equation requires two initial conditions. In state space methods, an $m$th-order differential equation is broken down into $m$ first order differential equations that relate the inputs, states, and outputs. The state of a system, represented by a column vector $x$, is a minimal set of linearly independent variables equal in number to the number of initial conditions that must be specified to solve the differential equations.

The state is not unique and a given set of state variables is known as a *state representation*[4].

Though the state is not unique, state variables are typically chosen to be the variables found in the derivative portion of the component models described previously so that the states have physical meaning. For example, in electric circuits, the state variables are typically chosen to be the capacitor voltage and the inductor current, since these variables are found in the derivative portion of each component's math model[5].

[4] It should be noted that there are infinitely many possible sets of state variables that can be found using linear transformations to convert between state representations.

[5] Note also that these variables are found in the equation to calculate the stored energy in each of these components. This is no coincidence. A state space model details how the energy in a system changes with time.

The state space is the space in which the axes are defined by the state variables and the current state may be represented by a vector. If there are $p$ states, the state space will be a $p$-dimensional space. If the state vector is allowed to evolve through time, another dimension is added to the space.

### 19.4.1 State equations

#### A state space representation with no inputs or outputs

Since the state ($x$) completely determines the dynamics of the system, the rate of state change with time ($\dot{x}$) must be known. The simplest LTI state equation is:

$$\dot{x} = Ax \tag{19.1}$$

where $A$ is a constant matrix known as the *state transition matrix*. A plant model of this type is shown in Figure 19.1. The dashed line shows the boundary of the system. Equation 19.1 states that the rate of state change is a linear combination of the states, and $A$ is the mapping matrix. For a system with $p$ states, $A$ will be a square $p$x$p$ matrix.

Figure 19.1: State space plant model of system with only stored energy (initial conditions)

The state is enclosed entirely within system boundaries, therefore the state dynamics in Equation 19.1 describes only the internal behavior of a system from the standpoint of the initial system conditions. Given an initial state x(0) that describes the initial conditions, the state dynamics can be determined as the energy stored is dissipated over time. There are no inputs or outputs in the system in Equation 19.1 (refer to Figure 19.1), so there is no way to put energy into the system or tell what is happening inside the system.

#### A state space representation with inputs and outputs

A state space representation that takes into account inputs and outputs is a set of first-order differential equations that relate the current state of the system and the system's inputs and outputs

to the dynamics of the system according to the following equations:

$$\dot{x} = Ax + Bu \tag{19.2}$$

$$y = Cx + Du \tag{19.3}$$

where $x$ is the state vector, $\dot{x}$ is the rate of change of the state, $u$ is the input vector, and $y$ is the output vector. For a system with $m$ inputs, $n$ outputs, and $p$ states, the vectors will be column vectors with the respective number of rows. The matrices $A$, $B$, $C$, and $D$ are known as the state matrix, input matrix, output matrix, and feedforward matrix, respectively, and will have dimensions of $p$ x $p$, $p$ x $m$, $n$ x $p$, and $n$ x $m$, respectively.

The first equation is known as the state equation and the second equation is known as the measurement equation. Figure 19.2 shows the equations of state space representation implemented in block diagram form:

Figure 19.2: A block diagram illustrating a state space model

Since the state equation describes the dynamics of the state, knowledge of the current state, the inputs, and the matrix coefficients, all future states can be predicted by iterating on the equation and integrating. Note that the block diagram form shows the relationships between input $u$ and output $y$. This means that everything between these two variables is actually *internal* to the system.

### 19.4.2 Creating a state space model

**A state space example with no inputs or outputs**

The following shows the modeling of a system with no inputs or outputs, but that has initial conditions. The circuit is shown in Figure 19.3. Notice there are two energy storage devices, an inductor and a capacitor. This means a model of this system will be second order, so two states must be selected in state vector $x$ and state matrix $A$ will be a 2 x 2 matrix.

Inductor current and capacitor voltage are typically chosen to be the states in analysis of circuits. This is due to the following dynamic relationships for inductors and capacitors:

$$\frac{di_L}{dt} = \frac{1}{L} v_L$$
$$\frac{dv_C}{dt} = \frac{1}{C} i_C$$

which lend themselves well to representation in forms similar to Equations 19.1 and 19.3.

Figure 19.3: An RLC circuit with only initial conditions.

Inductor current and capacitor voltage are selected as the states, therefore $x = [i_L \ v_C]^T$. Using Kirchoff's Voltage and Current Laws (KVL and KCL), two equations can be written that relate the states to their rates of change:

$$\begin{aligned} v_C &= -L\frac{di_L}{dt} \\ i_L &= C\frac{dv_c}{dt} + \frac{v_C}{R} \end{aligned}$$

Rearranging the equations to solve for the state rates of change:

$$\begin{aligned} \frac{di_L}{dt} &= \frac{-v_C}{L} \\ \frac{dv_c}{dt} &= \frac{i_L}{C} + \frac{-v_C}{RC} \end{aligned}$$

Putting this into the format of Equation 19.1:

$$\frac{d}{dt}\begin{bmatrix} i_L \\ v_C \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix}\begin{bmatrix} i_L \\ v_C \end{bmatrix} \tag{19.4}$$

Thus, if a vector of initial conditions $x(0) = [i_L(0)\ v_C(0)]^T$ is given, the state trajectories can be determined through time until all the energy in the system has dissipated in the resistor. MATLAB can be used for simulating this (and all other simulations in this handout). Figure 19.4 shows a state transient response due to inital conditions for the example RLC circuit in Figure 19.3. This is known as the *zero-input response* because there is no input to the system.

Figure 19.4: The response of an RLC circuit based on initial conditions only.

### A state space example with input and output

The previous example is not a very interesting problem, however. It allows for no input and no output, and since all the dynamics happen inside the plant, it is never really possible to know what is happening. A plant that has inputs and outputs (or measurements) as shown in Figure 19.5 is more interesting.

Figure 19.5: An RLC circuit with input $v_i$ and output $v_o$.

The model will use the same states as before, and assign our output vector $y = v_o = v_c$. If KVL and KCL are once again applied, the following plant model can be developed:

$$\frac{d}{dt}\begin{bmatrix} i_L \\ v_C \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix}\begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_i$$

$$v_o = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} i_L \\ v_C \end{bmatrix} + 0v_i$$

When it is assumed that no initial conditions are present and the system response is simulated, the plots shown in Figures 19.6 and 19.7 are obtained. This is known as the *zero-state response* because the inital state conditions are zero. Note that the plot in Figure 19.7 is the same as the capacitor voltage subplot in Figure 19.6. This is because the output voltage was defined to be the voltage across the capacitor (see Figure 19.5).

Figure 19.6: The RLC circuit state response to a unit step input in voltage $v_i$ (no i.c.).

Figure 19.7: The RLC circuit output voltage $v_o$ response to a unit step input in voltage $v_i$. (no i.c.)

The system behavior can also be simulated for a combination of unit step in input voltage and transient due to initial conditions. This is known as the *total response*, and is the sum of the zero-input response and the zero-state response. The total response plots are shown in Figures 19.8 and 19.9. Note that the transient shape is similar to the zero-input response, but the transient ends at the steady-state values of the zero-state response.

Figure 19.8: The RLC circuit state response to a unit step input in voltage $v_i$ and i.c.

Figure 19.9: The RLC circuit output voltage $v_o$ response to a unit step input in voltage $v_i$ and i.c.

### 19.4.3 Transformations to/from state space representations

#### Linear transformations between state representations

Since linear matrix representation is used, a state space model is not unique, as any linear transformation can be used to transform the system into a different linear matrix representation. If given a state vector $x$, that vector may be equal to a different vector $w$ multiplied by a nonsingular matrix $T$: $x = Tw$. Therefore, $\dot{x} = T\dot{w}$ follows. Substituting these new vectors into the state space

representation in 19.3 gives:

$$\begin{aligned} T\dot{w} &= ATw + Bu \\ y &= CTw + Du \end{aligned}$$

Premultiplying both sides of the state equation by the inverse of T produces:

$$\begin{aligned} \dot{w} &= T^{-1}ATw + T^{-1}Bu \\ y &= CTw + Du \end{aligned}$$

Note that this is equivalent to:

$$\begin{aligned} \dot{w} &= \acute{A}w + \acute{B}u \\ y &= \acute{C}w + \acute{D}u \end{aligned}$$

where $\acute{A} = T^{-1}AT$, $\acute{B} = T^{-1}B$, $\acute{C} = CT$, and $\acute{D} = D$ with respect to the original state space model. This means that any of an infinite number of nonsingular transformation matrices $T$ may be used to produce another state space model. The internal state will be different (and may have no real physical meaning), but the same input-to-output relationships will be maintained.

### From state space to transfer function representation

To transform the representation from state space into a transfer function (or matrix of transfer functions for systems that are not SISO), simply use the Laplace transform operator on the state equation an solve for the relationship between $y$ and $u$. Remember, however, that matrix multiplication order matters (i.e., it is not commutative).

$$\begin{aligned} sX &= AX + BU \\ (sI - A)X &= BU \\ X &= (sI - A)^{-1}BU \\ Y &= C(sI - A)^{-1}BU + DU \\ Y &= [C(sI - A)^{-1}B + D]U \end{aligned}$$

The matrix $[C(sI - A)^{-1}B + D]$ is the transfer function matrix[6]. Even though a system can have infinitely many state space representations, there is only one transfer function from each input to each output since the input-output relationship is unique.

### From transfer function to state space representation

Given a single-input, single-output (SISO) transfer function, it is easy to convert it to a state space representation known as *phase-variable form*[7]. If an example transfer function is defined by:

$$H = \frac{b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3}$$

[6]Normally, a transfer function is formed by taking the ratio of the output and input signal representations in the complex frequency domain, but the product $YU^{-1}$ is not formed - this is because $Y$ and $U$ are vectors, which don't have inverses. However, the transfer function matrix $[C(sI - A)^{-1}B + D]$ does exist, and will have a number of rows equal to the number of outputs, and a number of columns equal to the number or inputs.

[7]For systems that have greater dimensions, refer to other texts, e.g.,

then the phase-variable state space representation $A$ matrix will be given by:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}$$

The $B$ matrix will be given by:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The $C$ matrix will be given by:

$$\begin{bmatrix} b_3 & b_2 & b_1 \end{bmatrix}$$

and the $D$ matrix will be given by:

$$\begin{bmatrix} 0 \end{bmatrix}$$

Thus, creating a phase-variable state space representation from a SISO transfer function consists of creating matrices of the appropriate dimensions and populating them with coefficients from the transfer function. The $A$ matrix is square with dimensions given by the highest coefficient of $s$. The lowest row of the $A$ matrix consists of writing (from left-to-right) the negative of the coefficients of the transfer function denominator as read from right-to-left, omitting the coefficient of the highest power of $s$. The remainder of the first column is filled in by zeros, then an identity matrix is used to fill in the remainder of the spaces beginning in the first row, second column. The $B$ vector has a one in the last row, but all other entries are zeros. The $C$ vector consists of writing (from left-to-right) the coefficients of the transfer function numerator as read from right-to-left. For any strictly proper transfer function (one with the order of the numerator higher than the order of the denominator), the $D$ vector will always be a 1x1 matrix that is zero. Note that the phase-variable form of a state space representation does not, in general, have states that are physically meaningful.

## 19.5 Linearization

Since transfer functions only exist for LTI systems, nonlinear systems do not have transfer functions. However, nonlinear systems usually act fairly linear in the vicinity of a given operating point. Three methods of linearizing nonlinear systems are introduced to tackle nonlinear continuously differentiable functions, nonlinear differential equations that contain continuously differentiable functions, and nonlinear differential equations that contain periodic switching functions.

### 19.5.1 Linearizing differentiable functions

A first-order linear approximation of a nonlinear differentiable function can be formed around operating point $\bar{x}$ by a Taylor series approximation truncated after the first-order term. That is, $f(x) \approx f(\bar{x}) + \frac{\partial f(x)}{\partial t}|_{x=\bar{x}}(x - \bar{x})$.

This can be extended to differentiable functions of any number of variables by taking the operating point value of the function plus partial derivative of the function with respect to each individual variable independently multiplied by its respective deviation from its value at the operating point.

### 19.5.2 Linearizing differentiable nonlinear differential equations

Differential equations can be linearized by using deviation variables around an operating point and the Jacobian sensitivity matrix formed from partial differentiation with respect to the states and outputs. This method works for functions of any number of variables. Given a first-order nonlinear differential equation $\frac{dy}{dt} = g\left[x(t), u(t)\right]$ with initial conditions $g(\bar{x}, \bar{u})$, the linearized multivariable differential equation is:

$$\frac{dx}{dt} \approx \frac{\partial g}{\partial x} \mid_{\bar{x},\bar{u}} (x - \bar{x}) + \frac{\partial g}{\partial u} \mid_{\bar{x},\bar{u}} (u - \bar{u})$$

### 19.5.3 Developing transfer functions for systems with differentiable nonlinearities

One procedure for developing a linearized transfer function model in deviation variables is:

- Linearize the model and determine the resulting dynamic equation.
- Find the steady-state equation from the dynamic equation (i.e., set the derivatives to zero).
- Subtract the steady-state equation from the dynamic equation
- Combine like terms to express the resulting model in terms of deviation variables.
- Take the Laplace transform and find the TF.

The following two examples illustrate the linearization of a system with a nonlinear function that is differentiable.

## Example 19.17

The temperature of a slab being heated by an electric heater is given by the differential equation

$$C\frac{dT(t)}{dt} = q(t) - \alpha\left[T^4(t) - T_s^4\right]$$

where $T(t)$ is the temperature of the slab in °R, assumed uniform, $q(t)$ is the rate of heat input in BTU/h, C = 180 BTU/°R, is the heat capacity of the slab, $T_s$ = 540 °R, is the surrounding temperature (constant), and $\alpha$ = 5x10$^{-8}$ BTU/h-°$R^4$, is the coefficient of heat radiation. Obtain the linear approximation of the differential equation around the initial steady-state temperature of 700 °R. Obtain also the Laplace transform of the temperature of the slab and find the gain and time constant of the transfer function.

**Answer to Example 19.17** The starting dynamic equation is:

$$\begin{aligned} C\frac{dT(t)}{dt} &= q(t) - \alpha\left[T^4(t) - T_s^4\right] \\ &= q(t) - \alpha T^4(t) + \alpha T_s^4 \end{aligned}$$

The initial steady-state equation is:

$$\begin{aligned} C\frac{dT_0}{dt} &= q_0 - \alpha\left[T_0^4 - T_s^4\right] \\ &= q_0 - \alpha T_0^4 + \alpha T_s^4 \end{aligned}$$

Subtracting the initial steady-state equation from the dynamic equation to form the deviation variables (and dropping the time notation for convenience):

$$C\frac{d(T-T_0)}{dt} = (q-q_0) - \alpha\left[T^4 - T_0^4\right]$$

Note that the surrounding temperature term drops out of the equation. However, there is still the nonlinear $T^4$ term ($T_0^4$ is constant).

To linearize this term around $T_0$:

$$\begin{aligned} T^4 &\approx T_0^4 + \left.\frac{dT_4}{dT}\right|_{T_0}(T-T_0) \\ &\approx T_0^4 + 4T_0^3(T-T_0) \end{aligned}$$

Substitute this in for $T^4$:

$$\begin{aligned} C\frac{d(T-T_0)}{dt} &= (q-q_0) - \alpha\left[T_0^4 + 4T_0^3(T-T_0) - T_0^4\right] \\ &= (q-q_0) - 4\alpha T_0^3(T-T_0) \end{aligned}$$

Substitute $\tilde{q}$ for $q-q_0$ and $\tilde{T}$ for $(T-T_0)$, and rearrange:

$$\begin{aligned} \tilde{q} &= C\frac{d\tilde{T}}{dt} + 4\alpha T_0^3\tilde{T} \\ \frac{1}{4\alpha T_0^3}\tilde{q} &= \frac{C}{4\alpha T_0^3}\frac{d\tilde{T}}{dt} + \tilde{T} \end{aligned}$$

Taking the Laplace transform (and omitting the $s$ dependence for convenience):

$$\frac{1}{4\alpha T_0^3}\tilde{Q} = \frac{C}{4\alpha T_0^3}s\tilde{T} + \tilde{T}$$

Solving for the transfer function:

$$\frac{\tilde{T}}{\tilde{Q}} = \frac{\frac{1}{4\alpha T_0^3}}{\frac{C}{4\alpha T_0^3}s + 1}$$

Since the transfer function is written in Bode form (all complex terms written as $\tau s+1$), the gain is the numerator term, as this is the result as $s \to 0$, and the coefficient of $s$ is the time constant. Substituting the numerical values gives the gain as 0.0146 h-°$R^4$/BTU and the time constant as 2.62 h.

## Example 19.18

Solve the previous example by linearizing the differential equation directly.

**Answer to Example 19.18**

Proceed as above to find the equation in terms of deviation variables.

$$C\frac{d(T-T_0)}{dt} = (q-q_0) - \alpha\left[T^4 - T_0^4\right]$$

Next, make use of the Taylor series expansion format given in the text to write:

$$\begin{aligned}\frac{d(T-T_0)}{dt} &= \frac{\partial}{\partial q}\left[\frac{1}{C}(q-q_0) - \frac{\alpha}{C}\left(T^4 - T_0^4\right)\right]\Bigg|_{q_0,T_0}(q-q_0) \\ &+ \frac{\partial}{\partial T}\left[(q-q_0) - \alpha\left(T^4 - T_0^4\right)\right]\Big|_{q_0,T_0}(T-T_0)\end{aligned}$$

Solving the partial differential equations that arise gives:

$$\frac{d(T-T_0)}{dt} = \frac{1}{C}(q-q_0) + \frac{-4\alpha T^3}{C}\Bigg|_{q_0,T_0}(T-T_0)$$

Rearranging and solving for the transfer function gives the same answer as when using the method of linearizing functions of one variable method.

### 19.5.4 Linearizing periodic switching differential equations

If a system has switches in it, the system is no longer linear. Switching functions are a nondifferentiable discontinuity since the switch is effectively producing a step change in the dynamic model of the system. However, many numerical simulation software packages require simulation of linear systems, and linear system simulations are certainly more tractable to solve by computer. Given a system with a $n$ switches in it, there will be two dynamic models defined by each position of the switch, for a possible set of $2n$ models. This section will discuss a method of linearizing this behavior around an operating point to develop a small-signal linear model of a system's behavior. A linearized system can be created for a periodic switching system by using the state space averaging (SSA)technique [2]. State space averaging is a well-known method used in modeling switching power converters. For a system with only two switched states with a nominal duty cycle, a model may be developed by determining the state and measurement equations for each of the two switch states, then calculating a weighted average of the two sets of equations using the nominal values of the time spent in each state as the weights. For example, given a duty cycle $D_s$, if a state space model $\{A_1, B_1, C_1, D_1\}$ can be developed with a switch in the on position and a second state space model $\{A_2, B_2, C_2, D_2\}$ can be developed with a switch in the off position. Then, the weighted average can be determined based on the duty cycle $D_s$ to find a linearized state space model $\{A, B, C, D\}$ where $A = D_sA_1+(1-D_s)A_2$, $B = D_sB_1+(1-D_s)B_2$, $C = D_sC_1+(1-D_s)C_2$, and $D = D_sD_1+(1-D_s)D_2$. Note that this is a small-signal linear model around the operating point established by the nominal duty cycle $D_s$.

The steps to follow when creating a state space average model for a system with one periodic switch are:

1. Define a state vector $x$
2. Define an output vector $y$ that contains all values to be measured
3. Define the duty cycle of the switch $D_s$
4. Draw the system configuration that results when the switch is closed
5. Write a differential equation for each energy storage element for this configuration
6. Determine the state space matrices $\{A_1, B_1, C_1, D_1\}$
7. Draw the circuit configuration that results when the switch is open
8. Write a differential equation for each energy storage element for this configuration
9. Determine the state space matrices $\{A_2, B_2, C_2, D_2\}$
10. Determine the weighted average of the two state space models based on their duty cycles
11. Find the steady-state value of the state vector $X = -A^{-1}BU$ where $U$ is the circuit input source
12. Find the steady-state value of the output vector $Y = CX + DU$
13. Find the coefficients of the duty cycle input $B_d = (A_1 - A_2)X + (B_1 - B_2)U$ and $D_d = (C_1 - C_2)X + (D_1 - D_2)U$ for the state and measurement equations, respectively
14. Write the state space averaged model $\dot{\tilde{x}} = A\tilde{x} + B\tilde{u} + B_d\tilde{d}$ and $\tilde{y} = C\tilde{x} + D\tilde{u}$ [where the tilde ( ˜ ) indicates a small signal deviation from nominal]

Then by adding the small signal model parameters to their DC values, the large signal model values may be obtained:

$$\begin{aligned} x &= X + \tilde{x} \\ u &= U + \tilde{u} \\ d &= D_s + \tilde{d} \\ y &= Y + \tilde{y} \end{aligned}$$

## Example 19.19

Create a state space averaged model of a buck converter operating in continuous conduction mode. The circuit diagram is shown in Figure 19.10.

Figure 19.10: The buck converter

Figure 19.11 shows the circuit configuration when switch $Q_1$ conducts.

Figure 19.11: The buck converter with controlled switch closed

For the circuit when $Q_1$ conducts, the differential equations are:

$$\begin{aligned} \frac{di_L}{dt} &= \frac{-1}{L}v_c + \frac{1}{L}v_g \\ \frac{dv_C}{dt} &= \frac{1}{C}i_L - \frac{1}{RC}v_c \\ v_o &= v_c \end{aligned}$$

Figure 19.12 shows the circuit configuration when switch $Q_1$ does not conduct.

Figure 19.12: The buck converter with controlled switch open

When $Q_1$ does not conduct, the differential equations are:

$$\begin{aligned} \frac{di_L}{dt} &= \frac{-1}{L}v_c \\ \frac{dv_C}{dt} &= \frac{1}{C}i_L - \frac{1}{RC}v_c \\ v_o &= v_c \end{aligned}$$

**Answer to Example 19.19** To develop the state space averaged model, the equations for the rate of inductor current change are used along with the equations for the rate of capacitor voltage change. Given a buck converter, the states correspond to inductor current ($i_L$) and capacitor voltage ($v_C$). For the purposes of developing a SSA model, the state vector will be defined as:

$$x = \begin{bmatrix} i_L \\ v_C \end{bmatrix}$$

The measurement equation needs to include all parameters of interest. For the buck converter example, the outputs selected may be inductor current $i_L$ and capacitor voltage $v_C$. Since the

output voltage $v_o$ equals the capacitor voltage $v_C$, let the output vector be defined as:

$$y = \begin{bmatrix} i_L \\ v_o \end{bmatrix}$$

Thus, when $Q_1$ conducts, the following state space matrices result based on the differential equation previously determined:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \\ B_1 &= \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \\ C_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ D_1 &= [0] \end{aligned}$$

When $Q_1$ is not conducting, the circuit model is represented by the following state space matrices:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \\ B_1 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ C_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ D_1 &= [0] \end{aligned}$$

Let the duty cycle of the switch $D_s$ and the input voltage $V_g$ be inputs. The following results may be obtained for a state space averaged model of the buck converter:

$$\begin{aligned} \frac{V_o}{V_g} &= D_s \\ D_s' &\triangleq 1 - D_s \end{aligned}$$

$$\begin{aligned} A &= D_s A_1 + D_s' A_2 \\ B &= D_s B_1 + D_s' B_2 \\ C &= D_s C_1 + D_s' C_2 \\ D &= D_s D_1 + D_s' D_2 \\ X &= -A^{-1} B V_g \\ B_d &= (A_1 - A_2)\, X + (B_1 - B_2)\, V_g \\ D_d &= (C_1 - C_2)\, X + (D_1 - D_2)\, V_g \end{aligned}$$

$$\begin{aligned} \dot{\tilde{x}} &= A\tilde{x} + B\tilde{v}_g + B_d \tilde{d}_s \\ \tilde{v_o} &= C\tilde{x} + D\tilde{v}_g + D_d \tilde{d}_s \end{aligned}$$

where the tilde ( ˜ ) indicates a small signal deviation from nominal. This model is represented as shown in Figure 19.13.

Figure 19.13: The state space average model

# References


[1] B. Friedland, Control System Design: An Introduction to State-Space Methods. New York, NY: McGraw-Hill, 1986.

[2] R. W. Erickson, Fundamentals of Power Electronics, 2nd ed. Secaucus, NJ: Kluwer Academic, 2000.

[3] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, Design of Feedback Control Systems, 4th ed. New York, NY: Oxford University Press, 2002.

[4] C. L. Phillips and R. D. Harbor, Feedback Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[5] J. R. Rowland, Linear Control Systems: Modeling, Analysis, and Design. New York, NY: John Wiley and Sons, 1986.

[6] R. C. Dorf and R. H. Bishop, Modern Control Systems, 10th ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2005.

[7] F. J. Hale, Introduction to Control System Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[8] A. P. Sage, Linear Systems Control. Champaign, IL: Matrix Publishers, 1978.

[9] M. Driels, Linear Control Systems Engineering. New York, NY: McGraw-Hill, 1995.

[10] W. J. Palm, Control Systems Engineering. New York, NY: John Wiley and Sons, 1986.

[11] S. C. Gupta and L. Hasdorff, Fundamentals of Automatic Control. New York, NY: John Wiley and Sons, 1970.

[12] A. F. D'Souza, Design of Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[13] T. E. Marlin, Process Control, 2nd ed. New York, NY: McGraw-Hill, 2000.

[14] B. A. Ogunnaike and W. H. Ray, Process Dynamics, Modeling, and Control. New York, NY: Oxford University Press, 1994.

[15] A. Tewari, Modern Control Design with MATLAB and Simulink. New York, NY: John Wiley and Sons, 2002.

[16] C. A. Smith and A. Corripio, Principles and Practice of Automatic Process Control, 3rd ed. New York, NY: John Wiley and Sons, 2006.

[17] H. V. Vu and R. S. Esfandiari, Dynamic Systems: Modeling and Analysis. Boston, MA: McGraw-Hill, 1997.

[18] M. Gopal, Control Systems: Principles and Design. Boston, MA: McGraw-Hill, 2008.

# Chapter 20

# System Identification

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: system identification, black box model, grey box model, white box model
2. Use system time domain responses of simple forcing functions to identify model parameters from data
3. List and discuss the steps typically performed in system identification
4. Draw block diagrams of the following types of system models and identify their suitability for use: AR, FIR, ARX, ARMA, ARMAX, BJ, OE

## 20.1 Introduction to system identification

System identification, also known as characterization, is the process of developing a mathematical model of a system from input/output data. The main steps involved in system identification are placing an automated system in manual, obtaining input/output data from experimentation, preprocessing the data to remove means and trends (and possibly rescaling the data, such as normalizing), selecting a model structure, estimating the model parameters, and validating the model. This process is often recursive as the initial model may need to be refined.

## 20.2 Obtaining data

Experiments to obtain input/output data must provide signals that are rich, meaning that the input signal contains enough energy over a wide enough range of frequencies that the output response shows the characteristics of the system. This type of input signal is also known as *persistently exciting*, because it excites the modes of the system enough that the system parameters can be identified. A system under closed-loop control is usually placed into manual operation to prevent the autoregressive properties of a closed-loop system from interfering with the identification process. Although closed-loop system identification techniques are available, the remainder of this chapter will focus on open-loop identification techniques.

### 20.2.1 Step responses

A step change is often used as part of the system identification process, and simple system transfer functions can be identified from the response. One of the most effective types of step inputs is a pulse input where the input signal is stepped up from its nominal operating condition and then stepped back down to its normal value. The magnitude of the change must be significant enough to cause the process to respond, and the duration of the change must be long enough to be able to see the system respond (the best case is to allow the step to occur long enough so that the system output reaches steady state). This type of input is often selected for systems with slower responses, such as industrial control processes.

The DC or steady-state gain can be found by dividing the steady-state value of the change in output by the magnitude of the step change in the input:

$$K_{dc} = \frac{\Delta y}{\Delta u}$$

#### First-order systems

A first-order system has a non-zero slope when the system response begins.

**Time constant** The time constant $\tau$ can be determined by the time it takes the response to reach 63% of its steady-state value ($1\tau$). Alternatively, it can be determined once the output value reaches 98% of the steady-state value ($4\tau$) or 100% of the steady-state value ($5\tau$).

**Second-order systems**

A second-order system has a slope of zero when the system response begins[1].

**Underdamped** The percent overshoot is found from the magnitudes of the peak and steady-state values:

$$\%OS = 100\frac{y_{pk} - y_{ss}}{y_{ss}}$$

where $y_{pk}$ is the peak magnitude of the output and $y_{ss}$ is the steady-state magnitude of the output.

The damping ratio can be calculated from the percent overshoot:

$$\zeta = \frac{-ln(\%OS/100)}{\sqrt{[\pi^2 + ln(\%OS/100)^2]}}$$

The damped frequency can be found using the period ($T_d$) of the oscillations in the response:

$$\omega_d = \frac{2\pi}{T_d}$$

The natural frequency can be found using the damping ratio and the damped frequency:

$$\omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}}$$

**Critically-damped and overdamped** Step response data typically does not provide enough information to determine parameters of a critically-damped or overdamped system other than the dc gain. The presence of an inflection point near the start of the transient is the only indication that the system is not first order, and even that may be obscured by signal noise.

### 20.2.2 Frequency responses

If the input of a system can be swept sinusoidally (called a *chirp*), the frequency response magnitude and phase can be obtained. Pole and zero frequencies can be identified from the break points shown on the magnitude graphs, and multiple poles or zeros occurring at a single frequency can be identified from slopes in the magnitude response that follow these break points, e.g., the slope of the response following a double pole will be -40 dB/decade. This type of input is very suitable for use in electronic circuits since it can be produced by a function generator. An example of a chirp signal is shown in Figure 20.1.

[1] All systems higher than first-order will have a slope of zero at the beginning of the step response transient.

Figure 20.1: An example of a chirp signal

### 20.2.3 More complex inputs

**Pseudo-random binary sequences**

A pseudo-random binary sequence (PRBS) is a sequence that appears to randomly change at each clock step between zero and one. It is called "pseudo-random" because the sequence actually repeats after a certain number of clock steps. Because this sequence consistes of a seemingly random series of step changes in the up and down direction, it contains both high and low frequency components, making it a good candidate for use as an input in system identification. For example, a PRBS may be added to the steady-state value of a control output to perturb the plant input. Since this is a type of step response, it also works well for industrial processes. An example of a pseudo-random binary sequence is shown in Figure 20.2.

Figure 20.2: An example of a pseudo-random binary sequence (PRBS)

#### Noise

Because noise can be generated with almost any frequency content, it is useful as an input signal. The frequency content of the noise can be tailored to the needs of the experimenter by passing it through a linear filter. Thus, the frequency content of the noise signal can be matched to excite the plant over a certain frequency range.

## 20.3 Data preprocessing

Once the data has been obtained using a complex input, it is usually preprocessed. One of the first steps after obtaining the data is to remove any means and trends. The system identification process is usually used to determine the small-signal transfer function, i.e., one in deviation variables, so it is necessary to remove the means from the input and output signals. If the data is particularly corrupted with high frequency noise, low-pass filtering may be used to eliminate the worst of the noise. If the data has been affected by low frequency disturbances that are not going to be modeled, high-pass filtering may be used to eliminate the worst of the disturbances. A processed data set is then often split into two subsets: a set used to perform the identification and a set used to validate the model that results from the identification.

## 20.4 Selecting a model

### 20.4.1 Types of models

#### Linear Parametric models

Parametric models have parameters that need to be determined. Transfer function coefficients and state space matrix entries are typical parameters for linear system models. A parameter search is typically used to determine the proper coefficients using a type of regression function. Parametric models are often given in terms of the variable $q$ which is a discrete-time forward shift operator that corresponds to the data sampling period[2]. In equation terms:

$$qu(k) = u(k + T)$$

where $k$ is the current time step and $T$ is the sampling period. Thus:

$$q^{-1}u(k) = u(k - T)$$

In order to account for both deterministic and stochastic components in an output signal $y(t)$, parametric models are often developed presuming a deterministic input or inputs $u(t)$ and a stochastic white noise input $e(t)$. The following paragraphs describe various types of parameterized linear system models.

[2] In a discrete-time system, the shift operator corresponds to the variable $s$ in continuous-time systems. The corresponding system in the time-domain is a difference equation instead of a differential equation.

**Autoregressive** The autoregressive (AR) model has only a noise input $e(t)$. It is given by the formula:

$$A(q)y(t) = e(t)$$

where $A(q)$ is the autoregressive equation that gives the system its dynamics. The block diagram representation is shown in Figure 20.3.

Figure 20.3: The block diagram representation of an Autoregressive (AR) system model

Note that the noise is presumed to go through the system's characteristic equation before being added to the output. As this model does not include an input, it is most suitable for representing a signal, since systems usually have at least one input. The result of the identification process is a discrete-time transfer function for the stochastic dynamics with numerator 1 and denominator $A(q)$.

**Finite Impulse Response** The finite impulse response (FIR) model has a noise input $e(t)$ and an exogenous (i.e., external) input $u(t)$. It is given by the formula:

$$y(t) = \sum_{i=1}^{m} B_i(q)u_i(t) + e(t)$$

where $B(q)$ is the finite impulse response, i.e., moving-average (MA), equation that gives the system its dynamics. The block diagram representation is shown in Figure 20.4.

Figure 20.4: The block diagram representation of an Finite Impulse Response (FIR) system model

Note that the stochastic process noise is added directly to the output. This model is only valid for systems that act as moving average filters. Since most systems have autoregression, it is limited in its application. The result of the identification process is a discrete-time transfer function for the deterministic dynamics with numerator $B(q)$ and denominator 1.

**Autoregressive with Exogenous Input** The autoregressive with exogenous input (ARX) model has a noise input $e(t)$ and an exogenous input $u(t)$, which may be made up of one or multiple input signals. It is given by the formula:

$$A(q)y(t) = \sum_{i=1}^{m} B_i(q)u_i(t) + e(t)$$

where $A(q)$ is the autoregressive equation and $B(q)$ is a moving average equation. A block diagram representation is shown in Figure 20.5.

Figure 20.5: A block diagram representation of an Autoregressive with Exogenous Input (ARX) system model

The application of this model is limited because it presumes that the stochastic input signal passes through the same autoregression as the deterministic input signal and the model has no moving-average dynamics. This is not typically the case in most systems. The result of the identification process is a discrete-time transfer function for the deterministic dynamics with numerator $B(q)$ and denominator $A(q)$ and a discrete-time transfer function for the stochastic dynamics with numerator 1 and denominator $A(q)$. This model is typically chosen when trying to estimate a simple model for a system with good signal-to-noise ratio in the output.

**Autoregressive Moving-Average** The autoregressive moving-average (ARMA) model has only a noise input $e(t)$. It is given by the formula:

$$A(q)y(t) = C(q)e(t)$$

where $A(q)$ is the autoregressive equation and $C(q)$ is a moving average equation that the noise passes through. The block diagram representation is shown in Figure 20.6.

Figure 20.6: The block diagram representation of an Autoregressive Moving-Average (ARMA) system model

This model gives addition degrees of freedom for the noise signal before it is added to the output. Similar to the AR model, this model does not include an input, so it is most suitable for representing a signal, not a system. The result of the identification process is a discrete-time transfer function for the stochastic dynamics with numerator $C(q)$ and denominator $A(q)$.

**Autoregressive Moving-Average with Exogenous Input** The autoregressive moving-average with exogenous input (ARMAX) model has a noise input $e(t)$ and an exogenous input $u(t)$, which may be made up of one or multiple input signals. It is given by the formula:

$$A(q)y(t) = \sum_{i=1}^{m} B_i(q)u_i(t) + C(q)e(t)$$

where $A(q)$ is the autoregressive equation and $B(q)$ is a moving average equation for the input signals and $C(q)$ is a moving average equation for the noise signal. The block diagram representation is shown in Figure 20.7.

Figure 20.7: The block diagram representation of an Autoregressive Moving-Average with Exogenous Input (ARMAX) system model

This model is useful when the stochastic process is presumed to enter upstream of the system's autoregression, such as a disturbance on the system's input. Identification of the model parameters results in transfer functions for both the deterministic and stochastic inputs that have the same characteristic equation. The result of the identification process is a discrete-time transfer function for the deterministic dynamics with numerator $B(q)$ and denominator $A(q)$ and a discrete-time transfer function for the stochastic dynamics with numerator $C(q)$ and denominator $A(q)$. This model is typically chosen when the dominant noise or disturbances enter at the input of the plant.

**Box-Jenkins** The Box-Jenkins (BJ) model has a noise input $e(t)$ and an exogenous input $u(t)$, which may be made up of one or multiple input signals. It is given by the formula:

$$y(t) = \sum_{i=1}^{m} \frac{B_i(q)}{A_i(q)} u_i(t) + \frac{C(q)}{D(q)} e(t)$$

where $A(q)$ is the autoregressive equation for the input and $D(q)$ is the autoregressive equation for the noise. $B(q)$ is a moving average equation for the input signals and $C(q)$ is a moving average equation for the noise signal. The block diagram representation is shown in Figure 20.8.

Figure 20.8: The block diagram representation of a Box-Jenkins (BJ) system model

Since the dynamics of the stochastic process are completely decoupled from the dynamics of the deterministic process, this model is valuable when noise is presumed to enter downstream of the system's output, such as measurement noise. Identification of the model parameters results in completely separate transfer functions for both the deterministic and stochastic inputs. The result of the identification process is a discrete-time transfer function for the deterministic dynamics with numerator $B(q)$ and denominator $A(q)$ and a discrete-time transfer function for the stochastic dynamics with numerator $C(q)$ and denominator $D(q)$. This model is typically chosen when the dominant noise process is sensor noise because it models noise that may not be white.

**Output Error** The output error (OE) model has a noise input $e(t)$ and an exogenous input $u(t)$, which may be made up of one or multiple input signals. It is given by the formula:

$$y(t) = \sum_{i=1}^{m} \frac{B_i(q)}{A_i(q)} u_i(t) + e(t)$$

where $A(q)$ is the autoregressive equation and $B(q)$ is a moving average equation for the input signals. The block diagram representation is shown in Figure 20.9

Figure 20.9: The block diagram representation of an Output Error (OE) system model

Note that the noise is added directly to the output and does not pass through the dynamics of the system, i.e., the noise has a transfer function of 1. This model does not attempt to determine the stochastic dynamics and assumes the noise is white. It only includes the dynamics those of the system itself. The result of the identification process is a discrete-time transfer function for the deterministic dynamics with numerator $B(q)$ and denominator $A(q)$. This model is typically chosen when the parameters of the system are to be identified, but no noise model is needed.

**State Space** The discrete-time state space (SS) model has a disturbance input $w(k)$, the system state $x(k)$, a noise input $v(k)$ that directly affects the output in the measurement equation $y(k)$, and an exogenous deterministic input $u(k)$, which may be made up of one or more input signals. It is given by the formula:

$$\begin{aligned} x(k+T) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + Du(k) + v(k) \end{aligned}$$

where $T$ is the sampling interval. The result of the identification process is a discrete-time state space model for the sytem with system matrix $A$, deterministic input matrix $B$, output matrix $C$, and feedforward matrix $D$. The model for the discrete-time state space model with disturbance and noise inputs is shown in Figure 20.10.

Figure 20.10: A block diagram illustrating a state space model

This model is typically chosen for multi-input, multi-output (MIMO) systems.

### 20.4.2 Classes of models

#### Black box models

If nothing is known *a priori* about the system, the system is said to be a *black box*.

#### Grey box models

If partial model structure is known *a priori*, the system is said to be a *grey box*. For example, since the transfer function of a permanent-magnet DC motor from input voltage to output velocity can be shown second-order with no zeros, the system identification process may be used to look for coefficients of second-order parametric models. Also, some parameters may be known exactly, or at least be known within a tolerance band before the identification process takes place.

### White box models

If the system model structure is completely known *a priori*, the system is said to be a *white box*. For example, the manufacturer's data sheet for a particular temperature transmitter provides the value of the first order lag time constant, and the gain of the transmitter is a function of its calibrated range. Thus, a transfer function model can be developed for the deterministic process.

## 20.5 Model validation

One method of validating a model is to take the validation input subset, pass it through the system, and compare the model's response to the validation output subset. If the two outputs are similar, the model may be judged to be acceptable (this is often an engineering judgement call). If the two outputs are not similar enough, a model with a different structure may be determined. A good way of determining similarity is to plot the residuals, i.e., the difference between the validation output and the model output given the same input signal. If the residuals are small and uncorrelated (i.e., appear to be white noise), the model may be judged to be acceptable. Testing on residuals can include autocorrelation to determine whiteness and cross-correlation with the input to determine that the residuals are independent of the inputs.

# References


[1] L. Ljung, System Identification: Theory for the User, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[2] B. N. Finn, A. Willis, and S. Gibson, "The University of Newcastle Identification Toolbox," presented at the IFAC World Congress, 2005. [Online]. Available: http://sigpromu.org/reports/EE05010.pdf

# Chapter 21

# System Analysis

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Determine the system type number from its transfer function
2. Use the characteristic equation and Routh-Hurwitz analysis to determine the stability of a system
3. Sketch the root locus plot of a transfer function as a parameter is varied
4. Sketch the following types of frequency domain plots for a transfer function: Bode, Nyquist, and Nichols
5. Determine the phase margin and gain margin of a system from a Bode plot
6. Use computer software to perform analysis of system transfer functions

## 21.1 Type number

The type number of a signal refers to the number of integrators in the signal's Laplace transform. The type number of a system refers to the number of integrators in the system's transfer function. Since an integrator is a pole at the origin, the power of $s$ in the denominator is the type number. Given the factored pole-zero representation:

$$G(s) = \frac{K \prod_m (s - z_m)}{s^r \prod_n (s - p_n)}$$

the type number of the system is $r$. For an input signal with a given type number, a system with a type number that is equal to or greater than the signal's type number will have a response with zero steady-state error, However, if the type number of the input signal exceeds the type number of the system by one, there will a constant steady-state error. Finally, if the type number of the input signal exceeds the type number of the system by more than one, there will an infinite steady-state error.

Table 21.1: Steady Etate Error for System Type

| | **Steady State Error** | | |
|---|---|---|---|
| **Input Signal** | Type 0 | Type 1 | Type 2 |
| Unit step | $1/(1+K_p)$, $K_p = \lim_{s\to 0} G$ | 0 | 0 |
| Unit ramp | $\infty$ | $1/K_v$, $K_v = \lim_{s\to 0} sG$ | 0 |
| Unit parabola | $\infty$ | $\infty$ | $1/K_a$, $K_a = \lim_{s\to 0} s^2 G$ |

where $K_p$, $K_p$, and $K_p$ are known as the position error coefficient, the velocity error coefficient, and the acceleration error coefficient, respectively.

**Example 21.20** Consider the following system:

$$T(s) = \frac{K}{s(s+a)}$$

Determine the system's type number, and the steady-state error to unit step, unit ramp, and unit parabola input signals.

**Answer to Example 21.20** Since there is one power of $s$ alone in the denominator, the system type number is one. The type number of a unit step is also one, therefore there will be zero steady-state error to a unit step input signal since the type numbers are equal. The type number of a unit ramp is two, therefore there will be constant steady-state error to a unit ramp input signal since the type number of the signal is one more than the type number of the system. The type number of a unit parabola is three, therefore there will be infinite steady-state error to a unit parabola input signal since the type number of the signal is two more than the type number of the system.

This idea is a particular example of the internal model principle, which essentially states that a system will only produce a response with zero steady-state error to a given input signal if the system contains a model of the input signal internally.

## 21.2 Stability

A system is considered stable if all of its poles have real components in the open left-half of the complex plane (LHP). Tests for stability that rely on this requirement need the transfer function or state space model of a system to be known *a priori*.

Another way to examine stability is to look at the characteristic equation (i.e., the denominator of a transfer function). When the characteristic equation of a closed-loop transfer function equals zero, the system is unstable. Assume the following closed-loop form, $T(s) = \frac{num}{1+CG}$, this occurs when $CG = -1$, i.e., when the magnitude of $CG$ is 1 (0 dB) and the phase of $CG$ is -180°.

### 21.2.1 Routh-Hurwitz analysis

Given a transfer function, the Routh-Hurwitz stability analysis technique can be performed using the characteristic equation (i.e., the denominator of the transfer function). It is a go/no-go test - it can identify the number of poles that exist in the right half of the complex plane or on the imaginary axis, thus indicating stability, but it doesn't provide the values of the poles. Since even simple numerical analysis software can find the roots of the characteristic polynomial of a system, it is easy to see if any poles are outside of the open left-half of the complex plane. Thus, the Routh-Hurwitz stability analysis technique is no longer necessary for systems where the characteristic equation is known. However, the technique is useful when a coefficient in the characteristic equation is unknown, or if a coefficient is known only to a degree of uncertainty. For these cases, the stability of the system can be determined for a range of the unknown/uncertain parameter. This section will first show how to develop the Routh-Hurwitz array to teach the principle, then show how it can be used to determine stability margins for uncertain systems.

Given the denominator polynomial:

$$d(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0$$

the Routh-Hurwitz array of numbers may be established:

$$\begin{array}{c|ccc}
s^n & a_n & a_{n-2} & a_{n-4} \\
s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} \\
s^{n-2} & b_1 & b_2 & b_3 \\
s^{n-3} & c_1 & c_2 & c_3 \\
\vdots & \vdots & \vdots & \vdots \\
s^0 & h_1 & &
\end{array}$$

By examination, the first two rows are created using by alternating the coefficients of the characterstic polynomial, starting with the coefficient of the highest order of $s$ in the first column of the first row. The entries in the third row and beyond are calculated by using the coefficients of previous rows. For example, $b_1$ is determined using the following calculation:

$$b_1 = \frac{-\begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}$$

The entry $b_2$ is determined by:

$$b_2 = \frac{-\begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}$$

The entry $c_1$ is determined by:

$$c_1 = \frac{-\begin{vmatrix} a_{n-1} & a_{n-3} \\ b_1 & b_2 \end{vmatrix}}{b_1} = \frac{b_1 a_{n-3} - b_2 a_{n-1}}{b_1}$$

Therefore, the general rule is:

*"The entry in row $x+2$, column $y$ is calculated by forming the negative of the determinant formed using the entries from the previous two rows in column 1 and the previous two rows in column $y+1$, then dividing it by the entry from the previous row in column 1."*

The number of sign changes in the leftmost column of the array indicates the number of poles in the open right-half of the complex frequency plane.

**Example 21.21** Given a transfer function with a characteristic polynomial of $s^4 + 3s^3 + 4s^2 + 6s + 1$, form the Routh-Hurwitz array and determine whether or not the system is stable. Verify the result with the MATLAB *roots* command.

**Answer to Example 21.21** The initial part of the array is formed as:

$$\begin{array}{c|ccc} s^4 & 1 & 4 & 1 \\ s^3 & 3 & 6 & 0 \\ s^2 & & & \\ s^1 & & & \\ s^0 & & & \end{array}$$

The first element of the $s^2$ row is calculated by:

$$b_1 = \frac{-\begin{vmatrix} 1 & 4 \\ 3 & 6 \end{vmatrix}}{3} = \frac{3 \cdot 4 - 1 \cdot 6}{3} = 2$$

The second element of the $s^2$ row is calculated by:

$$b_2 = \frac{-\begin{vmatrix} 1 & 1 \\ 3 & 0 \end{vmatrix}}{3} = \frac{3 \cdot 1 - 1 \cdot 0}{3} = 1$$

The first element of the $s^1$ row is calculated by:

$$c_1 = \frac{-\begin{vmatrix} 3 & 6 \\ 2 & 1 \end{vmatrix}}{2} = \frac{2 \cdot 6 - 3 \cdot 1}{2} = 4.5$$

The second element of the $s^1$ row is calculated by:

$$c_2 = \frac{-\begin{vmatrix} 3 & 0 \\ 2 & 0 \end{vmatrix}}{2} = \frac{2 \cdot 0 - 3 \cdot 0}{2} = 0$$

Further calculations reveal the other $s^1$ row elements are all zeros.
The first element of the $s^0$ row is calculated by:

$$d_1 = \frac{-\begin{vmatrix} 2 & 1 \\ 4.5 & 0 \end{vmatrix}}{2} = \frac{4.5 \cdot 1 - 2 \cdot 0}{4.5} = 1$$

Further calculations reveal the other $s^0$ row elements are all zeros.
Thus, the full array is:

$$\begin{array}{c|ccc} s^4 & 1 & 4 & 1 \\ s^3 & 3 & 6 & 0 \\ s^2 & 2 & 1 & 0 \\ s^1 & 4.5 & 0 & 0 \\ s^0 & 1 & 0 & 0 \end{array}$$

Since no sign changes occur in the leftmost column, the system is stable.

Check: since the characteristic polynomial was known, the *roots* command was used in MATLAB to determine the poles were at [-2.312 -0.251±j1.501, -0.187], which not only shows that all poles have real parts in the open left-half of the complex frequency plane, but gives their locations as well.

Once the Routh-Hurwitz array has been created, there are some special cases that may occur that require handling.

### Zero in first column

Because the calculation of elements in the Routh-Hurwitz array requires division by an element in the first column, if that element happens to be zero, the calculation breaks down. For arrays where only the leftmost element in a row is zero, a new row must be formed with a nonzero first element in order to complete the array. Identifying the row with the leading zero as Row A, form Row B by shifting Row A to the left until a zero is no longer the first element, padding on the right with zeros, and multiplying the row formed by -1 for each shift. Then, add Row A and Row B to each other using element-by-element addition to form Row C. Finally, replace Row A in the original array with Row C and continue calculating the remainder of the original array terms.

## Example 21.22

Given a transfer function with a characteristic polynomial of $2s^4 + 3s^3 + 4s^2 + 6s + 7$, form the Routh-Hurwitz array, then determine the number of poles in the open right-half of the complex frequency plane. Verify the result with the MATLAB *roots* command.

**Answer to Example 21.22**

The initial part of the array is formed as:

$$\begin{array}{c|ccc} s^4 & 2 & 4 & 7 \\ s^3 & 3 & 6 & 0 \\ s^2 & 0 & 7 & 0 \\ s^1 & & & \\ s^0 & & & \end{array}$$

The $s^2$ row becomes Row A. Shifting this once the the left, padding with zeros, and multiplying by -1 (due to only one shift to the left) forms Row B = [-7 0]. Adding Rows A and B together forms Row C = [-7 7]. This now goes into the original array in place of Row A and the remaining elements are calculated as indicated below.

$$\begin{array}{c|ccc} s^4 & 2 & 4 & 7 \\ s^3 & 3 & 6 & 0 \\ s^2 & -7 & 7 & 0 \\ s^1 & 9 & 0 & 0 \\ s^0 & 7 & 0 & 0 \end{array}$$

Since there are two sign changes in the first column, there are two poles in the open right-half of the complex frequency plane, and the system is unstable.

Check: since the characteristic polynomial was known, the *roots* command was used in MATLAB to determine the poles were at [0.384±j1.337, -1.134±j0.722], which not only shows two poles whose real parts are in the open right-half of the complex frequency plane, but gives all of the pole locations as well.

**A row of zeros**

If a row in the Routh-Hurwitz array consists entirely of zeros, there will be poles that are located symmetrically about the origin in the complex frequency plane. If using the Routh-Hurwitz array as an go/no-go test of stability, the row of zeros indicates a condition where poles exist symmetrically along either the real axis or the imaginary axis. This is enough to indicate that the system is unstable. However, when a parameterized characteristic polynomial occurs, it is of interest to be able to determine all of the elements in the Routh-Hurwitz array. Since division by zero is not allowed, the row of zeros must be replaced with a nonzero row. This is accomplished by creating a divisor polynomial $p(s)$ from the row above the row containing the zeros and finding its derivative with respect to $s$, then replacing the row of zeros with the coefficients in the derivative. The divisor polynomial is created by multiplying the first term in the row by the power of $s$ associated with the row, then multiplying each additional term by decreasing the power of $s$ by a factor of 2 each time, and adding all of the terms. For example, assume the $s^2$ row was [3 3] and the $s^1$ row was [0 0]. The first term in the divisor polynomial is created by multiplying 3 by $s^2$ to get $3s^2$. The second term in the divisor polynomial is created by multiplying 3 by $s^0$ to get 3. Therefore, the divisor polynomial is $p(s) = 3s^2 + 3$. Taking the derivative with respect to $s$ gives $6s$, therefore the new $s^1$ row is [6 0].

## Example 21.23

Given a unity gain feedback system with proportional controller K, and a system transfer function of:

$$\frac{10}{s^3 + 9s^2 + 23s + 15}$$

form the Routh-Hurwitz array, then determine the range of K for which the system is stable. Verify the result with the MATLAB *roots* command.

**Answer to Example 21.23** The closed-loop transfer function is:

$$T(s) = \frac{10K}{s^3 + 9s^2 + 23s + (15 + 10K)}$$

The Routh-Hurwitz array is formed as:

$$\begin{array}{c|cc} s^3 & 1 & 23 \\ s^2 & 9 & 15+10K \\ s^1 & \frac{9\cdot 23 - 1\cdot(15+10K)}{9} & \\ s^0 & 15+10K & \end{array}$$

Evaluation of the first column terms indicates that 15+10K < 207 and 15+10K > 0. These two inequalities lead to a range for stable operation that is -1.5 < K < 19.2.

Check: The characteristic polynomial with K = -1.5 is $s^3 + 9s^2 + 23s$. The *roots* command was used in MATLAB to determine the poles were at [-4.5±j1.6583, 0], which shows a pole at the origin of the complex plane, and therefore the system is unstable. The characteristic polynomial with K = 19.2 is $s^3 + 9s^2 + 23s + 207$. The *roots* command was used in MATLAB to determine the poles were at [0±j4.7958, -9], which shows a complex pole pair on the $j\omega$ axis, and therefore the system is unstable. Thus, the range for stable operation is verified to be -1.5 < K < 19.2.

## 21.3 Root locus analysis

A root locus is the set of points traced in the complex plane by the closed-loop poles (i.e., roots of the characteristic equation) of a system as a single system parameter is varied from 0 to infinity. The closed-loop poles can only lie on the root locus, nowhere else. When the system parameter $K$ is zero, the closed-loop poles are at the open loop poles. As $K \to \infty$, the closed-loop poles approach the open loop zeros (if the transfer function is proper) or approach infinity along an asymptote (if the transfer function is strictly proper). If there is an excess of zeros, the locus extends from infinity along an asymptote.

The following procedure may be used for drawing the root locus:

- Create a pole-zero plot of the open-loop poles and zeros
- Find real axis portions of locus
- Find asymptote angles
- Find asymptote centroids
- Find breakaway/entry points from the real axis
- Find departure/approach angles to complex poles or zeros

A root locus may be drawn according to the following rules. Given an open-loop transfer function $T$ with $n$ poles and $m$ zeros that is rational (i.e., $n \geq m$):

**Rule 1 - Branches** The locus consists of parameterized continuous curves that start at each of the open-loop poles of the system when the parameter $K$ is zero and end at the open-loop zeros when the parameter $K$ is infinite. If the system has more poles than zeros, then $n - m$ branches of the locus will end at zeros located at infinity.

**Rule 2 - Real Axis** All points on the real axis to the left of an odd number of poles and zeros are on the root locus.

**Rule 3 - Asymptote Angles** As $K \to \infty$, the branches are asymptotic to lines at angles $\theta$ given by:

$$\theta = 180^\circ \frac{(2q+1)}{n-m}$$

where $q = 0, 1, 2, ..., n-m-1$ until all $|n-m|$ angles are obtained. This essentially means that a locus with one branch is asymptotic to $-180^\circ$, a locus with two branches will have asymptotes at $\pm 90^\circ$, a locus with three branches will have asymptotes at $\pm 60^\circ$ and $-180^\circ$, a locus with four branches will have asymptotes at $\pm 45^\circ$ and $\pm 135^\circ$, etc.

**Rule 4 - Asymptote Centroid** The centroid of the asymptotes ($\sigma$) is on the real axis at:

$$\sigma = \frac{\sum_n \Re(p_n) - \sum_m \Re(z_m)}{n-m}$$

where $\Re$ indicates the real part of the pole or zero.

**Rule 5 - Real Axis Breakaway and Entry** Breakaway and entry points (points where the locus leaves or enters the real axis) occur at perpendicularly to the real axis (i.e., at $\pm 90^\circ$), at points given by the roots of the polynomial $P$:

$$P = N\frac{dD}{ds} - D\frac{dN}{ds}$$

where $N$ and $D$ are the numerator and denominator polynomials, respectively.

**Rule 6 - Angle of Departure from Complex Poles** The angle of the locus departure from a complex pole of order $k$ is given by $\phi$:

$$\phi = \frac{-\sum \text{other pole angles} + \sum \text{zero angles} + (1+2i)180^\circ}{k}$$

for i = 0, 1, 2, ... (k-1). A pole angle or zero angle is always referenced counterclockwise from the real axis.

**Rule 7 - Angle of Approach to Complex Zeros** The angle of the locus approach to a complex zero of order $k$ is given by $\phi'$:

$$\phi' = \frac{\sum \text{pole angles} - \sum \text{other zero angles} - (1+2i)180^\circ}{k}$$

for i = 0, 1, 2, ... (k-1).

**Example 21.24** Plot the root locus for the following transfer function by hand:

$$T(s) = \frac{5s+5}{2s+1}$$

**Answer to Example 21.24** This transfer function has a single pole at -0.5 and a single zero at -1. Since only one pole exists, Rule 1 says that there is only one branch of the root locus, and it begins at the pole and ends at the zero. Rule 2 says that the root locus will lie on the real axis to the left of the pole at -0.5 since it gives an odd number of poles and zeros when passed (i.e., 1), but then it ends at the zero at -1 since now an even number of poles and zeros have been passed (i.e., 2). The plot is shown below:

**Example 21.25** Plot the root locus for the following transfer function by hand:

$$T(s) = \frac{1}{s^2 + 15s + 50}$$

**Answer to Example 21.25** This transfer function has two real poles at $-5$ and $-10$ and no zeros. Since two poles and no zeros exist, Rule 1 says that there are two branches of the root locus, and they begin at the poles and end at zeros at infinity. Rule 2 says that a portion of the root locus will lie on the real axis to the left of the pole at $-5$ because an odd number of poles and zeros has been passed moving from left to right, (i.e., 1) and end at the pole at $-10$ because an even number of poles and zeros has been passed moving from left to right (i.e., 2). Rule 3 can be used to calculate that the branches are asymptotic to lines at $\pm 90°$. Rule 4 may be used to show that the centroid of the asymptotes lies at $-7.5$. Rule 5 applies because the root locus moves left from the pole at $-5$ and right from the pole at $-10$. The root locus must then break away from the axis at $-7.5$ due to

the following:

$$\begin{aligned} P &= 1\left(\frac{d(s^2+15s+50)}{s}\right) - \left(s^2+15s+50\right)\left(\frac{1}{s}\right) \\ 0 &= 2s+15 \\ s &= -7.5 \end{aligned}$$

Note that for a system with only two real poles, the asymptote centroid and the breakaway position are at the same location. Rule 5 applies to yield departure angles from the real axis of $\pm 90°$. Rules 6 and 7 do not apply. The plot is shown below:

**Example 21.26** Plot the root locus for the following transfer function by hand:

$$T(s) = \frac{1}{s^2+s+1}$$

**Answer to Example 21.26** This transfer function has a pair of complex poles at $-0.5 \pm j0.866$ and no zeros. Since two poles and no zeros exist, Rule 1 says that there are two branches of the root locus, and they begin at the poles and end at zeros at infinity. Rule 2 says that the root locus will not lie on the real axis because there are an even number of poles and zeros to the right of the pole location (i.e., 0) and an even number of poles and zeros to the left of the pole location (i.e., 2). Rule 3 can be used to calculate that the branches are asymptotic to lines at $\pm 90°$. Rule 4 may be used

to show that the centroid of the asymptotes lies at $-0.5$. Rule 5 does not apply because the root locus cannot leave or enter the real axis as there is only one branch on it. Rule 6 applies to yield complex pole departure angles of $\pm 90°$ for the poles at $-0.5 \pm j0.866$, respectively. This is shown by the following:

$$\frac{-(-90°) + 0 + (1 + 2 \cdot 0)180°}{1} = 90°$$
$$\frac{-(90°) + 0 + (1 + 2 \cdot 1)180°}{1} = -90°$$

Rule 7 does not apply. The plot is shown below:

**Example 21.27** Plot the root locus for the following transfer function by hand:

$$T(s) = \frac{1}{s^3 + 11s^2 + 44s + 34}$$

**Answer to Example 21.27** This transfer function has a pair of complex poles at $-5 \pm j3$, a single real pole at $-1$, and no zeros. Since three poles and no zeros exist, Rule 1 says that there are three branches of the root locus, and they begin at the poles and end at zeros at infinity. Rule 2 says that one branch of the root locus will lie on the real axis to the left of the pole at $-1$ because there are an even number of poles and zeros to the right of the pole location (i.e., 0) and an even number of

poles and zeros to the left of the pole location (i.e., 2). Repeated application of Rule 3 can be used to calculate that the three branches are asymptotic to lines at 180° (this is the negative real axis, and confirms that particular locus branch determined using Rule 2) and ±60°. This is due to the following calculations:

$$\begin{aligned}
\theta_0 &= 180^\circ \frac{(2 \cdot 0 + 1)}{3 - 0} = 60^\circ \\
\theta_1 &= 180^\circ \frac{(2 \cdot 1 + 1)}{3 - 0} = 180^\circ \\
\theta_2 &= 180^\circ \frac{(2 \cdot 2 + 1)}{3 - 0} = 300^\circ = -60^\circ
\end{aligned}$$

Rule 4 may be used to show that the centroid of the asymptotes lies at $-11/3$ using the following calculation:

$$\sigma = \frac{(-1 - 5 - 5) - (0)}{3 - 0} = \frac{-11}{3}$$

where zero has been used to show that there are no contributions from zeros in the formula given for Rule 4.

Rule 5 does not apply, as the root locus remains on the real axis due to the real pole at $-1$. Rule 6 can then be used repeatedly for the two complex poles. The departure angles in question for the pole at $-5 + j3$ are shown below:

The angles for the other pole follow similarly. The calculations yield breakaway angles of ∓53.13°

for the poles at $-5 \pm j3$, respectively. This is shown by the following:

$$\frac{-(90^\circ + \arctan(3/4)^\circ) + 0 + (1 + 2 \cdot 0)180^\circ}{1} = -53.13^\circ$$
$$\frac{-(-90^\circ - \arctan(3/4)^\circ) + 0 + (1 + 2 \cdot 1)180^\circ}{1} = 53.13^\circ$$

Rule 7 does not apply as there are no zeros in the transfer function. The plot is shown below:

Adding poles or zeros in the controller transfer function changes the root locus since it changes the open-loop poles or zeros. Adding controller poles tends to decrease stability because the root locus asymptote angles change and the root locus can end up in the right half of the complex plane, meaning some process parameter values will make the system unstable. Adding controller zeros tends to increase stability because the root locus asymptote angles change and the root locus moves away from the right half of the complex plane as the angles change.

## 21.4 Frequency response analysis

For an LTI system, a sinusoidal input produces a sinusoidal output whose amplitude and phase are changed based on the frequency response of the system. Multiple plots can indicate a system's frequency response. They contain the same information, but represent that information in different forms. All involve sweeping a unit sine wave at a given frequency through a system.

### 21.4.1 Bode plot

The Bode plot consists of two separate plots: a plot of the logarithmic gain or magnitude of a system (usually in dB) versus a logarithmic frequency scale (in Hz or rad/s) and a plot of the phase angle (usually in degrees) of a system versus the same logarithmic frequency scale.

#### Asymptotic Bode plots - the easy way

If sketching by hand, asymptotic Bode plots are usually sketched based on the Bode form of the transfer function. While many textbooks instruct readers to plot the individual pole and zero contributions, then add them all, this is not the easiest way of producing an asymptotic Bode plot. Adding the contributions typically leads to arithmetic mistakes that result in an incorrect plot.

A simpler method starts by determining the frequency range required for the plot. The frequency range for the plot should begin at one decade below the lowest frequency term and end at one decade above the highest frequency term. This allows the full range of phase effects for every term to be plotted.

Once the frequency endpoints have been determined, the magnitude at each of the endpoints can be determined by substituting each endpoint frequency in turn into the transfer function to find the magnitude, then converting the magnitudes determined into dB. Next, the pole and zero frequencies are marked on the frequency axis. Finally, the magnitude plot can be created by changing slopes as appropriate between the endpoint magnitudes as each pole or zero frequency is passed moving from low frequency to high frequency. Each real pole contributes a slope of -20 dB/decade and each real zero contributes a slope of +20 dB/decade, while each complex pole pair contributes a slope of -40 dB/decade and each complex zero pair contributes a slope of +40 dB/decade.

The phase at the low frequency endpoint can be determined by whether or not any integrators occur in the transfer function, as each integrator contributes a phase angle of -90°. The phase at the high frequency endpoint can be determined by adding the phase angles contributed by the poles and zeros to the phase determined at the low frequency endpoint. Add -90°for each real pole and +90°for each real zero; add Add -180°for each complex pole pair and +180°for each complex zero pair. Next, the pole and zero frequencies are marked on the frequency axis, and a region of influence that extends one decade above and below each is marked out. Finally, using the fact that each real pole contributes a slope of -45°/decade over its two-decade region of influence and each zero contributes a slope of +45°/decade over its two-decade region of influence, the phase plot can be created by changing slopes as appropriate between the endpoint phase angles.

#### Gain margin

Gain margin is the amount of gain (in dB) that can be added to a system before it will become unstable. It can be determined from a Bode plot by determining the gain difference between the Bode magnitude plot and the 0 dB reference at the frequency where the phase crosses -180°. This result is the smallest gain that makes the magnitude exceed 0 dB when the phase is -180°.

**Phase margin**

Phase margin is the amount of phase lag (in degrees) that can be added to a system before it will become unstable. It can be determined from a Bode plot by determining the phase difference between the Bode phase plot and the -180°reference at the frequency where the gain crosses 0 dB.

### 21.4.2 Nyquist plot

A Nyquist plot is a plot of the imaginary component versus the real component as the parameter of frequency varies from 0 to $\infty$. A Nyquist plot can easily be created from a Bode plot by converting the polar value of magnitude (when it is not in dB) and phase angle that occur at various frequencies to rectangular coordinates. If the coordinates of a particular point on the Nyquist plot are known, the angular value determined by trigonometry based on the real and imaginary components is the phase angle that occurs at a particular frequency, and the magnitude of the vector drawn from the origin to that point is the linear magnitude (not in dB). Thus, magnitude and phase information are easily calculated from a Nyquist plot (whereas on a Bode plot they are directly read).

### 21.4.3 Nichols plot

A Nichols plot is a plot of the gain (again, not in dB) versus the phase of a system's response as the parameter of frequency varies from 0 to $\infty$. This is easily developed from a Bode plot of gain and phase by choosing values of the two parameters at various frequencies.

# References


[1] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, Design of Feedback Control Systems, 4th ed. New York, NY: Oxford University Press, 2002.

[2] C. L. Phillips and R. D. Harbor, Feedback Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[3] J. R. Rowland, Linear Control Systems: Modeling, Analysis, and Design. New York, NY: John Wiley & Sons, 1986.

[4] R. C. Dorf and R. H. Bishop, Modern Control Systems, 10th ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2005.

[5] F. J. Hale, Introduction to Control System Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[6] A. P. Sage, Linear Systems Control. Champaign, IL: Matrix Publishers, 1978.

[7] M. Driels, Linear Control Systems Engineering. New York, NY: McGraw-Hill, 1995.

[8] W. J. Palm, Control Systems Engineering. New York, NY: John Wiley and Sons, 1986.

[9] S. C. Gupta and L. Hasdorff, Fundamentals of Automatic Control. New York, NY: John Wiley and Sons, 1970.

[10] A. F. D'Souza, Design of Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[11] T. E. Marlin, Process Control, 2nd ed. New York, NY: McGraw-Hill, 2000.

[12] B. A. Ogunnaike and W. H. Ray, Process Dynamics, Modeling, and Control. New York, NY: Oxford University Press, 1994.

[13] A. Tewari, Modern Control Design with MATLAB and Simulink. New York, NY: John Wiley and Sons, 2002.

[14] C. A. Smith and A. Corripio, Principles and Practice of Automatic Process Control, 3rd ed. New York, NY: John Wiley and Sons, 2006.

[15] H. V. Vu and R. S. Esfandiari, Dynamic Systems: Modeling and Analysis. Boston, MA: McGraw-Hill, 1997.

[16] M. Gopal, Control Systems: Principles and Design. Boston, MA: McGraw-Hill, 2008.

# Chapter 22

# Principles of Feedback Controllers

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: reverse action, direct action, proportional controller, steady-state error (offset), proportional band, proportional-integral controller, proportional-integral-derivative controller, reset windup, bumpless transfer, manual, automatic
2. Describe how on-off controllers function and discuss their limitations
3. State PID controller algorithms and explain how the individual terms respond to error signals
4. Explain advantages and disadvantages of variations of PID control
5. Compare and contrast digital PID implementation platforms

This chapter explains some of the fundamental principles of automatic process control.

## 22.1 Basic feedback control principles

Before beginning a discussion on process control, here are a few key terms. The *process* is the physical system to monitor and control. For the sake of illustration, consider a heat exchanger that uses high-temperature steam to transfer heat to a lower-temperature liquid. Heat exchangers are used frequently in the chemical industries to maintain the necessary temperature of a chemical solution, so the desired blending, separation, or reactions can occur. A very common design of heat exchanger is the "shell-and-tube" style, where a metal shell serves as a conduit for the chemical solution to flow through, while a network of smaller tubes runs through the heating space, carrying steam or some other heating medium. The hotter steam flowing through the tubes transfers heat energy to the cooler process fluid surrounding the tubes, inside the shell of the heat exchanger shown in Figure 22.1.

Figure 22.1: A steam-to-water shell-and-tube heat exchanger diagram

In this case, the *process* is the entire heating system, consisting of the fluid to be heated, the heat exchanger, and the steam delivering the required heat energy. In order to maintain steady control of the process fluid's exiting temperature, it must be measured and that measurement available in

signal form so it may be interpreted by other instruments taking some form of control action. In instrumentation terms, the measuring device is known as a *transmitter*, because it *transmits* the process measurement in the form of a signal.

Transmitters are represented in process diagrams by instrument bubbles with identifying letters inside as described in Chapter 2, in Figure 22.2 this is "TT," which stands for **T**emperature **T**ransmitter:

Figure 22.2: A temperature transmitter on the outlet water stream

The signal coming from the transmitter (shown in the illustration by the dashed line), representing the heated fluid's exiting temperature, is called the *process variable*. Like a variable in a mathematical equation that represents some story-problem quantity, this signal represents the measured quantity to be controlled in the process.

In order to exert control over the process variable, there must be some way of altering fluid flow through the heat exchanger, either of the process fluid, the steam, or both. Generally, it makes more sense to alter the flow of the heating medium (the steam), and let the process fluid flow rate be dictated by the demands of the larger process. If this heat exchanger were part of an oil refinery unit, for example, it would be far better to throttle steam flow to control oil temperature rather than to throttle the oil flow itself, since altering the oil's flow will undoubtedly affect other processes upstream and downstream of the exchanger. Ideally, the exchanger will act as a device that provides even, consistent temperature oil out, for any given temperature and flow-rate of oil in.

One convenient way to throttle steam flow into the heat exchanger is to use a control valve (labeled "TV" because it is a **T**emperature **V**alve) as shown in Figure 22.3. In general terms, a control valve is known as a *final control element*. Other types of final control elements exist (servo motors, variable-flow pumps, and other mechanical devices used to vary some physical quantity at will), but valves are the most common, and probably the simplest to understand. With a final control element in place, the steam flow becomes known as the *manipulated variable*, because it is the quantity that will be manipulated in order to gain control over the process variable:

Figure 22.3: A temperature valve on the inlet steam stream

Valves come in a wide variety of sizes and styles. Some valves are hand-operated: that is, they have a "wheel" or other form of manual control that may be moved to "pinch off" or "open up" the flow passage through the pipe. Other valves come equipped with signal receivers and positioner devices, which move the valve mechanism to various positions at the command of a signal (usually an electrical signal, like the type output by transmitter instruments). This feature allows for remote control, so a human operator or computer device may exert control over the manipulated variable from a distance.

Finally, the most critical component of the heat exchanger temperature control system is the *controller*. This is a device designed to interpret the transmitter's process variable signal and decide

how far open the control valve needs to be in order to maintain that process variable at the desired value.

In Figure 22.4, the circle with the letters "TC" in the center represents the controller. Those letters stand for **T**emperature **C**ontroller, since the process variable being controlled is the process fluid's *temperature*. Usually, the controller consists of a computer making automatic decisions to open and close the valve as necessary to stabilize the process variable at some predetermined *setpoint*.

Figure 22.4: A temperature controller linking the sensing function with the manipulating function

These elements comprise the essentials of a *feedback control system*: the *process* (the system to be controlled), the *process variable* (the specific quantity to be measured and controlled), the *transmitter* (the device used to measure the process variable and output a corresponding signal), the *controller* (the device that decides what to do to bring the process variable as close to setpoint as possible), the *final control element* (the device that directly exerts control over the process), and the *manipulated variable* (the quantity to be directly altered to effect control over the process variable).

Feedback control may be viewed as a sort of information "loop," from the transmitter (measuring the process variable), to the controller, to the final control element, and through the process itself,

back to the transmitter. Ideally, a process control "loop" not only holds the process variable at a steady level (the setpoint), but also maintains control over the process variable given changes in setpoint, and even changes in other variables of the process as shown in Figure 22.5.

Figure 22.5: Diagram of a feedback control system

For example, if the temperature setpoint in the heat exchanger process were to be raised, the controller would automatically call for more steam flow by opening the control valve, thus introducing more heat energy into the process, thus raising the temperature to the new setpoint level. If the process fluid flow rate (an uncontrolled, or *wild* variable) were to suddenly increase, the heat exchanger outlet temperature would fall due to the physics of heat transfer, but once this drop was detected by the transmitter and reported to the controller, the controller would automatically call for additional steam flow to compensate for the temperature drop, thus bringing the process variable back in agreement with the setpoint. Ideally, a well-designed and well-tuned control loop will sense and compensate for *any* change in the process or in the setpoint, the end result being a process variable value that always holds steady at the setpoint value.

Many types of processes lend themselves to feedback control. Consider an aircraft autopilot system, keeping an airplane on a steady course heading: reading the plane's heading (process variable) from an electronic compass and using the rudder as a final control element to change the plane's "yaw." An automobile's "cruise control" is another example of a feedback control system, with the process variable being the car's velocity, and the final control element being the engine's throttle. Steam boilers with automatic pressure controls, electrical generators with automatic voltage and frequency controls, and water pumping systems with automatic flow controls are further examples of how feedback may be used to maintain control over certain process variables.

Modern technology makes it possible to control nearly anything that may be measured in an industrial process. This extends beyond the pale of simple pressure, level, temperature, and flow variables to include even certain chemical properties.

In municipal water and wastewater treatment systems, numerous chemical quantities must be measured and controlled automatically to ensure maximum health and minimum environmental impact. Take for instance the chlorination of treated wastewater before it leaves the wastewater treatment facility into a large body of water such as a river, bay, or ocean as shown in the P&ID in Figure 22.6. Chlorine is added to the water to kill any residual bacteria so they do not consume oxygen in the body of water they are released to. Too little chlorine added, and not enough bacteria are killed, resulting in a high *biological oxygen demand* or *BOD* in the water which will asphyxiate the fish swimming in it. Too much chlorine added, and the chlorine itself poses a hazard to marine life. Thus, the chlorine content must be carefully controlled at a particular setpoint, and the control system must take aggressive action if the dissolved chlorine concentration strays too low or too high:

Figure 22.6: Diagram of a chlorine concentration control system

*Algorithms* are used in the controller to maintain a process variable at setpoint. For the scope of this topic, an "algorithm" is a mathematical relationship between the process variable and setpoint inputs of a controller, and the output (manipulated variable). Control algorithms determine *how* the manipulated variable quantity is deduced from PV and SP inputs, and range from the elementary to the very complex. In the most common form of control algorithm, the so-called "PID" algorithm, calculus is used to determine the proper final control element action for any combination of input signals.

## 22.2 On/off control

Once in a large manufacturing facility, a mechanic asked what it was that an instrument technician did. The instrument technician began to explain his job, which was essentially to calibrate, maintain,

troubleshoot, document, and modify (as needed) all automatic control systems in the facility. The mechanic seemed puzzled during the explanation of "tuning" loop controllers, especially those controllers used to maintain the temperature of large, gas-fired industrial furnaces holding many tons of molten metal. "Why does a controller have to be 'tuned'?" he asked. "All a controller does is turn the burner on when the metal's too cold, and turn it off when it becomes too hot!"

In its most basic form, the mechanic's assessment of the control system was correct: to turn the burner on when the process variable (molten metal temperature) drops below setpoint, and turn it off when it rises above setpoint. However, the actual algorithm is much more complex than that, finely adjusting the burner intensity according to the amount of *error* between PV and SP, the amount of time the error has accumulated, and the rate-of-change of the error over time. In his limited observation of the furnace controllers, though, he had noticed nothing more than the full-on/full-off action of the controller.

The technical term for a control algorithm that merely checks for the process variable exceeding or falling below setpoint is *on/off control*. In colloquial terms, it is known as *bang-bang* control, since the manipulated variable output of the controller rapidly switches between fully "on" and fully "off" with no intermediate state. Control systems this crude usually provide very imprecise control of the process variable. Consider the example of the shell-and-tube heat exchanger with simple on/off control[1]. Figure 22.7 shows the performance of a typical on-off control system.

Figure 22.7: Trend showing performance of an on-off control system

As can be seen, the degree of control is rather poor. The process variable "cycles" between the upper and lower setpoints (USP and LSP) without ever stabilizing at the setpoint, because that would require the steam valve to be position somewhere *between* fully closed and fully open.

[1] To be precise, this form of on/off control is known as *differential gap* because there are two setpoints with a gap in between. While on/off control is possible with a single setpoint (FCE on when below setpoint and off when above), it is usually not practical due to the frequent cycling of the final control element.

This simple control algorithm may be adequate for temperature control in a house, but not for a sensitive chemical process! Imagine what it would be like if an automobile's cruise control system relied on this algorithm. Not only is the lack of precision a problem, but the frequent cycling of the final control element may contribute to premature failure due to mechanical wear. In the heat exchanger scenario, thermal cycling (hot-cold-hot-cold) will cause metal fatigue in the tubes, resulting in a shortened service life. Furthermore, every excursion of the process variable above setpoint is wasted energy, because the process fluid is being heated to a greater temperature than what is necessary.

Clearly, the only practical answer to this dilemma is a control algorithm able to *proportion* the final control element rather than just operate it at zero or full effect (the control valve fully closed or fully open). This, in its simplest form, is called *proportional control*.

## 22.3 Proportional-only control

Here is where math starts to enter the algorithm: a proportional controller calculates the difference between the process variable signal and the setpoint signal, and calls it the *error*. This is a measure of how far off the process is deviating from its setpoint, and may be calculated as SP−PV or as PV−SP, depending on whether or not the controller has to produce an *increasing* output signal to cause an increase in the process variable, or output a *decreasing* signal to do the same thing[2]. This choice in how the subtraction occurs determines whether the controller will be *reverse-acting* or *direct-acting*. The direction of action required of the controller is determined by the nature of the process, transmitter, and final control element. In this case, it is assumed that an increasing output signal sent to the valve results in increased steam flow, and consequently higher temperature, so the algorithm will need to be reverse-acting (i.e., an increase in measured temperature results in a decrease in output signal; error must be calculated as SP−PV). This error signal is then multiplied by a constant value called the *gain*, which is programmed into the controller. The resulting figure, plus a "bias" quantity, becomes the output signal sent to the valve to proportion it:

$$m = K_p e + b$$

Where,
$m$ = Controller output
$e$ = Error (difference between PV and SP)
$K_p$ = Proportional gain
$b$ = Bias

If this equation appears to resemble the standard slope-intercept form of linear equation ($y = mx + b$), it is more than coincidence. Often, the response of a proportional controller is shown graphically as a line, the slope of the line representing gain and the y-intercept of the line representing the output bias point, or what value the output signal will be when there is zero error (PV precisely equals SP) as shown in Figure 22.8

[2]The ability to select the order of subtraction is needed because controllers typically only allow gains $\geq 0$.

Figure 22.8: Output versus error in proportional control system

The bias value is 50% and the gain of the controller is 1 in the graph in Figure 22.8.

Proportional controllers give us a choice as to how "sensitive" the controller should be to changes in process variable (PV) and setpoint (SP). With the simple on/off ("bang-bang") approach, there was no adjustment. Here, though, the controller can be programmed for any desired level of aggressiveness. The gain value ($K_p$) of a controller is something which may be altered by a technician or engineer. In pneumatic controllers, this takes the form of a lever or valve adjustment; in analog electronic controllers, a potentiometer adjustment; in digital control systems, a programmable parameter.

If the controller could be configured for infinite gain, its response would duplicate on/off control. That is, *any* amount of error will result in the output signal becoming "saturated" at either 0% or 100%, and the final control element will simply turn on fully when the process variable drops below setpoint and turn off fully when the process variable rises above setpoint. Conversely, if the controller is set for zero gain, it will become completely unresponsive to changes in either process variable *or* setpoint: the valve will hold its position at the bias point no matter what happens to the process.

Obviously, then, the gain must be set somewhere between infinity and zero in order for this algorithm to function any better than on/off control. Just how much gain a controller needs to have depends on the process and all the other instruments in the control loop.

If the gain is set too high, there will be oscillations as the PV converges on a new setpoint value as shown in Figure 22.9.

Figure 22.9: Example of a system with a proportional gain that is too high

If the gain is set too low, the process response will be stable under steady-state conditions, but "sluggish" to changes in setpoint because the controller does not take aggressive enough action to cause quick changes in the process as shown in Figure 22.10.

Figure 22.10: Example of a system with a proportional gain that is too low

With proportional-only control, the only way to obtain fast-acting response to setpoint changes

or "upsets" in the process is to set the gain constant high enough that some "overshoot" results as shown in Figure 22.11. Note, however, that the conflicting objectives of minimizing rise time, overshoot, and settling time are all met well (this is a qualitative judgement).

Figure 22.11: Example of a system with a proportional gain that gives good response

As with on/off control, instances of overshoot (the process variable rising above setpoint) and undershoot (drifting below setpoint) are generally undesirable, and for the same reasons. Ideally, the controller will be able to respond in such a way that the process variable is made equal to setpoint as quickly as the process dynamics will allow, yet with no substantial overshoot or undershoot. With plain proportional control, however, this ideal goal is nearly impossible.

An unnecessarily confusing aspect of proportional control is the existence of two completely different ways to express the "aggressiveness" of proportional action. In the proportional-only equation shown earlier, the degree of proportional action was specified by the constant $K_p$, called *gain*. However, there is another way to express the sensitivity of proportional action, and that is to state the percentage of error change necessary to make the output ($m$) change by 100%. Mathematically, this is the inverse of gain, and it is called *proportional band* (PB):

$$K_p = \frac{1}{\text{PB}} \qquad \text{PB} = \frac{1}{K_p}$$

Gain is always specified as a unitless value[3], whereas proportional band is always specified as a percentage. For example, a gain value of 2.5 is equivalent to a proportional band value of 40%.

[3] In electronics, the unit of *decibels* is commonly used to express gains. Thankfully, the world of process control was spared the introduction of decibels as a unit of measurement for controller gain. The last thing needed is a *third* way to express the degree of proportional action in a controller!

Due to the existence of these two completely opposite conventions for specifying proportional action, the proportional term of the control equation may be written differently depending on whether the author assumes the use of gain or the use of proportional band:

$$K_p = \text{gain} \qquad \text{PB} = \text{proportional band}$$

$$K_p e \qquad\qquad \frac{1}{\text{PB}} e$$

Many modern digital electronic controllers allow the user to conveniently select the unit they wish to use for proportional action. However, even with this ability, anyone tasked with adjusting a controller's "tuning" values may be required to translate between gain and proportional band, especially if certain values are documented in a way that does not match the unit configured for the controller.

When communicating the proportional action setting of a process controller, always be careful to specify either "gain" or "proportional band" to avoid ambiguity. *Never* simply say something like, "The proportional setting is twenty," for this could mean either:

- Gain = 20; Proportional band = 5% . . . *or* . . .
- Proportional band = 20%; Gain = 5

## 22.4 Proportional-only offset

A fundamental limitation of proportional control has to do with its response to changes in setpoint and changes in process *load*. A "load" in a controlled process is any variable subject to change which has an impact on the variable being controlled (the process variable), but is not subject to correction by the controller. In other words, a "load" is any variable in the process that cannot or is not controlled, yet affects the process variable under control.

In the hypothetical heat exchanger system shown in Figure 22.12, the temperature of the incoming process fluid is an example of a load.

Figure 22.12: Diagram illustrating load disturbances

If the incoming fluid temperature were to suddenly decrease, the immediate effect this would have on the process would be to decrease the outlet temperature (which is the temperature to be maintained at a steady value). It should make intuitive sense that a colder incoming fluid will require more heat input to raise it to the same outlet temperature as before. If the heat input remains the same (at least in the immediate future), this colder incoming flow must make the outlet flow colder than it was before. Thus, incoming feed temperature has an impact on the outlet temperature, and the control system has no way to regulate how warm or cold the process fluid is before it enters the heat exchanger. This is precisely the definition of a "load."

Of course, it is the job of the controller to counteract any tendency for the outlet temperature to stray from setpoint, but as will be discussed this cannot be perfectly achieved with proportional control alone.

Let us carefully analyze the scenario of sudden inlet fluid temperature decrease to see how a proportional controller would respond. Imagine that previous to this sudden drop in feed temperature, the controller was controlling outlet temperature exactly at setpoint (PV = SP) and everything was stable. Recall that the equation for a proportional controller is as follows:

$$m = K_p e + b$$

Where,
$m$ = Controller output
$e$ = Error (difference between PV and SP)
$K_p$ = Proportional gain
$b$ = Bias

A decrease in feed temperature will result in a decrease of outlet temperature with all other factors remaining the same. From the equation, a decrease in process variable (PV) will cause the Output value in the proportional controller equation to increase. This means a wider-open steam valve, admitting more heating steam into the heat exchanger. All this is good, the controller should call for more steam as the outlet temperature drops. But will this action be enough to bring the outlet temperature back up to setpoint where it was prior to the load change? Unfortunately it will not, although the reason for this may not be evident upon first inspection.

In order to prove that the PV will never go back to SP as long as the incoming feed temperature has dropped, let us imagine for a moment that somehow it did. According to the proportional controller equation, this would mean that the steam valve would resume its old pre-load-change position, only letting through the original flow rate of steam to heat the process fluid. Obviously, if the incoming process fluid is colder than before, and the flow rate is unchanged, the same amount of heat input (from steam) will be inadequate to maintain the outlet temperature at setpoint. If it were adequate, the outlet temperature never would have decreased and the controller never would have had to adjust the steam valve position at all. In other words, if the steam valve goes back to its old position, the outlet temperature will fall just as it did when the incoming flow suddenly became colder. This tells us the controller *cannot* bring the outlet temperature exactly to setpoint by proportional action alone.

What *will* happen is that the controller's output will increase with falling outlet temperature, until there is enough steam flow admitted to the heat exchanger to prevent the temperature from falling any further. But in order to maintain this greater flow rate of steam (for greater heating effect), an error must develop between PV and SP. In other words, the process variable (temperature) *must deviate from setpoint* in order for the controller to call for more steam, in order that the process variable does not fall any further than this.

This necessary error between PV and SP is called *proportional-only offset*, sometimes less formally known as *droop*. The amount of droop depends on how severe the load change is, and how aggressive the controller responds (i.e., how much gain it has). The term "droop" is very misleading, as it is possible for the error to develop the other way (i.e., the PV might rise above SP due to a load change!). Imagine the opposite load-change scenario, where the incoming feed temperature suddenly *rises* instead of falls. If the controller was controlling exactly at setpoint before this upset, the final result will be an outlet temperature that settles at some point above setpoint, enough so the controller is able to pinch the steam valve far enough closed to stop any further rise in temperature.

Proportional-only offset also occurs as a result of setpoint changes. The same sort of effect follows an operator's increase of setpoint for the temperature controller on the heat exchanger. After increasing the setpoint, the controller immediately increases the output signal, sending more steam to the heat exchanger. As temperature rises, though, the proportional algorithm causes the output signal to decrease. When the rate of heat energy input by the steam equals the rate of heat energy carried away from the heat exchanger by the heated fluid (a condition of *energy balance*, the temperature stops rising. This new equilibrium temperature will not be at setpoint, assuming

the temperature was holding at setpoint prior to the human operator's setpoint increase. The new equilibrium temperature indeed *cannot* ever achieve any setpoint value higher than the one it did in the past, for if the error ever returned to zero (PV = SP), the steam valve would return to its old position, which would be insufficient to raise the temperature of the heated fluid.

An example of proportional-only control in the context of electronic power supply circuits is the op-amp voltage regulator used to stabilize voltage to a load with power supplied by an unregulated voltage source as shown in Figure 22.13.

Figure 22.13: A steam-to-water shell-and-tube heat exchanger diagram

In this circuit, a zener diode establishes a "reference" voltage (which may be thought of as a "setpoint" for the controlling op-amp to follow). The operational amplifier acts as the proportional-only controller, sensing voltage at the load (PV), and sending a driving output voltage to the base of the power transistor to keep load voltage constant despite changes in the supply voltage or changes in load current (both "loads" in the process-control sense of the word, since they tend to influence voltage at the load circuit without being under the control of the op-amp).

If everything functions properly in this voltage regulator circuit, the load's voltage will be stable over a wide range of supply voltages and load currents. However, the load voltage cannot ever *precisely* equal the reference voltage established by the zener diode, even if the operational amplifier (the "controller") is without defect. The reason for this incapacity to perfectly maintain "setpoint" is the simple fact that in order for the op-amp to generate any output signal at all, there *absolutely must be* a differential voltage between the two input terminals for the amplifier to amplify. Operational amplifiers (ideally) generate an output voltage equal to the enormously high gain value ($A_V$) multiplied by the difference in input voltages (in this case, $V_{ref} - V_{load}$). If $V_{load}$ (the "process variable") were to ever achieve equality with $V_{ref}$ (the "setpoint"), the operational amplifier would experience absolutely no differential input voltage to amplify, and its output signal driving the power transistor would fall to zero. Therefore, there must always exist some *offset* between $V_{load}$ and $V_{ref}$ (between process variable and setpoint) in order to give the amplifier some input voltage to amplify.

The amount of offset is ridiculously small in such a circuit, owing to the enormous gain of the operational amplifier. If the op-amp's transfer function is taken to be $V_{out} = A_V(V_{(+)} - V_{(-)})$, then an equation may be set up predicting the load voltage as a function of reference voltage (assuming

a constant 0.7 volt drop between the base and emitter terminals of the transistor):

$$V_{out} = A_V(V_{(+)} - V_{(-)})$$

$$V_{out} = A_V(V_{ref} - V_{load})$$

$$V_{load} + 0.7 = A_V(V_{ref} - V_{load})$$

$$V_{load} + 0.7 = A_V V_{ref} - A_V V_{load}$$

$$(A_V + 1)V_{load} + 0.7 = A_V V_{ref}$$

$$(A_V + 1)V_{load} = A_V V_{ref} - 0.7$$

$$V_{load} = \frac{A_V V_{ref} - 0.7}{A_V + 1}$$

If, for example, the zener diode produced a reference voltage of 5.00000 volts and the operational amplifier had an open-loop voltage gain of 250,000, the load voltage would settle at a theoretical value of 4.9999772 volts: just barely below the reference voltage value. If the op-amp's open-loop voltage gain were much less – say only 100 – the load voltage would only be 4.94356 volts. This still is quite close to the reference voltage, but definitely not as close as it would be with a greater op-amp gain!

Clearly, then, proportional-only offset can be minimized by increasing the gain of the process controller gain (i.e., decreasing its proportional band). This makes the controller more "aggressive" so it will move the control valve further for any given change in PV or SP. Thus, not as much error needs to develop between PV and SP to move the valve to any new position it needs to go. However, too much controller gain and the control system will become unstable: at best it will exhibit residual oscillations after setpoint and load changes, and at worst it will oscillate out of control altogether. Extremely high gains work well to minimize offset in operational amplifier circuits, only because time delays are negligible between output and input. In applications where large physical processes are being controlled (e.g., furnace temperatures, tank levels, gas pressures, etc.) rather than voltages across small electronic loads, such high controller gains would be met with debilitating oscillations.

If the gain that can be programmed in to the controller is limited, how do can this offset be minimized? One way is for a human operator to periodically place the controller in manual mode and move the control valve just a little bit more so the PV once again reaches SP, then place the controller back into automatic mode. In essence this technique adjusts the "Bias" term of the controller equation. The disadvantage of this technique is rather obvious: it requires frequent human intervention. What's the point of having an automation system that needs periodic human intervention to maintain setpoint?

A more sophisticated method for eliminating proportional-only offset is to add a different control action to the controller: one that takes action based on the amount of error between PV and SP and the amount of time that error has existed. This control mode is called *integral*, or *reset*. This will be the subject of the next section.

## 22.5 Integral (reset) control

Integration is provided either by a mechanism (in the case of a pneumatic controller), an op-amp circuit (in the case of an analog electronic controller), or by a microprocessor executing a digital integration algorithm. The variable being integrated is *error* (the difference between PV and SP). Thus the integral mode of the controller ramps the output either up or down over time, the direction of ramping determined by the sign of the error (PV greater or less than SP), and the rate of ramping determined by the magnitude of the error (how far away PV is from SP).

If proportional action is where the error tells the output how *far* to move, integral action is where the error tells the output how *fast* to move. One might think of integral as being how "impatient" the controller is, with integral action constantly ramping the output as far as it needs to go in order to eliminate error. Once the error is zero (PV = SP), of course, the integral action stops ramping, leaving the controller output (valve position) at its last value just like a stopped car's odometer holds a constant value.

Adding an integral term to the controller equation results in[4]:

$$m = K_p e + \frac{1}{\tau_i} \int e \, dt + b$$

Where,
$m$ = Controller output
$e$ = Error (difference between PV and SP)
$K_p$ = Proportional gain
$\tau_i$ = Integral time constant (minutes)
$t$ = Time
$b$ = Bias

The controller integral term multiplies error by time (for very short segments of time, $dt$) and continuously adds up all those products to contribute to the output signal which then drives the control valve (or other final control element). The integral time constant ($\tau_i$) is a value set by the technician or engineer configuring the controller, proportioning this cumulative action to make it more or less aggressive over time.

To see how this works in a practical sense, let's imagine how a proportional + integral controller would respond to the scenario of a heat exchanger whose inlet temperature suddenly dropped. As was seen with proportional-only control, an inevitable offset occurs between PV and SP with changes in load, because an error *must* develop if the controller is to generate the different output signal value necessary to halt further change in PV. This effect was called *proportional-only offset*.

Once this error develops, though, integral action begins to work. Over time, a larger and larger quantity accumulates in the integral mechanism (or register) of the controller because an error persists over time. That accumulated value adds to the controller's output, driving the steam control valve further and further open. This, of course, adds heat at a faster rate to the heat exchanger, which causes the outlet temperature to rise. As the temperature re-approaches setpoint, the error becomes smaller and thus the integral action proceeds at a slower rate (like a car's odometer ticking by at a slower rate when the car's speed decreases). So long as the PV is below SP (the outlet

[4]The equation for a proportional + integral controller is often written without the bias term ($b$), because the presence of integral action makes it unnecessary.

temperature is still too cool), the controller will continue to integrate upwards, driving the control valve further and further open. Only when the PV rises to exactly meet SP does integral action finally rest, holding the valve at a steady position. Integral action ceaselessly works to eliminate any offset between PV and SP, thus neatly eliminating the offset problem experienced with proportional-only control action.

As with proportional action, there are (unfortunately) two completely opposite ways to specify the degree of integral action offered by a controller. One way is to specify integral action in terms of *minutes* or *minutes per repeat*. A large value of "minutes" for a controller's integral action means a less aggressive integral action over time, just as a large value for proportional band means a less aggressive proportional action. The other way to specify integral action is the inverse: how many *repeats per minute*, equivalent to specifying proportional action in terms of gain (large value means aggressive action). For this reason, the integral term of a PID equation is sometimes written differently:

$$\tau_i = \text{minutes per repeat} \qquad K_i = \text{repeats per minute}$$

$$\frac{1}{\tau_i} \int e\, dt \qquad\qquad K_i \int e\, dt$$

Many modern digital electronic controllers allow the user to select the unit they wish to use for integral action, just as they allow a choice between specifying proportional action as gain or as proportional band.

Integral is a highly effective mode of process control. In fact, some processes respond so well to integral controller action that it is possible to operate the control loop on integral action alone, without proportional. Typically, though, process controllers are designed to operate as proportional-only (P), proportional plus integral (PI).

Just as too much proportional gain will cause a process control system to oscillate, too much integral action (i.e., an integral time constant that is too short) will also cause oscillation. If the integration happens at too fast a rate, the controller's output will "saturate" either high or low before the process variable can make it back to setpoint. Once this happens, the only condition that will "unwind" the accumulated integral quantity is for an error to develop of the opposite sign, and remain that way long enough for a canceling quantity to accumulate. Thus, the PV must cross over the SP, guaranteeing at least another half-cycle of oscillation.

A similar problem called *reset windup* (or *integral windup*) happens when external conditions make it impossible for the controller to hold the process variable equal to setpoint. Imagine what would happen in the heat exchanger system if the steam boiler suddenly stopped producing steam. As outlet temperature dropped, the controller's proportional action would open up the control valve in a futile effort to raise temperature. If and when steam service is restored, proportional action would just move the valve back to its original position as the process variable returned to its original value (before the boiler died). This is how a proportional-only controller would respond to a steam "outage": nice and predictably. If the controller had integral action, however, a much worse condition would result. All the time spent with the outlet temperature below setpoint causes the controller's integral term to "wind up" in a futile attempt to admit more steam to the heat exchanger. This accumulated quantity can only be un-done by the process variable rising above setpoint for an equal error-time product, which means when the steam supply resumes, the temperature will rise well above setpoint until the integral action finally "unwinds" and brings the control valve back to a sane

position again.

Various techniques exist to manage integral windup. Controllers may be built with limits to restrict how far the integral term can accumulate under adverse conditions. In some controllers, integral action may be turned off completely if the error exceeds a certain value. The surest fix for integral windup is human operator intervention, by placing the controller in manual mode. This typically resets the integral accumulator to a value of zero and loads a new value into the bias term of the equation to set the valve position wherever the operator decides. Operators usually wait until the process variable has returned at or near setpoint before releasing the controller into automatic mode again.

While it might appear that operator intervention is again a problem to be avoided (as it was in the case of having to correct for proportional-only offset), it is noteworthy to consider that the conditions leading to integral windup usually occur only during shut-down conditions. It is customary for human operators to run the process manually anyway during a shutdown, and so the switch to manual mode is something they would do anyway and the potential problem of windup often never manifests itself.

## 22.6 Derivative (rate) control

The final facet of PID control is the "D" term, which stands for *derivative*. Simply put, derivative is the expression of a variable's *rate-of-change* with respect to another variable. Finding the derivative of a function (differentiation) is the inverse operation of integration. With integration, the accumulated value of some variable's product with time was calculated. With derivative, the ratio of a variable's change per unit of time is calculated. Whereas integration is fundamentally a multiplicative operation (products), differentiation always involves division (ratios).

A controller with derivative (or *rate*) action looks at how fast the process variable changes per unit of time, and takes action proportional to that rate of change. In contrast to integral (reset) action which represents the "impatience" of the controller, derivative (rate) action represents the "cautious" side of the controller.

If the process variable starts to change at a high rate of speed, the job of derivative action is to move the control valve in such a direction as to counteract this rapid change, and thereby moderate the speed at which the process variable changes.

What this will do is make the controller "cautious" with regard to rapid changes in process variable. If the process variable is headed toward the setpoint value at a rapid rate, the derivative term of the equation will diminish the output signal, thus slowing tempering the control response and slowing the process variable's approach toward setpoint. To use an automotive analogy, it is as if a driver, driving a very heavy vehicle, preemptively applies the brakes to slow the vehicle's approach to an intersection, knowing that the vehicle doesn't "stop on a dime." The heavier the vehicle, the sooner a wise driver will apply the brakes, to avoid "overshoot" beyond the stop sign and into the intersection.

If the controller equation is modified to incorporate differentiation, it will look something like this:

$$m = K_p e + \frac{1}{\tau_i} \int e \, dt + \tau_d \frac{de}{dt} + b$$

Where,

$m$ = Controller output
$e$ = Error (difference between PV and SP)
$K_p$ = Proportional gain
$\tau_i$ = Integral time constant (minutes)
$\tau_d$ = Derivative time constant (minutes)
$t$ = Time
$b$ = Bias

It is also possible to build a controller with proportional and derivative actions, but lacking integral action. These are most commonly used in applications prone to wind-up, and where the elimination of offset is not critical:

$$m = K_p e + \tau_d \frac{de}{dt} + b$$

Many PID controllers offer the option of calculating derivative response based on rates of change for the process variable (PV) only, rather than the error (PV − SP or SP − PV). This avoids huge "spikes" in the output of the controller if ever a human operator makes a sudden change in setpoint[5]. The mathematical expression for such a controller would look like this:

$$m = K_p e + \frac{1}{\tau_i} \int e \, dt + \tau_d \frac{d\mathrm{PV}}{dt} + b$$

It should be mentioned that derivative mode should be used with caution. Since it acts on rates of change, derivative action will "go crazy" if it sees substantial noise in the PV signal. Even small amounts of noise possess extremely large rates of change (defined as percent PV change per minute of time) owing to the relatively high frequency of noise compared to the timescale of physical process changes.

Ziegler and Nichols, the engineers who wrote the ground-breaking paper entitled "Optimum Settings for Automatic Controllers" had these words to say regarding "pre-act" control (page 762 of the November 1942 *Transactions of the A.S.M.E.*):

> The latest control effect made its appearance under the trade name "Pre-Act." On some control applications, the addition of pre-act response made such a remarkable improvement that it appeared to be in embodiment of mythical "anticipatory" controllers. On other applications it appeared to be worse than useless. Only the difficulty of predicting the usefulness and adjustment of this response has kept it from being more widely used.

## 22.7 Summary of PID control terms

PID control can be a confusing concept to understand. Here, a brief summary of each term within PID (P. I, and D) is presented.

[5] It should not be assumed that such spikes are always undesirable. In processes characterized by long lag times, such a response may be quite helpful in overcoming that lag for the purpose of rapidly achieving new setpoint values. Slave (secondary) controllers in cascaded systems – where the controller receives its setpoint signal from the output of another (primary, or master) controller – may similarly benefit from derivative action calculated on error instead of just PV. As usual, the specific needs of the application dictate the ideal controller configuration.

### 22.7.1 Proportional control mode (P)

*Proportional* – sometimes called *gain* or *sensitivity* – is a control action reproducing changes in input as changes in output. Proportional controller action responds to present changes in input by generating immediate and commensurate changes in output. When thinking of "proportional action" (P), think *punctual*: this control action works immediately (never too soon or too late) to match changes in the input signal.

Mathematically defined, proportional action is the ratio of output change to input change. This may be expressed as a quotient of differences, or as a derivative (a rate of change, using calculus notation):

$$\text{Gain value} = \frac{\Delta\text{Output}}{\Delta\text{Input}}$$

$$\text{Gain value} = \frac{d\text{Output}}{d\text{Input}} = \frac{dm}{de}$$

For example, if the PV input of a proportional-only process controller with a gain of 2 suddenly changes ("steps") by 5 percent, and the output will immediately jump by 10 percent ($\Delta\text{Output} = \text{Gain} \times \Delta\text{Input}$). The direction of this output jump in relation to the direction of the input jump depends on whether the controller is configured for direct or reverse action.

A legacy term used to express this same concept is *proportional band*: the mathematical reciprocal of gain. "Proportional band" is defined as the amount of input change necessary to evoke full-scale (100%) output change in a proportional controller. Incidentally, it is always expressed as a percentage, never as fraction or as a decimal:

$$\text{Proportional Band value} = \frac{\Delta\text{Input}}{\Delta\text{Output}}$$

$$\text{Proportional Band value} = \frac{d\text{Input}}{d\text{Output}} = \frac{de}{dm}$$

Using the same example of a proportional controller exhibiting an output "step" of 10% in response to a PV "step" of 5%, the proportional band would be 50%: the reciprocal of its gain ($\frac{1}{2} = 50\%$). Another way of saying this is that a 50% input "step" would be required to change the output of this controller by a full 100%, since its gain is set to a value of 2.

### 22.7.2 Integral control mode (I)

*Integral* – sometimes called *reset* or *floating control* – is a control action causing the output signal to change over time at a rate proportional to the amount of error (the difference between PV and SP values). Integral controller action responds to error accumulated over time, ramping the output signal are far as it needs to go to completely eliminate error. If proportional (P) action tells the output how *far* to go when an error appears, integral (I) action tells the output how *fast* to move when an error appears. If proportional (P) action acts on the *present*, integral (I) action acts on the *past*. Thus, how far the output signal gets driven by integral action depends on the *history* of the error over time: how much error existed, and for how long. When thinking of "integral action"

(I), think *impatience*: this control action drives the output further and further the longer PV fails to match SP.

Mathematically defined, integral action is the ratio of output *velocity* to input error:

$$\text{Integral value (repeats per minute)} = \frac{\text{Output velocity}}{\text{Input error}}$$

$$\text{Integral value (repeats per minute)} = \frac{\frac{dm}{dt}}{e}$$

An alternate way to express integral action is to use the reciprocal unit of "minutes per repeat." If integral action is defined in these terms, the defining equations must be reciprocated:

$$\text{Integral time constant (minutes per repeat)} = \tau_i = \frac{\text{Input error}}{\text{Output velocity}}$$

$$\text{Integral time constant (minutes per repeat)} = \tau_i = \frac{e}{\frac{dm}{dt}}$$

For example, if an error of 5% appears between PV and SP on an integral-only process controller with an integral value of 3 repeats per minute (i.e., an integral time constant of 0.333 minutes per repeat), the output will begin ramping at a rate of 15% per minute ($\frac{dm}{dt}$ = Integral_value $\times$ $e$, or $\frac{dm}{dt} = \frac{e}{\tau_i}$). In many PI and PID controllers, integral response is also multiplied by proportional gain, so the same conditions applied to a PI controller that happened to also have a gain of 2 would result in an output ramping rate of 30% per minute ($\frac{dm}{dt}$ = Gain_value $\times$ Integral_value $\times$ $e$, or $\frac{dm}{dt}$ = Gain_value $\times$ $\frac{e}{\tau_i}$). The direction of this ramping in relation to the direction (sign) of the error depends on whether the controller is configured for direct or reverse action.

### 22.7.3 Derivative control mode (D)

*Derivative* – sometimes called *rate* or *pre-act* – is a control action causing the output signal to be offset by an amount proportional to the rate at which the input is changing. Derivative controller action responds to how quickly the input changes over time, biasing the output signal commensurate with that rate of input change. If proportional (P) action tells the output how *far* to go when an error appears, derivative (D) action tells the output how far to go when the input *ramps*. If proportional (P) action acts on the *present* and integral (I) action acts on the *past*, derivative (D) action acts on the *future*: it effectively "anticipates" overshoot by tempering the output response according to how fast the process variable is rising or falling. When thinking of "derivative action" (D), think *discretion*: this control action is cautious and prudent, working against change.

Mathematically defined, derivative action is the ratio of output offset to input *velocity*:

$$\text{Derivative time constant (minutes)} = \tau_d = \frac{\text{Output offset}}{\text{Input velocity}}$$

$$\text{Derivative time constant (minutes)} = \tau_d = \frac{\Delta\text{Output}}{\frac{de}{dt}}$$

For example, if the PV signal begins to ramp at a rate of 5% per minute on a process controller with a derivative time constant of 4 minutes, the output will immediately become offset by 20% ($\Delta$Output = Derivative_value $\times$ $\frac{de}{dt}$). In most PD and PID controllers, derivative response is also multiplied by proportional gain, so the same conditions applied to a PD controller that happened to also have a gain of 2 would result in an immediate offset of 40% ($\Delta$Output = Gain_value $\times$ Derivative_value $\times$ $\frac{de}{dt}$). The direction (sign) of this offset in relation to the direction of the input ramping depends on whether the controller is configured for direct or reverse action.

## 22.8 P, I, and D responses graphed

A very helpful method for understanding the operation of proportional, integral, and derivative control terms is to analyze their respective responses to the same input conditions over time. This section is divided into subsections showing P, I, and D responses for several different input conditions, in the form of graphs. In each graph, the controller is assumed to be *direct-acting* (i.e., an increase in process variable results in an increase in output).

It should be noted that these graphic illustrations are all qualitative, not quantitative. There is too little information given in each case to plot exact responses. The illustrations of P, I, and D actions focus only on the *shapes* of the responses, not their exact numerical values.

In order to *quantitatively* predict PID controller responses, one would have to know the values of all PID settings, as well as the original starting value of the output before an input change occurred and a time index of when the change(s) occurred.

### 22.8.1 Responses to a single step-change

Figure 22.14 shows the individual PID component responses to a step change in the process variable.

Figure 22.14: PID individual component responses to a step change in the PV

Proportional action directly mimics the shape of the input change (a step). Integral action ramps at a rate proportional to the magnitude of the input step. Since the input step holds a constant value, the integral action ramps at a constant rate (a constant *slope*). Derivative action interprets the step as an *infinite* rate of change, and so generates a "spike" driving the output to saturation.

When combined into one PID output, the three actions produce the response shown in Figure 22.15.

Figure 22.15: The combined PID controller output response to a step change in PV

### 22.8.2 Responses to a momentary step-and-return

Figure 22.16 shows the individual PID component responses to a pulse change in the process variable.

Figure 22.16: PID individual component responses to a pulse change in the PV

Proportional action directly mimics the shape of the input change (an up-and-down step). Integral action ramps at a rate proportional to the magnitude of the input step, for as long as the PV is unequal to the SP. Once PV = SP again, integral action stops ramping and simply holds the last value[6]. Derivative action interprets both steps as *infinite* rates of change, and so generates a "spike" at the leading and at the trailing edges of the step. Note how the leading (rising) edge causes derivative action to saturate high, while the trailing (falling) edge causes it to saturate low.

When combined into one PID output, the three actions produce the response shown in Figure 22.17.

[6]This is a good example of how integral controller action represents the *history* of the PV − SP error. The continued offset of integral action from its starting point "remembers" the area accumulated under the rectangular "step" between PV and SP. This offset will go away only if a *negative* error appears having the same percent-minute product (area) as the positive error step.

Figure 22.17: The combined PID controller output response to a pulse change in PV

### 22.8.3 Responses to two momentary steps-and-returns

Figure 22.18 shows the individual PID component responses to two pulse changes in the process variable.

Figure 22.18: PID individual component responses to pulse changes in the PV

Proportional action directly mimics the shape of all input changes. Integral action ramps at a rate proportional to the magnitude of the input step, for as long as the PV is unequal to the SP. Once PV = SP again, integral action stops ramping and simply holds the last value. Derivative action interprets each step as an *infinite* rate of change, and so generates a "spike" at the leading and at the trailing edges of each step. Note how a leading (rising) edge causes derivative action to saturate high, while a trailing (falling) edge causes it to saturate low.

When combined into one PID output, the three actions produce the response shown in Figure 22.19.

Figure 22.19: The combined PID controller output response to pulse changes in PV

### 22.8.4 Responses to a ramp-and-hold

Figure 22.20 shows the individual PID component responses to a ramp-and-hold change in the process variable.

Figure 22.20: PID individual component responses to a ramp-and-hold change in the PV

Proportional action directly mimics the ramp-and-hold shape of the input. Integral action ramps slowly at first (when the error is small) but increases ramping rate as error increases. When error stabilizes, integral rate likewise stabilizes. Derivative action offsets the output according to the input's ramping rate.

When combined into one PID output, the three actions produce the response shown in Figure 22.21.

Figure 22.21: The combined PID controller output response to a ramp-and-hold change in PV

### 22.8.5 Responses to an up-and-down ramp

Figure 22.22 shows the individual PID component responses to an up-and-down ramp change in the process variable.

Figure 22.22: PID individual component responses to an up-and-down ramp change in the PV

Proportional action directly mimics the up-and-down ramp shape of the input. Integral action ramps slowly at first (when the error is small) but increases ramping rate as error increases, then ramps slower as error decreases back to zero. Once PV = SP again, integral action stops ramping and simply holds the last value. Derivative action offsets the output according to the input's ramping rate: first positive then negative.

When combined into one PID output, the three actions produce the response shown in Figure 22.23.

Figure 22.23: The combined PID controller output response to an up-and-down ramp change in PV

### 22.8.6 Responses to a sine wave

Figure 22.24 shows the individual PID component responses to a sinusoidal change in the process variable.

Figure 22.24: PID individual component responses to a sinusoidal change in the PV

As always, proportional action directly mimics the shape of the input. The $90^o$ phase shift seen in the integral and derivative responses, compared to the PV wave, is no accident or coincidence. The derivative of a sinusoidal function is *always* a cosine function, which is mathematically identical to a sine function with the angle advanced by $90^o$:

$$\frac{d(\sin x)}{dx} = \cos x = \sin(x + 90^o)$$

Conversely, the integral of a sine function is *always* a negative cosine function[7], which is mathematically identical to a sine function with the angle retarded by $90^o$:

$$\int \sin x \, dx = -\cos x = \sin(x - 90^o)$$

In summary, the derivative operation always adds a positive (leading) phase shift to a sinusoidal input waveform, while the integral operation always adds a negative (lagging) phase shift to a sinusoidal input waveform.

When combined into one PID output, these particular integral and derivative actions mostly cancel, since they happen to be sinusoidal wave components of equal amplitude and opposite phase. Thus, the only way that the final (PID) output differs from proportional-only action in this particular case is the "steps" caused by derivative action responding to the input's sudden rise at the beginning and end of the wave as shown in Figure 22.25[8].

[7] In this example, the constant of integration ($C$) has been omitted to keep things simple. The actual integral is as such: $\int \sin x \, dx = -\cos x + C = \sin(x - 90^o) + C$. This constant value is essential to explaining why the integral response does not immediately "step" like the derivative response does at the beginning of the PV sine wave.

[8] The internal model principle says that steady-state error-free control can only be achieved if the plant or controller explicitly contains a representation of the disturbance signal. This is why proportional controllers cannot track or reject step changes with no steady-state error - they don't contain a $1/s$ term. However, PI and PID controllers can

Figure 22.25: The combined PID controller output response to a sinusoidal change in PV

If the I and D tuning parameters were such that the integral and derivative responses were *not* equal in amplitude, their effects would not completely cancel. Rather, the resultant of P, I, and D actions would be a sine wave having a phase shift somewhere between -90$^o$ and +90$^o$ exclusive, depending on the relative strengths of the P, I, and D actions.

The 90 degree phase shifts associated with the integral and derivative operations are useful to understand when tuning PID controllers. If one is familiar with these phase shift relationships, it is relatively easy to analyze the response of a PID controller to a sinusoidal input (such as when a process oscillates following a sudden load or setpoint change) to determine if the controller's response is dominated by any one of the three actions. This may be helpful in "de-tuning" an over-tuned (overly aggressive) PID controller, if an excess of P, I, or D action may be identified from a phase comparison of PV and output waveforms.

## 22.9 Different PID equations

### 22.9.1 Parallel PID equation

The equation used to describe PID control so far in this chapter is the simplest form, sometimes called the *parallel* equation, because each action (P, I, and D) occurs in separate terms of the equation, with the combined effect being a simple sum:

$$m = K_p e + \frac{1}{\tau_i} \int e \, dt + \tau_d \frac{de}{dt} + b \qquad \textbf{Parallel PID equation}$$

track or reject step changes without steady-state error because the intergrator provides a $1/s$ term. It also explains why P, PI, and PID controllers fail to track or reject sinusoids without steady-state error - they contain no terms that have the equivalent transform of a sinusoid.

In the parallel equation, each action parameter ($K_p$, $\tau_i$, $\tau_d$) is independent of the others. At first, this may seem to be an advantage, for it means each adjustment made to the controller should only affect one aspect of its action. However, there are times when it is better to have the gain parameter affect all three control actions (P, I, and D).

### 22.9.2 Ideal PID equation

An alternate version of the PID equation exists to provide this very functionality. This version is called the *Ideal* or *ISA* equation:

$$m = K_p \left( e + \frac{1}{\tau_i} \int e \, dt + \tau_d \frac{de}{dt} \right) + b \qquad \textbf{Ideal} \text{ or } \textbf{ISA PID equation}$$

Here, the gain constant ($K_p$) is distributed to all terms within the parentheses, equally affecting all three control actions. Increasing $K_p$ in this style of PID controller makes the P, the I, *and* the D actions equally more aggressive.

### 22.9.3 Series PID equation

A third version, with origins in the peculiarities of pneumatic and analog electronic circuits, is called the *Series* or *Interacting* equation:

$$m = K_p \left( e + \frac{1}{\tau_i} \int e \, dt \right) \left( 1 + \tau_d \frac{de}{dt} \right) + b \qquad \textbf{Series} \text{ or } \textbf{Interacting PID equation}$$

Here, the gain constant ($K_p$) affects all three actions (P, I, and D) just as with the "ideal" equation. The difference, though, is the fact that both the integral and derivative constants have an effect on proportional action as well! That is to say, adjusting either $\tau_i$ or $\tau_d$ does not merely adjust those actions, but also influences the aggressiveness of proportional action.

This "interacting" equation was an artifact of certain pneumatic and electronic controller designs. Back when these were the dominant technologies, and PID controllers were modularly designed such that integral and derivative actions were separate hardware modules included in a controller at additional cost beyond proportional-only action, the easiest way to implement the integral and derivative actions was in a way that just happened to have an interactive effect on controller gain. In other words, this odd equation form was a sort of compromise made for the purpose of simplifying the design of the controller hardware.

Interestingly enough, some digital PID controllers still implement the "interacting" PID equation even though it is no longer a necessary artifact of controller design.

## 22.10 Analog electronic PID controllers

Although analog electronic process controllers are considered a newer technology than pneumatic process controllers, they are actually "more obsolete" than pneumatic controllers. Panel-mounted (inside a control room environment) analog electronic controllers were a great improvement over panel-mounted pneumatic controllers when they were first introduced to industry, but they were superseded by digital controller technology later on. Field-mounted pneumatic controllers were either

replaced by panel-mounted electronic controllers (either analog or digital) or left alone. Applications still exist for field-mounted pneumatic controllers, even now at the beginning of the $21^{st}$ century, but very few applications exist for analog electronic controllers in any location.

Analog electronic controllers enjoy only two advantages over digital electronic controllers: greater reliability and faster response. Now that digital industrial electronics has reached a very high level of reliability, the first advantage is academic, leaving only the second advantage for practical consideration. The advantage of faster speed may be fruitful in applications such as motion control, but for most industrial processes even the slowest digital controller is fast enough[9]. Furthermore, the numerous advantages offered by digital technology (data recording, networking capability, self-diagnostics, flexible configuration, function blocks for implementing different control strategies) severely weaken the relative importance of reliability and speed.

Most analog electronic PID controllers utilized operational amplifiers in their designs. It is relatively easy to construct circuits performing amplification (gain), integration, differentiation, summation, and other useful control functions with just a few op-amps, resistors, and capacitors.

The schematic diagram in Figure 22.26 shows a full PID controller implemented using eight operational amplifiers, designed to input and output voltage signals representing PV, SP, and Output:

[9]The real problem with digital controller speed is that the time delay between successive "scans" translates into dead time for the control loop. Dead time is the single greatest impediment to feedback control.

Figure 22.26: An analog PID controller

This controller implements the *parallel*, or *independent* PID algorithm, since each tuning adjustment (P, I, and D) act independently of each other:

$$m = K_p e + \frac{1}{\tau_i} \int e \, dt + \tau_d \frac{de}{dt} + b \qquad \textbf{Parallel PID equation}$$

It is possible to construct an analog PID controller with fewer operational amplifiers. An example is shown in Figure 22.27.

Figure 22.27: An analog PID controller made with fewer amplifiers

As can be seen, a *single* operational amplifier does all the work of calculating proportional, integral, *and* derivative responses. The first three amplifiers do nothing but buffer the input signals and calculate error (PV−SP, or SP−PV, depending on the direction of action).

This controller design happens to implement the *series* or *interacting* PID equation. Adjusting either the derivative or integral potentiometers also has an effect on the proportional (gain) value, and adjusting the gain of course has an effect on all terms of the PID equation:

$$m = K_p \left( e + \frac{1}{\tau_i} \int e \, dt \right) \left( 1 + \tau_d \frac{de}{dt} \right) + b$$

**Series** or **Interacting PID equation**

It should be apparent now why analog controllers tend to implement the series equation instead of the parallel or ideal PID equations: they are simpler and less expensive to build that way.

One popular analog electronic controller was the Foxboro model 62H, shown in the following photographs in Figure 22.28. Like the model 130 pneumatic controller, this electronic controller was designed to fit into a rack next to several other controllers. Tuning parameters were adjustable by moving potentiometer knobs under a side-panel accessible by partially removing the controller from its rack:

(a) Controller drawer (b) Tuning knobs

Figure 22.28: Foxboro 62H analog PID controller

The Fisher corporation manufactured a series of analog electronic controllers called the $AC^2$, which were similar in construction to the Foxboro model 62H, but very narrow in width so that many could be fit into a compact panel space.

One of the most technologically advanced analog electronic products manufactured for industrial control applications was the Foxboro SPEC 200 system[10]. Although the SPEC 200 system used panel-mounted indicators, recorders, and other interface components resembling panel-mounted control systems, the actual control functions were implemented in a separate equipment rack which Foxboro called a *nest*[11]. Printed circuit boards plugged into each "nest" provided all the control functions (PID controllers, alarm units, integrators, signal selectors, etc.) necessary, with analog signal wires connecting the various functions together with panel-mounted displays and with field instruments to form a working system.

Analog field instrument signals (4-20 mA, or in some cases 10-50 mA) were all converted to a 0-10 VDC range for signal processing within the SPEC 200 nest. Operational amplifiers (mostly the model LM301) formed the "building blocks" of the control functions, with a +/- 15 VDC power supply providing DC power for everything to operate.

An an example of SPEC 200 technology, the following photographs in Figure 22.29 show a model 2AX+A4 proportional-integral (PI) controller card, inserted into a metal frame (called a "module" by Foxboro). This module was designed to fit into a slot in a SPEC 200 "nest" where it would reside alongside many other similar cards, each card performing its own control function:

[10] Although the SPEC 200 system – like most analog electronic control systems – is considered obsolete, working installations may still be found. A report published by the Electric Power Research Institute (see References at the end of this chapter) in 2001 documents a SPEC 200 analog control system installed in a nuclear power plant in the United States as recently as 1992, and another as recently as 2001 in a Korean nuclear power plant.

[11] Foxboro provided the option of a self-contained, panel-mounted SPEC 200 controller unit with all electronics contained in a single module, but the split architecture of the display/nest areas was preferred for large installations where many dozens of loops (especially cascade, feedforward, ratio, and other multi-component control strategies) would be serviced by the same system.

(a) Controller drawer (b) Tuning and alarm knobs

Figure 22.29: SPEC 200 analog PID controller

Tuning and alarm adjustments may be seen in the right-hand photograph. This particular controller is set to a proportional band value of approximately 170, and an integral time constant of just over 0.01 minutes per repeat. A two-position rotary switch near the bottom of the card selected either reverse ("Dec") or direct ("Inc") control action.

The array of copper pins at the top of the module form the male half of a cable connector, providing connection between the control card and the front-panel instrument accessible to operations personnel. Since the tuning controls appear on the face of this controller card (making it a "card tuned" controller), they were not accessible to operators but rather only to the technical personnel with access to the nest area. Other versions of controller cards ("control station tuned") had blank places where the P and I potentiometer adjustments appear on this model, with tuning adjustments provided on the panel-mounted instrument displays for easier access to operators.

The set of ten screw terminals at the bottom of the module provided connection points for the input and output voltage signals. These are shown in Figure 22.30 The following list gives the general descriptions of each terminal pair, with the descriptions for this particular PI controller written in *italic* type.

Figure 22.30: SPEC 200 analog controller I/O terminals

- Terminals (1+) and (1-): Input signal #1 (*Process variable input*)
- Terminals (2+) and (2-): Output signal #1 (*Manipulated variable output*)
- Terminals (3+) and (3-): Input #2, Output #4, or Option #1 (*Remote setpoint*)
- Terminals (4+) and (4-): Input #3, Output #3, or Option #2 (*Optional alarm*)
- Terminals (5+) and (5-): Input #4, Output #2, or Option #3 (*Optional 24 VAC*)

A photograph of the printed circuit board (card) removed from the metal module clearly shows the analog electronic components:

Foxboro went to great lengths in their design process to maximize reliability of the SPEC 200 system, already an inherently reliable technology by virtue of its simple, analog nature. As a result, the reliability of SPEC 200 control systems is the stuff of legend[12].

[12] Tony Kuphaldt once encountered an engineer who joked that the number "200" in "SPEC 200" represented the number of years the system was designed to continuously operate. At another facility, he encountered instrument technicians who were a bit afraid of a SPEC 200 system running a section of their plant: the system had *never suffered a failure of any kind* since it was installed decades ago, and as a result no one in the shop had any experience troubleshooting it. As it turns out, the entire facility was eventually shut down and sold, with the SPEC 200 nest running faithfully until the day its power was turned off!

## 22.11 Digital PID controllers

The vast majority of PID controllers in service today are digital in nature. Microprocessors executing PID algorithms provide many advantages over any form of analog PID control (pneumatic or electronic), not the least of which being the ability to network with personal computer workstations and other controllers over wired or wireless (radio) networks.

### 22.11.1 Digital PID algorithms

As opposed to the continuous time domain PID algorithms given in Section 22.9, digital PID algorithms are quantized in both time (i.e., sampled) and amplitude. Discretization of the position PID algorithm using a backward difference method with sampling interval $T_s$ yields the absolute value of the control effort at time step $n$:

$$u_n = K_c \left[ e_n + \frac{T_s}{\tau_i} \sum_{k=1}^{n} e_k + \frac{\tau_d}{T_s} \left( e_n - e_{n-1} \right) \right]$$

The velocity PID algorithm is found by taking the position algorithm, delaying it one time step, and subtracting it from the position algorithm. This yields the relative (i.e., incremental) value of the control effort at time step $n$:

$$\Delta u_n = K_c \left[ \left( e_n - e_{n-1} \right) + \frac{T_s}{\tau_i} e_n + \frac{\tau_d}{T_s} \left( e_n - 2e_{n-1} + e_{n-2} \right) \right]$$

The current value of the change in control output $\Delta u_n$ is then added to the previous value of the control output $u_{n-1}$ to find the current control output value $u_n$, i.e., $u_n = u_{n-1} + \Delta u_n$.

From a human operator's point of view, the position algorithm and the velocity algorithm are identical with one exception: how each controller reacts to a sudden change in gain ($K_c$). To understand this difference, perform a "thought experiment" by imagining a condition of constant error between PV and SP. Suppose the controller is operating in automatic mode, with a setpoint of 60% and a (steady) process variable value of 57%. A constant error like this might exist for a proportional-only controller trying to control a Type 0 system.

How will this controller react if the gain is suddenly increased in value while operating in automatic mode? If the controller executes the position algorithm, the result of a sudden gain change will be a sudden change in its output value, since output is a direct function of error and gain. However, if the controller executes the velocity algorithm, the result of a sudden gain change will be no change to the output at all, so long as the error *remains constant*. Only when the error begins to change will there be any noticeable difference in the controller's behavior compared to how it acted before the gain change. This is because the velocity algorithm is a function of gain and *change in error*, not error directly.

Comparing the two responses, the velocity algorithm's response to changes in gain is regarded as "better-mannered" than the position algorithm's response to changes in gain. When tuning a controller, it would be better if the controller's output doesn't suddenly jump in response to simple gain changes[13], and so the velocity algorithm is generally preferred. If the gain of the algorithm is

[13] It should be noted that this is precisely what happens when changing the gain in a pneumatic or an analog electronic controller, since all analog PID controllers implement the "position" equation. Although the choice between "position" and "velocity" algorithms in a digital controller is arbitrary, it is *much* easier to build an analog mechanism or circuit implementing the position algorithm than it is to build an analog "velocity" controller.

allowed to be set by another process variable[14], the need for "stable" gain-change behavior becomes even more important.

### 22.11.2 Stand-alone digital controllers

An example of a popular panel-mounted digital controller is the Siemens model 353 (formerly the Moore Products model 353), shown in Figure 22.31.

Figure 22.31: The Siemens model 353 single-loop PID controller

This particular controller, like many high-end digital controllers and larger digital control systems, is programmed in a function block language. Each function block in the controller is a software subroutine performing a specific function on input signals, generating at least one output signal. Each function block has a set of configuration parameters telling it how to behave. For example, the PID function block in a digital controller would have parameters specifying direct or reverse action, gain ($K_p$), integral time constant ($\tau_i$), derivative time constant ($\tau_d$), output limits, etc.

Even the "stock" configuration for simple, single-loop PID control is a collection of function blocks linked together as seen in Figure 22.32.

[14]This is called an *adaptive gain* or *gain scheduled* control system.

Figure 22.32: The Siemens model 353 single-loop PID controller default programming

The beauty of function block programming is that the same blocks may be easily re-linked to implement custom control strategies. Take for instance the following function block program written for a Siemens model 353 controller to provide a pulse-width-modulation (PWM, or time-proportioned) output signal instead of the customary 4-20 mA DC analog output signal. The application is for an electric oven temperature control system, where the oven's heating element could only be turned on and off fully rather than continuously varied as shown in Figure 22.33.

Figure 22.33: The Siemens model 353 single-loop PID controller PWM programming

In order to specify links between function blocks, each of the used lettered block inputs is mapped to the output channel of another block. In the case of the time-proportioned function block program, for example, the "P" (process variable) input of the PID function block is set to get its signal from the "01" output channel of the AIN1 (analog input 1) function block. The "TV" (tracking value) input of the SETPT (setpoint) function block is also set to the "01" output channel of the AIN1 function block, so that the setpoint value generator has access to the process variable value in order to implement setpoint tracking. Any function block output may drive an unlimited number of function block inputs (fan-out), but each function block input may receive a signal from *only one* function block output. This is a rule followed within all function block languages to prevent multiple block output signals from conflicting (attempting to insert different signal values into the same input).

In the Siemens controllers, function block programming may be done by entering configuration data using the front-panel keypad, or by using graphical software running on a personal computer networked with the controller.

For applications not requiring so much capability, and/or requiring a smaller form factor, other panel-mounted digital controllers exist. The Honeywell model UDC3000 seen in Figure 22.34 is a popular example of a 1/4 DIN size(96 mm $\times$ 96 mm) digital controller.

Figure 22.34: The Honeywell UDC3000 single-loop PID controller

Even smaller panel-mounted controllers are produced by a wide array of manufacturers for applications requiring minimum functionality: 1/8 DIN (96 mm × 48 mm), 1/16 DIN (48 mm × 48 mm), and even 1/32 DIN (48 mm × 24 mm) sizes are available.

One of the advantageous capabilities of modern stand-alone controllers is the ability to exchange data over digital networks. This provides operations and maintenance personnel alike the ability to remotely monitor and even control (adjust setpoints, switch modes, change tuning parameters, etc.) the process controller from a computer workstation. The Siemens model 353 controller (with appropriate options) has the ability to digitally network over Ethernet, a very common and robust digital network standard. The photographs in Figure 22.35 show three such controllers connected to a network through a common 4-port Ethernet "hub" device.

(a) Front of panel (b) Interior of panel

Figure 22.35: Siemens model 353 controllers networked over Ethernet

Special software (in this case, Siemens *Procidia*) running on a computer workstation connected to the same Ethernet network acquires data from and sends data to the networked controllers. Figure 22.36 shows a screenshot of the Procidia user interface allowing complete control over the function of the process controllers.

Figure 22.36: Procidia PID controller user interface software by Siemens

### 22.11.3 Direct digital controllers (DDC)

A microprocessor operating at sufficient clock speed is able to execute more than one PID control algorithm for a process loop, by "time-sharing" its calculating power: devoting slices of time to the evaluation of each PID equation in rapid succession. This not only makes multiple-loop digital control possible for a single microprocessor, but also makes it very attractive given the microprocessor's natural ability to manage data archival, transfer, and networking. A single computer is able to execute PID control for multiple loops, and also make that loop control data accessible between loops (for purposes of cascade, ratio, feedforward, and other control strategies) and accessible on networks for human operators and technicians to easily access.

Such *direct digital* control (DDC) has been applied with great success to the problem of building automation, where temperature and humidity controls for large structures benefit from large-scale data integration. When the term DDC is brought up, it almost invariably refers to commercial building controls[15]. Figure 22.37 shows a Siemens APOGEE building automation system with

[15] Another term used in this context is building automation system (BAS)

multiple I/O (input/output) cards providing interface between analog instrument signals and the microprocessor's digital functions:

Figure 22.37: An APOGEE DDC system

A close-up view of the processor in Figure 22.38 shows the device that handles all of the mathematical calculations for the PID control.

Figure 22.38: An APOGEE DDC system controller (bottom)

Other than a few LEDs, there is no visual indication in this panel of what the system is doing at any particular time. Operators, engineers, and technicians alike must use software running on a networked personal computer to access data in this control system.

### 22.11.4 Programmable logic controllers (PLC)

A more common application in industrial control is to use a programmable logic controller (PLC) as a PID controller for multiple loops. PLCs were originally invented for on/off (discrete) process control functions, but have subsequently grown in speed and capability to handle analog PID control functions as well. A PLC is often a less expensive option for PID control than a stand-alone controller, which explains the prevalence of this control philosophy in many industrial environments.

Figure 22.39 shows an Allen-Bradley (Rockwell) ControlLogix PLC used to control the operation of a gas turbine engine. The PLC is in the upper-left corner of the enclosure, with the rest of the enclosure devoted to terminal blocks and accessory components.

Figure 22.39: A ControlLogix PLC system

A strong advantage of using PLCs for analog loop control is the ability to easily integrate discrete controls with the analog controls. It is quite easy, for example, to coordinate the sequential start-up and shut-down functions necessary for intermittent operation with the analog PID controls necessary for continuous operation, all within one programmable logic controller. It should be noted, however, that many early PLC implementations of PID algorithms were crude at best, lacking the finesse of stand-alone PID controllers. Even some modern PLC analog functions are mediocre[16], but they continue to improve rapidly in newer versions of PLCs called programmable automation controllers (PACs).

[16]An aspect common to many PLC implementations of PID control is the use of the "parallel" PID algorithm instead of the superior "ISA" or "non-interacting" algorithm. The choice of algorithm may have a profound effect on tuning, and on tuning procedures, especially when tuning parameters must be readjusted to accommodate changes in transmitter range.

### 22.11.5 Supervisory Control And Data Acquisition (SCADA)

A similar control system architecture to Direct Digital Control (DDC) – assigning a single microprocessor to the task of managing multiple control functions, with digital communication between the microprocessor units – is used for the management of systems which are by their very nature spread over wide geographical regions. Such systems are generally referred to as *SCADA*, which is an acronym standing for *Supervisory Control And Data Acquisition*.

A typical SCADA system is shown in Figure 22.40 and consists of multiple *Remote Terminal Unit* (RTU) devices connected to process transmitters and final control elements, implementing basic control functions such as motor start/stop and PID loop control. These RTU devices communicate digitally to a *Master Terminal Unit* (MTU) device at a central location where human operators may monitor the process and issue commands.

Figure 22.40: A SCADA system diagram

A photograph of an RTU "rack" operating at a large electric power substation is shown here:

Figure 22.41: An RTU rack at an electrical substation

Some RTU hardware, such as the substation monitoring system shown above, is custom-manufactured for the application. Other RTU hardware is more general in purpose, such as the ROC 800 manufactured by Fisher, intended for the monitoring and control of natural gas and oil production wells, but applicable to other applications as well. The Fisher ROC units are designed to operate with a minimum of electrical power, so that a single solar panel and battery will be sufficient for year-round operation in remote environments. Radio communication is a standard feature for ROC units, as there are no runs of communications cable linking remote wells separated by dozens of miles.

Standard programmable logic controllers (PLCs) are ideal candidates for use as RTU devices. Modern PLCs have all the I/O, networking, and control algorithm capability necessary to function as remote terminal units. Commercially available Human-Machine Interface (HMI) software allowing personal computers to display PLC variable values potentially turns every PC into a Master Terminal Unit (MTU) where operators can view process variables, change setpoints, and issue other commands for controlling the process.

Figure 22.42 shows a photograph of such HMI software used to monitor a SCADA system for a set of natural gas compressors.

Figure 22.42: An HMI screen for a plant with multiple gas compressors

Figure 22.43 shows another photograph of a similar system used to monitor and control drinking water reservoirs for a city is shown here:

Figure 22.43: An HMI screen for a city water supply system

### 22.11.6 Distributed Control Systems (DCS)

A radical concept appeared in the world of industrial control in the mid-1970s: the notion of *distributed* digital control. Direct digital control during that era suffered a substantial problem: the potential for catastrophic failure if the single digital computer executing *multiple* PID control functions were to ever halt. Digital control brings many advantages, but it isn't worth the risk if the entire operation will shut down (or catastrophically fail!) following a hardware or software failure within that one computer.

Distributed control directly addressed this concern by having multiple control computers – each one responsible for only a handful of PID loops – networked together to share information with each other and with operator display consoles. This way, there would be less concentration of liability as there would be with a single-computer DDC system. Furthermore, distributed control introduced the concept of *redundancy* to industrial control systems: where digital signal acquisition and processing hardware units were equipped with "spare" units designed to automatically take over all critical functions in the event of a primary failure.

Figure 22.44 is an illustration that shows a typical distributed control system (DCS) architecture.

Figure 22.44: An illustration of a DCS system

Each "rack" contains a microprocessor to implement all necessary control functions, with individual I/O (input/output) "cards" for converting analog field instrument signals into digital format, and visa-versa. To address the possibility of rack and network failure, distributed control systems are often built with multiple levels of *redundancy*: "spare" controllers and network cables designed to take over in the event of the other's failure. Some systems provide redundancy even at the I/O level: with primary and secondary input and output cards servicing the same field instruments, so that if even a single card fails, another (identical) card will be on-line to take over the lost I/O functions.

If there ever was a total failure in one of the "control racks" where the redundancy proved insufficient for the fault(s), the only PID loops faulted will be those resident in that rack, not any of the other loops throughout the system. Likewise, if ever a network cable becomes severed or otherwise faulted, only the information flow between those two points will suffer; the rest of the system will continue to communicate data normally. Thus, one of the "hallmark" features of a DCS is its tolerance to serious faults: even in the event of severe hardware or software faults, the impact to process control is minimized by design.

An additional advantage gained by the distribution of control functions to multiple controller racks is the reduction of field instrument wiring. Instead of requiring twisted-pair signal wiring run all the way from field instruments to a central control room, the controller racks could be located in rooms or enclosures much closer to the process areas, with nothing more than a network cable connecting these rooms/enclosures to the central control room where operators would monitor the process. Not only was this a cost savings (far less wire and conduit!), but it also resulted in less maintenance replacing old wiring due to normal degradation or incidental damage.

One of the very first distributed control systems in the world was the Honeywell TDC2000 system[17], introduced in 1975. By today's standards, the technology was crude[18], but the concept was revolutionary.

Each rack (called a "box" by Honeywell) consisted of an aluminum frame holding several large printed circuit boards with card-edge connectors. A "basic controller" box appears in Figure 22.45a. Figure 22.45b shows the termination board where the field wiring (4-20 mA) connections were made. A thick cable connected each termination board to its respective controller box.

[17] To be fair, the Yokogawa Electric Corporation of Japan introduced their CENTUM distributed control system the same year as Honeywell.

[18] Just to give some perspective, the original TDC2000 system used whole-board processors rather than microprocessor chips, and magnetic core memory rather than static or dynamic RAM circuits! Communication between controller nodes and operator stations was handled by thick coaxial cables, implementing a master/slave protocol with a separate device (a "Hiway Traffic Director" or HTD) coordinating all communications between nodes. Like Bob Metcalfe's original version of Ethernet, these coaxial cables were terminated at their end-points by termination resistors, with coaxial "tee" connectors providing branch points for multiple nodes to connect along the network.

(a) DCS Controller (b) Moore IPT

Figure 22.45: A legacy DCS rack

Controller redundancy in the TDC2000 DCS took the form of a "spare" controller box serving as a backup for up to eight other controller boxes. Thick cables routed all analog signals to this spare controller, so that it would have access to them in the event it needed to take over for a failed controller. The spare controller would become active on the event of *any* fault in any of the (up to eight) other controllers, including failures in the I/O cards. Thus, this redundancy system provided for processor failures as well as I/O failures. All TDC2000 controllers communicated digitally by means of a dual coaxial cable network known as the "Data Hiway." The dual cables provided redundancy in network communications.

A typical TDC2000 operator workstation appears in Figure 22.46

Figure 22.46: A legacy DCS operator workstation

Over the years following its 1975 introduction, the Honeywell system grew in sophistication with faster networks (the "Local Control Network" or LCN), more capable controller racks (the "Process

Manager" or PM series), and better operator workstations. Many of these improvements were incremental, consisting of add-on components that could work with existing TDC2000 components so that the entire system need not be replaced to accept the new upgrades.

Other control equipment manufacturers responded to the DCS revolution started by Honeywell and Yokogawa by offering their own distributed control systems. The Bailey Network 90 (Net90) DCS, Bailey Infi90 DCS, and the Fisher Provox systems are examples. Foxboro, already an established leader in the control system field with their SPEC 200 analog system, first augmented the SPEC 200 with digital capabilities (the VIDEOSPEC workstation consoles, FOX I/A computer, INTERSPEC and FOXNET data networks), then developed an entirely digital distributed control system, the SPECTRUM.

Some modern distributed control systems include:

- ABB *800xA*
- Emerson *DeltaV* and *Ovation*
- Foxboro/Invensys *I/A*
- Honeywell *Experion PKS*
- Yokogawa *CENTUM VP* and *CENTUM CS*

For a visual comparison with the Honeywell TDC2000 DCS, Figure 22.47 contains a photograph of an Emerson DeltaV DCS rack, with processor and multiple I/O modules.

Figure 22.47: A modern DCS rack

Many modern distributed control systems such as the Emerson DeltaV use regular personal computers rather than proprietary hardware as operator workstations. This cost-saving measure leverages existing computer and display technologies without sacrificing control-level reliability (since the control hardware and software is still industrial-grade):

Figure 22.48: A modern DCS operator workstation

As previously mentioned in the Direct Digital Control (DDC) subsection, programmable logic controllers (PLCs) are becoming more and more popular as PID control platforms due to their ever-expanding speed, functionality, and relatively low cost. It is now possible with modern PLC hardware and networking capabilities to build a truly distributed control system with individual PLCs as the processing nodes, so that a failure in one PLC does not interrupt PID control throughout the system. Such a system may be purchased at a fraction of the up-front cost of a fully-fledged DCS.

However, what is currently lacking in the PLC world is the same level of hardware and software integration necessary to build a functional distributed control system that comes as ready-to-use as a system pre-built by a DCS manufacturer. In other words, if an enterprise chooses to build their own DCS using programmable logic controllers, they must be prepared to do a *lot* of programming work in order to emulate the same level of functionality and power as a pre-engineered DCS[19]. Any engineer or technician who has experienced the power of a modern DCS – with its self-diagnostic, "smart" instrument management, event auditing, advanced control strategy, redundancy, data collection and analysis, and alarm management capabilities – realizes that these features are not luxuries, but rather necessities for managing mission-critical process controls on a large scale. However, as PLC systems evolve into PAC systems with improvements in both programming and user interface software, their capabilities continue to expand and may make them competitive with DCS systems. Also, their lower cost continue to make PLCs preferred for smaller system deployment, or for systems that are oriented toward machine or motion control. Engineers should weigh the advantages and disadvantages of PAC and DCS platforms before making a selection.

[19] In one major industrial manufacturing facility (which shall remain nameless), a PLC vendor promised the same technical capability as a full DCS at approximately one-tenth the installed cost. Several years and several tens of thousands of man-hours later, the sad realization was this "bargain" did not live up to its promise, and the decision was made to remove the PLCs and go with a complete DCS from another manufacturer. *Caveat emptor!*

### 22.11.7 Fieldbus control

The DCS revolution started in the mid-1970s was fundamentally a moving of control system "intelligence" from a centralized location to distributed locations. Rather than have a single computer (or a panel full of single-loop controllers) located in a central control room implement PID control for a multitude of process loops, many (smaller) computers located closer to the process areas would handle the PID and other control functions, with network cables shuttling data between those distributed locations and the central control room.

Beginning in the early 1990's, the next logical step in this evolution of control architecture saw the relocation of control "intelligence" to the field instruments themselves. In other words, the new idea was to equip individual transmitters and control valve positioners with the necessary computational power to implement PID control all on their own, using digital networks to carry process data between the field instruments and any location desired. This is the fundamental concept of *fieldbus*.

"Fieldbus" as a technical term has multiple definitions. Many manufacturers use the word "fieldbus" to describe any digital network used to transport data to and from field instruments. In this subsection, the word "fieldbus" is used to describe a design philosophy where field instruments possess all the necessary "intelligence" to control the process, with no need for separate centralized (or even distributed) control hardware. *FOUNDATION Fieldbus* is the first standard to embody this fully-distributed control concept, the technical details of this open standard maintained and promoted by the *Fieldbus Foundation*. The aim of this Foundation is to establish an open, technician standard for *any* manufacturer to follow in the design of their fieldbus instruments. This means a FOUNDATION Fieldbus (FF) transmitter manufactured by Smar will work seamlessly with a FF control valve positioner manufactured by Fisher, communicating effortlessly with a FF-aware host system manufactured by ABB, and so on. This may be thought of in terms of being the digital equivalent of the 3-15 psig pneumatic signal standard or the 4-20 mA analog electronic signal standard: so long as all instruments "talk" according to the same standard, brands and models may be freely interchanged to build any control system desired.

To illustrate the general fieldbus concept, consider the flow control system shown in Figure 22.49.

Figure 22.49: A modern DCS operator workstation

Here, a fieldbus *coupling device* provides a convenient junction point for cables coming from the transmitter, valve positioner, and host system. FOUNDATION Fieldbus devices both receive DC power and communicate digitally over the same twisted-pair cables. In this case, the host system provides DC power for the transmitter and positioner to function, while communication of process data occurs primarily between the transmitter and positioner (with little necessary involvement of the host system[20]).

Just like distributed control systems, FOUNDATION Fieldbus instruments are programmed using a function block language. In this case, there must be an analog input (for the transmitter's measurement), a PID function block, and an analog output (for the valve positioner) to make a complete flow control system, as shown in Figure 22.50.

Figure 22.50: Function block programming example for a FOUNDATION Fieldbus control function

[20]Although it is customary for the host system to be configured as the *Link Active Scheduler* (LAS) device to schedule and coordinate all fieldbus device communications, this is not absolutely necessary. Any suitable field instrument may also serve as the LAS, which means a host system is not even necessary except to provide DC power to the instruments, and serve as a point of interface for human operators, engineers, and technicians.

The analog input (AI) block must reside in the transmitter, and the analog output (AO) block must reside in the valve positioner, since those blocks necessarily relate to the measured and controlled variables, respectively. However, the PID block may reside in *either* field device as shown in Figure 22.51.

Figure 22.51: The programming block for a FF PID control function may reside in either the transmitter or the positioner

Practical reasons do exist for choosing one location of the PID function block over the other, most notably the difference in communication loading between the two options[21]. However, there is no *conceptual* limitation to the location of the PID function block. In a fieldbus control system where the control "intelligence" is distributed all the way to the field instrument devices themselves, there are no limits to system flexibility.

## 22.12 Practical PID controller features

In order for any PID controller to be practical, it must be able to do more than just implement the PID equation. This section identifies and explains some of the basic features found on most (but not all!) modern PID controllers:

- Manual versus Automatic mode
- Output tracking
- Setpoint tracking
- Alarming
- PV characterization and damping
- Setpoint limits
- Output limits
- PID tuning security

### 22.12.1 Manual and automatic modes

When a controller continually calculates output values based on PV and SP values over time, it is said to be operating in *automatic* mode. This mode, of course, is what is necessary to regulate any process. There are times, however, when it is desirable to allow a human operator to manually "override" the automatic action of the PID controller. Applicable instances include process start-up and shut-down events, emergencies, and maintenance procedures. A controller that is being "overridden" by a human being is said to be in *manual mode*.

A very common application of manual mode is during maintenance of the sensing element or transmitter. If an instrument technician needs to disconnect a process transmitter for calibration or replacement, the controller receiving that transmitter's signal cannot be left in automatic mode. If it is, then the controller may[22] take sudden corrective action the moment the transmitter's signal goes dead. If the controller is first placed in manual mode before the technician disconnects the transmitter, however, the controller will ignore any changes in the PV signal, letting its output signal be adjusted at will by the human operator. If there is another indicator of the same process variable as the one formerly reported by the disconnected transmitter, the human operator may elect to read that other indicator and play the part of a PID controller, manually adjusting the final

[21] With the PID function block programmed in the flow transmitter, there will be twice as many scheduled communication events per macrocycle than if the function block is programmed into the valve positioner. This is evident by the number of signal lines connecting circled block(s) to circled block(s) in the above illustration.

[22] The only reason to say "may" instead of "will" is because some modern digital controllers are designed to automatically switch to manual-mode operation in the event of a sensor or transmitter signal loss. Any controller not "smart" enough to shed its operating mode to manual in the event of PV signal loss will react dramatically when that PV signal dies, and this is not a good thing for an operating loop!

control element to maintain the alternate indicator at setpoint while the technician completes the transmitter's maintenance.

An extension of this "mode" concept applies to controllers configured to receive a setpoint from another device (called a *remote* or *cascaded* setpoint). In addition to an automatic and a manual mode selection, a third selection called *cascade* exists to switch the controller's setpoint from human operator control to remote (or "cascade") control.

### 22.12.2 Output and setpoint tracking

The provision of manual and automatic operating modes creates a set of potential problems for the PID controller. If, for example, a PID controller is switched from automatic to manual mode by a human operator, and then the output is manually adjusted to some new value, what will the output value do when the controller is switched *back* to automatic mode? In some crude PID controller designs, the result would be an immediate "jump" back to the output value calculated by the PID equation while the controller was in manual. In other words, some controllers never stop evaluating the PID equation – even while in manual mode – and will default to that automatically-calculated output value when the operating mode is switched from manual to automatic.

This can be very frustrating to the human operator, who may wish to use the controller's manual mode as a way to change the controller's bias value. Imagine, for example, that a PD controller (no integral action) is operating in automatic mode at some low output value, which happens to be too low to achieve the desired setpoint. The operator switches the controller to manual mode and then raises the output value, allowing the process variable to approach setpoint. When PV nearly equals SP, the operator switches the controller's mode back to automatic, expecting the PID equation to start working again from this new starting point. In a crude controller, however, the output would jump back to some lower value, right where the PD equation would have placed it for these PV and SP conditions.

A feature designed to overcome this problem – which is so convenient that some consider it an essential feature of any controller with a manual mode – is called *output tracking*. With output tracking, the bias value of the controller shifts every time the controller is placed into manual mode and the output value manually changed. Thus, when the controller is switched from manual mode to automatic mode, the output does *not* immediately jump to some previously-calculated value, but rather "picks up" from the last manually-set value and begins to control from that point as dictated by the PID equation. In other words, output tracking allows a human operator to arbitrarily offset the output of a PID controller by switching to manual mode, adjusting the output value, and then switching back to automatic mode. The output will continue its automatic action from this new starting point instead of the old starting point.

A very important application of output tracking is in the manual correction of integral wind-up (sometimes called *reset windup* or just *windup*). This is what happens to a controller with integral action if for some reason the process variable *cannot* achieve setpoint no matter how far the output signal value is driven by integral action. An example might be on a temperature controller where the source of heat for the process is a steam system. If the steam system shuts down, the temperature controller *cannot* warm the process up to the temperature setpoint value no matter how far open the steam valve is driven by integral action. If the steam system is shut down for too long, the result will be a controller output saturated at maximum value in a futile attempt to warm the process. If and when the steam system starts back up, the controller's saturated output will now send *too much*

*heating steam* to the process, causing the process temperature to overshoot setpoint until integral action drives the output signal back down to some reasonable level. This situation may be averted, however, if the operator switches the temperature controller to manual mode as soon as the steam system shuts down. Even if this preventative step is not taken, the problem of overshoot may be averted upon steam system start-up if the operator uses output tracking by quickly switching the controller into manual mode, adjusting the output down to a reasonable level, and then switching back into automatic mode so that the controller's output value is no longer "wound up" at a high level[23].

A similar feature to output tracking – also designed for the convenience of a human operator switching a PID controller between automatic and manual modes – is called *setpoint tracking*. The purpose of setpoint tracking is to equalize SP and PV while the controller is in manual mode, so that when the controller gets switched back into automatic mode, it will begin its automatic operation with zero error (PV = SP).

This feature is most useful during system start-ups, where the controller may have difficulty controlling the process in automatic mode under unusual conditions. Operators often prefer to run certain control loops in manual mode from the time of initial start-up until such time that the process is near normal operating conditions. At that point, when the operator is content with the stability of the process, the controller is assigned the responsibility of maintaining the process at setpoint. With setpoint tracking present in the controller, the controller's SP value will be held equal to the PV value (whatever that value happens to be) for the entire time the controller is in manual mode. Once the operator decides it is proper to switch the controller into automatic mode, the SP value freezes at that last manual-mode PV value, and the controller will continue to control the PV at that SP value. Of course, the operator is free to adjust the SP value to any new value while the controller is in automatic mode, but this is at the operator's discretion.

Without setpoint tracking, the operator would *have to* make a setpoint adjustment either before or after switching the controller from manual mode to automatic mode, in order to ensure the controller was properly set up to maintain the process variable at the desired value. With setpoint tracking, the setpoint value will default to the process variable value when the controller was last in manual mode, which (it is assumed) will be close enough to the desired value to suffice for continued operation.

Unlike output tracking, for which there is virtually no reason not to have the feature present in a PID controller, there may very well be applications where setpoint tracking is undesirable. For some

[23] A technician once had the misfortune of working on an analog PID controller for a chlorine-based wastewater disinfection system that lacked output tracking. The chlorine sensor on this system would occasionally fail due to sample system plugging by algae in the wastewater. When this happened, the PV signal would fail low (indicating abnormally low levels of chlorine gas dissolved in the wastewater) even though the actual dissolved chlorine gas concentration was adequate. The controller, thinking the PV was well below SP, would ramp the chlorine gas control valve further and further open over time, as integral action attempted to reduce the error between PV and SP. The error never went away, of course, because the chlorine sensor was plugged with algae and simply could not detect the actual chlorine gas concentration in the wastewater. By the time the technician arrived to address the "low chlorine" alarm, the controller output was already wound up to 100%. After cleaning the sensor, and seeing the PV value jump up to some outrageously high level, the controller would take a long time to "wind down" its output because its integral action was very slow. The tech could not use manual mode to "unwind" the output signal, because this controller lacked the feature of output tracking. The "work-around" solution to this problem was to re-tune the integral term of the controller to some really fast time constant, watch the output "wind down" in fast-motion until it reached a reasonable value, then adjust the integral time constant back to its previous value for continued automatic operation.

processes[24], the setpoint value *should* remain fixed at all times, and as such it would be undesirable to have the setpoint value drift around with the process variable value every time the controller was placed into manual mode.

### 22.12.3 Alarm capabilities

A common feature on many instrument systems is the ability to alert personnel to the onset of abnormal process conditions. The general term for this function is *alarm*. Process alarms may be triggered by process switches directly sensing abnormal conditions (e.g., high-temperature switches, low-level alarms, low-flow alarms, etc.), in which case they are called *hard alarms*. A *soft alarm*, by contrast, is an alarm triggered by some continuous measurement (i.e., a signal from a process transmitter rather than a process switch) exceeding a pre-programmed alarm limit value.

Since PID controllers are designed to input continuous process measurements, it makes sense that a controller could be equipped with programmable alarm limit values as well, to provide "soft" alarm capability without adding additional instruments to the loop[25]. Not only is PV alarming easy to implement in most PID controllers, but *deviation* alarming is easy to implement as well. A "deviation alarm" is a soft alarm triggered by excessive deviation (error) between PV and SP. Such an event indicates control problems, since a properly-operating feedback loop should be able to maintain reasonable agreement between PV and SP at all times.

Alarm capabilities find their highest level of refinement in modern distributed control systems (DCS), where the networked digital controllers of a DCS provide convenient access and advanced management of hard and soft alarms alike. Not only can alarms be accessed from virtually any location in a facility in a DCS, but they are usually time-stamped and archived for later analysis, which is an *extremely* important feature for the analysis of emergency events, and the continual improvement of process safety.

### 22.12.4 Output and setpoint limiting

In some process applications, it may not be desirable to allow the controller to automatically manipulate the final control element (control valve, variable-speed motor, heater) over its full 0% - 100% range. In such applications, a useful controller feature is an *output limit*. For example, a PID flow controller may be configured to have a minimum output limit of 5%, so that it is not able to close the control valve any further than the 5% open position in order to maintain "minimum flow" through a pump. The valve may still be fully closed (0% stem position) in manual mode, but just not in automatic mode[26].

Similarly, setpoint values may be internally limited in some PID controllers, such that an operator cannot adjust the setpoint above some limiting value or below some other limiting value. In the event that the process variable *must* be driven outside these limits, the controller may be placed in manual mode and the process "manually" guided to the desired state by an operator.

[24] Boiler steam drum water level control, for example, is a process where the setpoint really should be left at a 50% value at all times, even if there maybe legitimate reasons for occasionally switching the controller into manual mode.

[25] It is very important to note that soft alarms are not a replacement for hard alarms. There is much wisdom in maintaining both hard and soft alarms for a process, so there will be redundant, non-interactive levels of alarming. Hard and soft alarms should complement each other in any critical process.

[26] Some PID controllers limit manual-mode output values as well, so be sure to check the manufacturer's documentation for output limiting on a particular PID controller!

### 22.12.5 Security

There is justifiable reason to prevent certain personnel from having access to certain parameters and configurations on PID controllers. Certainly, operations personnel need access to setpoint adjustments and automatic/manual mode controls, but it may be unwise to grant those same operators unlimited access to PID tuning constants and output limits. Similarly, instrument technicians may require access to a PID controller's tuning parameters, but perhaps should be restricted from editing configuration programs maintained by the engineering staff.

Most digital PID controllers have some form of security access control, allowing for different levels of permission in altering PID controller parameters and configurations. Security may be crude (a hidden switch located on a printed circuit board, which only the maintenance personnel should know about), sophisticated (login names and passwords, like a multi-user computer system), or anything in between, depending on the level of development invested in the feature by the controller's manufacturer.

An interesting solution to the problem of security in the days of analog control systems was the architecture of Foxboro's SPEC 200 analog electronic control system. The controller displays, setpoint adjustments, and auto/manual mode controls were located on the control room panel where anyone could access them. All other adjustments (PID settings, alarm settings, limit settings) could be located in the *nest* area where all the analog circuit control cards resided. Since the "nest" racks could be physically located in a room separate from the control room, personnel access to the nest room served as access security to these system parameters.

At first, the concept of controller parameter security may seen distrustful and perhaps even insulting to those denied access, especially when the denied persons possess the necessary knowledge to understand the functions and consequences of those parameters. It is not uncommon for soft alarm values to be "locked out" from operator access despite the fact that operators understand very well the purpose and functions of these alarms. At some facilities, PID tuning is the exclusive domain of process or I&C engineers, with instrument technicians and operators alike barred from altering PID tuning constants even though some operators and many technicians may well understand PID controller tuning.

When considering security access, there is more to regard than just knowledge or ability. At a fundamental level, security is a task of limiting access commensurate with *responsibility*. In other words, security restrictions exist to exclude those not charged with particular responsibilities. Knowledge and ability are necessary conditions of responsibility (i.e., one cannot reasonably be held responsible for something beyond their knowledge or control), but they are not *sufficient* conditions of responsibility (i.e., knowing how to, and being able to perform a task does not confer responsibility for that task getting completed). An operator may very well understand how and why a soft alarm on a controller works, but the responsibility for altering the alarm value may reside with someone else whose job description it is to ensure the alarm values correspond to plant-wide policies.

# References


[1] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, Design of Feedback Control Systems, 4th ed. New York, NY: Oxford University Press, 2002.

[2] C. L. Phillips and R. D. Harbor, Feedback Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[3] J. R. Rowland, Linear Control Systems: Modeling, Analysis, and Design. New York, NY: John Wiley & Sons, 1986.

[4] R. C. Dorf and R. H. Bishop, Modern Control Systems, 10th ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2005.

[5] F. J. Hale, Introduction to Control System Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[6] A. P. Sage, Linear Systems Control. Champaign, IL: Matrix Publishers, 1978.

[7] M. Driels, Linear Control Systems Engineering. New York, NY: McGraw-Hill, 1995.

[8] W. J. Palm, Control Systems Engineering. New York, NY: John Wiley and Sons, 1986.

[9] S. C. Gupta and L. Hasdorff, Fundamentals of Automatic Control. New York, NY: John Wiley and Sons, 1970.

[10] A. F. D'Souza, Design of Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[11] T. E. Marlin, Process Control, 2nd ed. New York, NY: McGraw-Hill, 2000.

[12] B. A. Ogunnaike and W. H. Ray, Process Dynamics, Modeling, and Control. New York, NY: Oxford University Press, 1994.

[13] A. Tewari, Modern Control Design with MATLAB and Simulink. New York, NY: John Wiley and Sons, 2002.

[14] C. A. Smith and A. Corripio, Principles and Practice of Automatic Process Control, 3rd ed. New York, NY: John Wiley and Sons, 2006.

[15] H. V. Vu and R. S. Esfandiari, Dynamic Systems: Modeling and Analysis. Boston, MA: McGraw-Hill, 1997.

[16] M. Gopal, Control Systems: Principles and Design. Boston, MA: McGraw-Hill, 2008.

[17] FOUNDATION Fieldbus, L454 EN, Samson AG, Frankfurt, Germany, 2000.

[18] "Identification and Description of Instrumentation, Control, Safety, and Information Systems and Components Implemented in Nuclear Power Plants," EPRI, Palo Alto, CA, Tech. Rep. 1001503, 2001.

# Chapter 23

# PID Controller Tuning

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms:
2. Tune PID controllers for servo and regulator systems using heuristic methods
3. Compare behavior of PIDs tuned differently using qualitative and quantitative methods

To *tune* a feedback control system means to adjust parameters in the controller to achieve good control over the process. "Good" in this context is usually defined as stability of the process variable despite changes in load, fast response to changes in setpoint, minimal oscillation following either type of change, and minimal offset (error between setpoint and process variable) over time. However, "good" can also be defined mathematically based on calculations made on the controlled variable signal.

"Good control" is far easier to define than it is to achieve. With PID (Proportional-Integral-Derivative) control being the most common feedback control algorithm used in industry, it is important for the controls engineer to understand how to tune these controllers effectively and with a minimum investment of time.

Different types of processes, having different dynamic (time-dependent) behaviors, require different levels of proportional, integral, and derivative control action to achieve stability and acceptable dynamic response. It is therefore imperative for anyone seeking to tune a PID controller to understand the dynamic nature of the process being controlled. For this reason, the chapter begins with an exploration of common process characteristics before introducing techniques useful in choosing practical P, I, and D tuning parameter values.

## 23.1 Before tuning

Much has been written about the benefits of PID control. Increased productivity, decreased equipment strain, and increased process safety are some of the advantages touted of proper PID tuning. What is often overlooked, though, are the negative consequences of poor PID controller tuning. If good PID control can increased productivity, then poor PID control can decrease productivity. If a well-tuned system helps equipment run longer and safer, then a badly tuned system may increased failure frequency and safety incidents. The instrumentation professional should be mindful of this dichotomy when proceeding to tune a PID control system. One should never think there is "nothing to lose" by trying different PID settings. Tuning a PID controller is as serious a matter as the productivity and safety impact of the process itself.

PID tuning parameters are easy to access, which makes them a tempting place to begin for technicians looking to improve the performance of a feedback loop. Another temptation driving technicians to focus on controller tuning as a first step is the prestige associated with being able to tame an unruly feedback loop with a few magical adjustments to the controller's PID tuning constants. For those who do not understand PID control (and this constitutes the vast majority of the human population, even in the industrial world), there is something "magic" about being able to achieve acceptable control behavior simply by making small adjustments to numbers in a computer (or to knobs in an analog controller). The reality, though, is that many poorly-behaving control systems are that way not due (at least purely) to a deficit of proper PID tuning values, but rather to problems external to the controller which no amount of "tuning" will solve. Adjusting PID tuning constants as a *first step* is almost always a bad idea.

This section aims to describe and explain some of the recommended considerations prior to making adjustments to the tuning of a loop controller. These considerations include:

- Identifying operational needs (i.e., "How do the operators want the system to respond?")
- Identifying process and system hazards before manipulating the loop
- Identifying whether it is a tuning problem, a field instrument problem, and/or a design problem

### 23.1.1 Identifying operational needs

As defined elsewhere in this book, "good" control is a stability of the process variable despite changes in load, fast response to changes in setpoint, minimal oscillation following either type of change, and minimal offset (error between setpoint and process variable) over time. However, these criteria are not equally valued in all processes, and neither are they equally attainable with simple PID control in all processes. It may be critical, for example, in a boiler water level control process to have fast response to changes in load, but minimal offset over time is not as important. It may be completely permissible to have a persistent 5% error between PV and SP in such a system, so long as the water level does not deviate much over 20% for any length of time due to load changes. In another process, such as liquid level control inside one stage ("effect") of a multi-stage ("multi-effect") evaporator system, a priority may be placed upon relatively steady flow control through the valve rather than steady level in the process. A level controller tuned for aggressive response to setpoint changes will cause large fluctuations in liquid flow rate to all successive stages ("effects") of the evaporator process in the event of a sudden load or setpoint change, which would be more detrimental to product quality than some deviation from setpoint in that one effect.

Thus, there is a need for whomever intends to tune a control system to determine the operational needs of the system. The operations personnel (operators, unit managers, process engineers) are the best resources here. Ultimately, they are the "customers." The loop tuner's task is to give the customers the system performance they need to do their jobs best.

Keep in mind the following process control objectives, knowing that it will likely be impossible to achieve *all* of them with any particular PID tuning. Try to rank the relative importance of these objectives, then concentrate on achieving those most important, at the expense of those least important:

- Minimum change in PV (dynamic stability) with changes in load
- Fast response to setpoint changes (minimum dynamic error)
- Minimum overshoot/undershoot/oscillation following sudden load or setpoint changes
- Minimum error (PV − SP) over time
- Minimum valve velocity

The control actions best suited for rapid response to load and/or setpoint changes are proportional and derivative. Integral action takes effect only *after* error has had time to develop, and as such cannot act as immediately as either proportional or derivative.

If the priority is to minimize overshoot, undershoot, and/or oscillations, the controller's response will likely need to be more sluggish than is typical. New setpoint values will take longer to achieve, and load changes will not be responded to with quite the same vigor. Derivative action may be helpful in some applications to "tame" the oscillatory tendencies of proportional and integral actions.

Minimum error over time can really only be addressed by integral action. No other controller action pays specific "attention" to the magnitude and duration of error. This is not to say that the process will work well on integral-only control, but rather that integral action will be absolutely necessary (i.e., a P-only or PD controller will not suffice).

Minimum valve velocity is a priority in processes where the manipulated variable has an effect on some *other* process in the system. For example, liquid level control in the multi-stage (multi-"effect") evaporator system in Figure 23.1 where the discharge flow from one evaporator becomes the incoming flow for another evaporator, is a system where sharp changes in the manipulated variable of one control loop can upset downstream processes.

Figure 23.1: Tight level control in an upstream evaporator can produce unwanted effects in the level of downstream units

In other words, an aggressively-tuned level controller on an upstream evaporator (e.g., Effect 1) may achieve its goal of holding liquid level very steady in that evaporator by varying its out-going flow, but it will do so at the expense of causing level variations in all downstream evaporators. Cases such as this call for controller tuning (at least in the upstream effects) responding slowly to errors. Proportional action will very likely be limited to low gain values (high proportional band values), and derivative action (if any is used at all) should be set to respond only to the process variable, not to error (PV − SP). Understand that tuning a PID loop with the goal of minimizing valve motion *will* result in more deviation from setpoint than if the controller were tuned more aggressively.

### 23.1.2 Identifying process and system hazards

When students practice PID control in an Instrumentation program, they usually do so using computer simulation software and/or "toy" processes constructed in a lab environment. A potential disadvantage to this learning environment is a failure to recognize real problems that may develop when tuning an actual production process. A completely isolated feedback loop is a rarity in industry: generally there are interactive effects between control loops in a process, which means one cannot proceed to tune a loop with impunity.

A very important question to ask the operations personnel before tuning a loop is, "How far and how fast is the process variable allowed to increase and decrease?" Processes and process equipment can become dangerously unstable, for example, if certain temperatures become to high (or too low, as is the case in process liquids that solidify when cold) or if the rate at which temperature ramps up or down is too high or low (thermal stresses are proportional to the rate of temperature change). It is not uncommon for certain control loops in a process to be equipped with alarms, either hard or soft, that automatically *shut down* equipment if exceeded. Clearly, these "shutdown" limits must be avoided during the tuning of the process loop.

One should also examine the control strategy before proceeding to tune. Is this a cascaded loop? If so, the slave controller needs to be tuned before the master. Does this loop have feedforward added incorporated to handle load changes? If so, the effectiveness of that feedforward loop (gain, dynamic compensation) should be checked and adjusted before the feedback loop is tuned. Are there limits in this loop? Is this a selector or override control strategy? Which loop components are selected and which signals are being limited must be clear at any given time.

Another consideration is whether or not the process is in a "normal" condition before attempting to improve its performance. Ask the operations personnel if this is a typical day, or if there is some abnormal condition in effect (equipment shutdown, re-routing of flows, significantly different production rates, etc.) that might skew the response of the process loop to be tuned. Once again there is a need for input from the operations personnel, because they know the day-to-day behavior of the system better than anyone else.

### 23.1.3 Identifying the problem(s)

One of the questions to ask of operators when diagnosing any process problem is simply, "How long has this problem existed?" The age of a problem can be a very important indicator of possible causes. If the problem suddenly developed after the last night shift, be inclined to suspect an equipment failure, or something else that could happen *suddenly* (e.g., a hand valve someone opened or shut when they shouldn't have). Alternatively, if a problem has been in existence since the day the process was constructed, be more inclined to suspect an issue with system design or improper installation. This same diagnostic technique – obtaining a "history" of the "patient" – applies to loop tuning as well. A control loop that suddenly stopped working as it should might be suffering from an instrument failure (or an unauthorized change of controller parameters), whereas a chronically misbehaving loop would more likely be suffering from poor design, bad instrument installation, or a controller that was never tuned properly.

In either case, poor control is just as likely to be caused by field instrument problems as it is by incorrect PID tuning parameters. No PID settings can possibly compensate for faulty instrumentation, but it is possible for some instrument problems to be "masked" by controller tuning. Your first step in actually manipulating the control loop should be a check of instrument health. Thankfully, this is relatively easy to do by performing a series of "step-change" tests with the controller in manual mode. By placing the controller in manual and making small changes in output signal (remember to check with operations to see how far the output can be moved, and how far the PV is allowed to drift!), much about the process and the loop instrumentation can be determined, including:

- Whether the PV signal is "noisy" (first turn off all damping in the controller and transmitter)
- How much "stiction" is in the control valve
- Whether the process is integrating or self-regulating
- Process gain (and whether this gain is stable or if it changes as PV changes)
- Process lag time and lag degree (first-order versus multiple-order)
- Process dead time

Such an open-loop test might reveal potential problems without pinpointing the exact nature or location of those problems. For example, a large lag time may be intrinsic to the process, or it may be the result of a poorly-installed sensor (e.g., a thermocouple not pushed to touch the bottom of its thermowell) or even a control valve in need of a volume booster or positioner. Dead time measured in an open-loop test may also be intrinsic to the process, or it could be the result of stiction in the valve. The only way to definitively identify the problem is to test the instruments themselves, ideally in the field location.

In order to obtain the best "view" of the process, make sure the graphing trend display has sufficient resolution and responsiveness. If the trend fails to show fine details such as noise in the

process, it is possible that the graphing device will be insufficient for your needs.

If this is the case, response tests of the loop may still be performed, but some other tool will have to be used to graph the controller and process actions. A modern tool useful for this purpose is a portable computer with a data acquisition device connected, giving the computer the ability to read instrument signal voltages. Many data graphing programs exist for taking acquired data and plotting it over the time domain. Data acquisition modules with sample rates in the thousands of samples per second are available for very modest prices.

### 23.1.4 Final precautions

Be prepared to document all work! This means capturing and recording "screen shot" images of process trend graphs, both for the initial open-loop tests and the closed-loop PID trials. It also means documenting the original PID settings of the controller, and all PID setting values attempted during the tuning process (linked to their respective trend graphs, so it will be easy to tell which sets of PID constants produced which process responses). If there are any instrument configuration settings (e.g., damping time values in process transmitters) changed during the tuning exercise, both the original values and all changes need to be documented as well.

Many I&C engineer are against the use of auto-tuning controllers. With all due respect for the engineers who work hard to make controllers "intelligent" enough to adjust their own PID settings, there is no controller in the world that can account for all the factors described in this "Before tuning" section. Feel free to use the automatic tuning feature of a controller, but only *after* all of the instrument and process problems are corrected, and *after* confirming the tuning goal of the controller matches the behavioral goal of the control loop as defined by the operators (e.g., fast response versus minimum overshoot, etc.). Some people in the automation business are over-confident with regard to the capabilities of auto-tuning controllers. This feature is a *tool*, not unlike any other tool in that it has specific purposes and limitations, and is certainly no panacea for a lack of understanding how PID control works.

## 23.2 Heuristic PID tuning procedures

A *heuristic* PID tuning procedure is a step-by-step approach leading directly to a set of numerical values to be used in a PID controller. These procedures may be split into two categories: *open loop* and *closed loop*. An "open loop" tuning procedure is implemented with the controller in manual mode: introducing a step-change to the controller output and then mathematically analyzing the results of the process variable response to calculate appropriate PID settings for the controller to use when placed into automatic mode. A "closed loop" tuning procedure is implemented with the controller in automatic mode: adjusting tuning parameters to achieve an easily-defined result, then using those PID parameter values and information from a graph of the process variable over time to calculate new PID parameters.

Heuristic PID tuning got its start with a paper published in the November 1942 *Transactions of the American Society of Mechanical Engineers* written by two engineers named Ziegler and Nichols. "Optimum Settings For Automatic Controllers" is a seminal paper, and deserves to be read by any serious student of process control theory [1]. That Ziegler's and Nichols' recommendations for PID

controller settings may still be found in modern references more than 60 years after publication is a testament to its impact in the field of industrial control. Although dated in its terminology and references to pneumatic controller technology (some controllers mentioned as not even having adjustable proportional response, and others as having only discrete degrees of reset adjustment rather than continuously variable settings!), the PID algorithm described by its authors and the effects of P, I, and D adjustments on process control behavior are as valid today as they were then.

This section is devoted to a discussion of heuristic PID tuning procedures in general, and the "Ziegler-Nichols" methods in specific. Currently, the Ziegler-Nichols tuning methods are useful primarily as historical references, and indeed suffer from serious practical impediments. Perhaps the most serious problem with the Ziegler-Nichols methods (and in fact any heuristic procedure for PID tuning) is that it tends to absolve the practitioner of responsibility for understanding the process they intend to tune. Any time people are provided with step-by-step instructions to perform complex tasks, there will be a great many readers of those instructions tempted to mindlessly follow the instructions, even to their doom. PID tuning is one of these "complex tasks," and there is significant likelihood for a person to do more harm than good if all they do is implement a step-by-step approach rather than understand what they are doing, why they are doing it, and what it means if the results do not meet with satisfaction. Please bear this in mind with regards to any PID tuning procedure, Ziegler-Nichols or otherwise.

### 23.2.1 Ziegler-Nichols closed-loop ("Ultimate Gain")

*Closed-loop* refers to the operation of a control system with the controlling device in "automatic" mode, where the flow of the information from sensing element to transmitter to controller to control element to process and back to sensor represents a continuous ("closed") feedback loop. If the total amount of signal amplification provided by the instruments is too much, the feedback loop will self-oscillate. While oscillation is almost always considered undesirable in a control system, it may be used as an exploratory test of process dynamics if the controller acts purely on proportional action (no integral or derivative action): providing data useful for calculating effective PID controller settings. Thus, a "closed-loop" PID tuning procedure entails disabling any integral or derivative actions in the controller, then raising the gain value of the controller until self-sustaining oscillations of constant amplitude are achieved in the process variable[1]. The amount of controller gain necessary to sustain process oscillations of consistent amplitude is called the *ultimate sensitivity* ($S_u$) of the process, while the time (period) between successive oscillation peaks is called the *ultimate period* ($P_u$) of the process.

When performing such a test on a process loop, it is important to ensure the oscillation peaks do not reach the limits of the instrumentation, either measurement or final control element. In other words, in order for the oscillation to accurately reveal the process characteristics of ultimate sensitivity and ultimate period, the oscillations must be naturally limited and not artificially limited by either the transmitter or the control valve saturating. Oscillations characterized by either the transmitter or the final control element reaching their range limits should be avoided in order to obtain the best closed-loop oscillatory test results. An illustration is shown in Figure 23.2 as a model of what to avoid.

[1] Note that in order to make the closed-loop system oscillate, the dominant poles of the system must be placed on the imaginary axis in the complex plane. Therefore, the system order has to be third-order or higher, meaning that some of the branches of the root locus cross the imaginary axis into the right half of the complex plane.

Figure 23.2: Example of a limit cycle oscillation

If the controller in question is proportional-only (i.e., capable of providing no integral or derivative control actions), Ziegler and Nichols' recommendation is to set the controller gain[2] to one-half the value of the ultimate sensitivity, which will be called ultimate *gain* ($K_u$) from now on:

$$K_p = 0.5K_u$$

Where,
$K_p$ = Controller gain
$K_u$ = "Ultimate" gain determined by increasing controller gain until self-sustaining oscillations are achieved

Generally, a controller gain of one-half the experimentally determined "ultimate" gain results in reasonably quick response to setpoint and process load changes. Oscillations of the process variable following such setpoint and load changes typically damp with each successive wave peak being approximately one-quarter the amplitude of the one preceding. This is known as *quarter-wave damping*. While certainly not ideal, it is a compromise between fast response and stability.

The process trend in Figure 23.3 shows what "quarter-wave damping" looks like, with the process variable (PV) oscillating a bit following a step-change in setpoint (SP).

[2]Note that this is truly the *gain* of the controller, not the *proportional band*. If a proportional band value is entered one-half the proportional band value necessary to sustain oscillations, the controller would (obviously) oscillate completely out of control!

Figure 23.3: Example of quarter-wave damping

Ziegler and Nichols were careful to qualify quarter-wave damping as less than optimal for some applications. In their own words (page 761):

> "The statement that a sensitivity setting of one half the ultimate with attendant 25 per cent amplitude ratio gives optimum control must be modified in some cases. For example, the actual level maintained by a liquid-level controller might not be nearly as important as the effect of sudden valve movements on further portions of the process. In this case the sensitivity should be lowered to reduce the amplitude ratio even though the offset is increased by so doing. On the other hand, a pressure-control application giving oscillations with very short period could be set to give an 80 or 90 per cent amplitude ratio. Due to the short period, a disturbance would die out in reasonable time, even though there were quite a few oscillations. The offset would be reduced somewhat though it should be kept in mind that it can never be reduced to less than one half of the amount given at our previously defined optimum sensitivity of one half the ultimate."

Many engineers and technicians have argued that quarter-wave damping exercises the control valve needlessly, causing undue stem packing wear over time and consuming large quantities of compressed air to operate. Given the fact that all modern process controllers have integral (reset) capability, unlike the simple pneumatic controllers of Ziegler and Nichols' day, there is really no need to tolerate prolonged offset (failure of the process variable to exactly equalize with setpoint over time) as a necessary cost of avoiding valve oscillation.

If the controller in question has integral (reset) action in addition to proportional, Ziegler and Nichols' recommendation is to set the controller gain to slightly less than one-half the value of the ultimate sensitivity, and to set the integral time constant[3] to a value slightly less than the ultimate

[3]Either minutes per repeat or seconds per repeat. If the controller's integral rate is expressed in units of repeats per minute (or second), the formula would be $K_i = \frac{1.2}{P_u}$.

period:

$$K_p = 0.45K_u$$

$$\tau_i = \frac{P_u}{1.2}$$

Where,
$K_p$ = Controller gain
$K_u$ = "Ultimate" gain determined by increasing controller gain until self-sustaining oscillations are achieved
$\tau_i$ = Controller integral setting (minutes per repeat)
$P_u$ = "Ultimate" period of self-sustaining oscillations (minutes)

If the controller in question has all three control actions present (full PID), Ziegler and Nichols' recommendation is to set the controller tuning constants as follows:

$$K_p = 0.6K_u$$

$$\tau_i = \frac{P_u}{2}$$

$$\tau_d = \frac{P_u}{8}$$

Where,
$K_p$ = Controller gain
$K_u$ = "Ultimate" gain determined by increasing controller gain until self-sustaining oscillations are achieved
$\tau_i$ = Controller integral setting (minutes per repeat)
$\tau_d$ = Controller derivative setting (minutes)
$P_u$ = "Ultimate" period of self-sustaining oscillations (minutes)

An important caveat with any tuning procedure based on ultimate gain is the potential to cause trouble in a process while experimentally determining the ultimate gain. Recall that "ultimate" gain is the amount of controller gain (proportional action) resulting in self-sustaining oscillations of constant amplitude. In order to precisely determine this gain setting, one must spend some time provoking the process with sudden setpoint changes (to induce oscillation) and experimenting with greater and greater gain settings until constant oscillation amplitude is achieved. Any more gain than the "ultimate" value, of course, leads to ever-*growing* oscillations which may be brought under control only by decreasing controller gain or switching to manual mode (thereby stopping all feedback in the system). The problem with this is, one never knows for certain when ultimate gain is achieved until this critical value has been exceeded, as evidenced by ever-growing oscillations. In other words, *the system must be brought to the brink of total instability in order to determine its ultimate gain value.* Not only is this time-consuming to achieve – especially in systems where the natural period of oscillation is long, as is the case with many temperature and composition control applications – but potentially hazardous to equipment and certainly detrimental to process

quality[4]. In fact, one might argue that any process tolerant of such abuse probably doesn't need to be well-tuned!

Despite its practical limitations, the rules given by Ziegler and Nichols do shed light on the relationship between realistic P, I, and D tuning parameters and the operational characteristics of the process. Controller gain should be some fraction of the gain necessary for the process to self-oscillate. Integral time constant should be proportional to the process time constant; i.e., the "slower" the process is to respond, the "slower" (less aggressive) the controller's integral response should be. Derivative time constant should likewise be proportional to the process time constant, although this has the opposite meaning from the perspective of aggressiveness: a "slow" process deserves a long derivative time constant; i.e., *more aggressive* derivative action.

### 23.2.2 Ziegler-Nichols open-loop

In contrast to the first tuning technique presented by Ziegler and Nichols in their landmark 1942 paper where the process was made to oscillate using proportional-only automatic control and the parameters of that oscillation served to define PID tuning parameters, their second tuning technique did not even rely on the presence of a controller. Instead, this second technique consisted of making a manual "step-change" of the control element (valve) and analyzing the resulting effect on the process variable.

After making the step-change in output signal with the controller in manual mode, the process variable trend is closely analyzed for two salient features: the *dead time* and the *reaction rate*. Dead time ($L$)[5] is the amount of time delay between the output step-change and the first indication of process variable change. Reaction rate is the maximum rate at which the process variable changes following the output step-change (the maximum time-derivative of the process variable). The measurement of reaction rate is shown graphically in Figure 23.4.

[4]For a graphic demonstration of just how impractical this may be, just try telling a lead operations manager or a unit supervisor in a chemical processing facility that the temperature controller in the main reaction furnace or the pressure controller in one of the larger distillation towers will be tuned until it nearly oscillates out of control, and that doing so may necessitate hours of unstable operation before the perfect gain setting is found. An escort to the control room door by security personnel following this declaration of intent is a likely outcome.

[5]Unfortunately, Ziegler and Nichols chose to refer to dead time by the word *lag* in their paper. In modern technical parlance, "lag" refers to a first-order inverse-exponential function, which is fundamentally different from dead time.

Figure 23.4: Example of a limit cycle oscillation

Dead time and reaction rate are responses common to self-regulating and integrating processes alike. Whether or not the process variable ends up stabilizing at some new value, its rate of rise will reach some maximum value following the output step-change, and this will be the reaction rate of the process[6]. The unit of measurement for reaction rate is *percent per minute*:

$$R = \frac{\Delta \text{PV}}{\Delta t} = \frac{[\text{Percent rise}]}{[\text{Minutes run}]}$$

While dead time in a process tends to be constant regardless of the output step-change magnitude, reaction rate tends to vary directly with the magnitude of the output step-change. For example, an output step-change of 10% will generally cause the PV to rise at a rate twice as steep compared to the effects of a 5% output step-change. In order to ensure the predictive calculations capture only what is inherent to the process and not arbitrary open-loop tuning actions, include the output step-change magnitude ($\Delta m$) in those calculations as well[7].

[6] Right away, a weakness in the Ziegler-Nichols open-loop method can be identified: it makes absolutely no distinction between self-regulating and integrating process types. This is problematic from the analysis of each process type in Chapter 18 beginning on page 778.

[7] Ziegler and Nichols' approach was to define a normalized reaction rate called the *unit reaction rate*, equal in

If the controller in question is proportional-only (i.e., capable of providing no integral or derivative control actions), Ziegler and Nichols' recommendation is to set the controller gain as follows:

$$K_p = \frac{\Delta m}{RL}$$

Where,
$K_p$ = Controller gain (unitless)
$\Delta m$ = Output step-change magnitude made while testing in open-loop (manual) mode (percent)
$R$ = Process reaction rate = $\frac{\Delta \text{PV}}{\Delta t}$ (percent per minute)
$L$ = Process dead time (minutes)

If the controller in question has integral (reset) action in addition to proportional, Ziegler and Nichols' recommendation is to set the controller gain to 90% of the proportional-only value, and to set the integral time constant to a value just over three times the dead time value:

$$K_p = 0.9\frac{\Delta m}{RL}$$

$$\tau_i = 3.33L$$

Where,
$K_p$ = Controller gain (unitless)
$\Delta m$ = Output step-change magnitude made while testing in open-loop (manual) mode (percent)
$R$ = Process reaction rate = $\frac{\Delta \text{PV}}{\Delta t}$ (percent per minute)
$L$ = Process dead time (minutes)
$\tau_i$ = Controller integral setting (minutes per repeat)

If the controller has full PID capability, Ziegler and Nichols' recommendation is to set the controller gain to 120% of the proportional-only value, to set the integral time constant to twice the dead time value, and to set the derivative time constant to one-half the dead time value:

$$K_p = 1.2\frac{\Delta m}{RL}$$

$$\tau_i = 2L$$

$$\tau_d = 0.5L$$

Where,
$K_p$ = Controller gain (unitless)
$\Delta m$ = Output step-change magnitude made while testing in open-loop (manual) mode (percent)
$R$ = Process reaction rate = $\frac{\Delta \text{PV}}{\Delta t}$ (percent per minute)
$L$ = Process dead time (minutes)
$\tau_i$ = Controller integral setting (minutes per repeat)
$\tau_d$ = Controller derivative setting (minutes)

value to $\frac{R}{\Delta m}$. It may be helpful to explicitly include $\Delta m$ in all the tuning parameter equations in order to avoid the possibility of confusing reaction rate with unit reaction rate.

### 23.2.3 Other heuristic PID tuning procedures

In contrast to Ziegler-Nichols or other tuning procedures where definite numerical values for P, I, and D controller settings are obtained through data collection and analysis, another type of *heuristic* tuning procedure is one where general rules are followed to obtain approximate or qualitative results. The majority of PID loops in the world have been tuned with such methods, for better or for worse (usually for worse). The goal in this section is to optimize the effectiveness of such tuning methods.

Here is one (crude) step-by-step procedure for loop tuning:

1. Configure the controller for proportional action only (integral and derivative control actions set to minimum effect), setting the gain near or at 1.
2. Increase controller gain until self-sustaining oscillations are achieved, "bumping" the setpoint value up or down as necessary to provoke oscillations.
3. When the ultimate gain is determined, set the controller gain for half that value.
4. Repeat steps 2 and 3, this time adjusting integral action instead of proportional.
5. Repeat steps 2 and 3, this time adjusting derivative action instead of proportional.

The first three steps of this procedure are identical to the steps recommended by Ziegler and Nichols for closed-loop tuning. The last two steps are someone else's contribution. It showcases the general spirit of most heuristic tuning methods: adjust each controller parameter to be more and more aggressive until some compromise is reached between fast action and instability.

Much improvement may be made to any "trial-and-terror" PID tuning procedure if one is aware of the process characteristics and recognizes the applicability of P, I, and D actions to particular characteristics. Simply experimenting with P, I, and D parameter values is tedious at best and dangerous at worst if one has no understanding of what each type of control action if useful for, and the limitations of each control action.

### 23.2.4 Features of P, I, and D actions

Purpose of each action

- **Proportional action** is the "universal" control action, capable of providing at least marginal control quality for any process.
- **Integral action** is useful for eliminating offset caused by load variations and process self-regulation.
- **Derivative action** is useful for canceling lags.

Limitations of each action

- **Proportional action** will cause oscillations if sufficiently aggressive, in the presence of lags and/or dead time. The more lags (higher-order), the worse the problem. It also directly amplifies process noise.
- **Integral action** will cause oscillation if sufficiently aggressive, in the presence of lags and/or dead time. Any amount of integral action will guarantee setpoint overshoot in purely integrating processes.
- **Derivative action** dramatically amplifies process noise, and will cause oscillations in fast-acting processes.

Special applicability of each action

- **Proportional action** works exceptionally well when aggressively applied to self-regulating processes dominated by first-order lag, and to purely integrating processes.
- **Integral action** works exceptionally well when aggressively applied to fast-acting, self-regulating processes. Has the unique ability to ignore process noise.
- **Derivative action** works exceptionally well to speed up the response of processes dominated by large lag times, and to help stabilize runaway processes.

Gain and phase shift of each action

- **Proportional action** acts on the *present*, adding no phase shift to a sinusoidal signal. Its gain is constant for any signal frequency.
- **Integral action** acts on the *past*, adding a $-90^o$ phase shift to a sinusoidal signal. Its gain decreases with increasing frequency.
- **Derivative action** acts on the *future*, adding a $+90^o$ phase shift to a sinusoidal signal. Its gain increases with increasing frequency.

### 23.2.5 Tuning recommendations based on process dynamics

Knowing which control actions to focus on first is a matter of characterizing the process (identifying whether it is self-regulating, integrating, runaway, noisy, has lag or dead time, or any combination of these traits based on an open-loop response test[8]) and then selecting the best actions to fit those characteristics. Figure 23.5 shows some general recommendations for fitting PID tuning to different process characteristics.

[8]Recall that an open-loop response test consists of placing the loop controller in manual mode, introducing a step-change to the controller output (manipulated variable), and analyzing the time-domain response of the process variable as it reacts to that perturbation.

| | | |
|---|---|---|
| | Pure self-regulating | *May be controlled with aggressive integral action, and perhaps with a bit of proportional action. Use absolutely no derivative action!* |
| | Self-reg w/ pure 1st order lag | *Responds well to aggressive proportional action, with integral action needed only for recovery from load changes.* |
| | Self-reg w/ multiple lags | *Proportional action needed for quick response to setpoint changes, integral action needed for recovery from load changes, and derivative needed to prevent overshoot. Proportional and integral actions are limited by tendency to oscillate.* |
| | Integrating w/ lag(s) | *Proportional action should be aggressive as possible without generating oscillations. Integral action needed only for recovery from load changes.* |
| | Pure integrating | *Responds well to aggressive proportional action, with integral action needed only for recovery from load changes.* |

Figure 23.5: Recommended PID control actions for various types of processes

**General rules:**

- Use no derivative action if the process signal is "noisy"
- Use proportional action sparingly if the process signal is "noisy"
- The slower the time lag(s), the less integral action to use
- The higher-order the time lag(s), the less proportional action (gain) to use
- Self-regulating processes *need* integral action
- Integrating processes *need* proportional action
- Dead time requires a reduction of all PID constants below what would normally work

Once the basic character of the process has been determined, and from that characterization the needs of the process are understood regarding P, I, and/or D control actions, "experiment" with different tuning values of P, I, and D until a combination is found yielding good control.

## 23.3 Tuning techniques compared

This section shows screenshots from a process loop simulation program illustrating the effectiveness of Ziegler-Nichols open-loop ("Reaction Rate") and closed-loop ("Ultimate") PID tuning methods, and then contrast them against the results of my own heuristic tuning. As will be seen in each case, the results obtained by either Ziegler-Nichols method were quite poor. This is not necessarily an indictment of Ziegler's and Nichols' recommendations as much as it is a demonstration of the power of understanding. Ziegler and Nichols presented a simple step-by-step procedure for obtaining

*approximate* PID tuning constant values based on closed-loop and open-loop process responses, which could be applied by anyone regardless of their level of understanding PID control theory. Ultimately, good PID control is attainable only at the hands of someone who understands how PID works, what each mode does (and why), and is able to distinguish between intrinsic process characteristics and instrument limitations. The purpose of this section is to clearly demonstrate the limitations of ignorantly-followed procedures, and contrast this "mindless" approach against the results of simple experimentation directed by qualitative understanding.

Each of the examples illustrated in this section were simulations run on a computer program called *PC-ControLab* developed by Wade Associates, Inc. Although these are simulated processes, in general similar results have been obtained using both Ziegler-Nichols tuning methods on real processes. The control criteria used for tuning were fast response to setpoint changes, with minimal overshoot or oscillation.

### 23.3.1 Tuning a "generic" process

The first process tuned in simulation was a "generic" process, unspecific in its nature or application. Performing an open-loop test (two 10% output step-changes made in manual mode, both increasing) on this process resulted in the behavior shown in Figure 23.6.

Figure 23.6: Step response of a generic process

From the trend, this process is self-regulating, with multiple lags and some dead time. The reaction rate ($R$) is 20% over 15 minutes, or 1.333 percent per minute. Dead time ($L$) appears to

be approximately 2 minutes. Following the Ziegler-Nichols recommendations for PID tuning based on these process characteristics (also including the 10% step-change magnitude $\Delta m$):

$$K_p = 1.2\frac{\Delta m}{RL} = 1.2\frac{10\%}{\frac{20\%}{15\text{ min}}2\text{ min}} = 4.5$$

$$\tau_i = 2L = (2)(2\text{ min}) = 4\text{ min}$$

$$\tau_d = 0.5L = (0.5)(2\text{ min}) = 1\text{ min}$$

Applying the PID values of 4.5 (gain), 4 minutes per repeat (integral), and 1 minute (derivative) gave the result in Figure 23.7 when in automatic mode.

Figure 23.7: Closed-loop setpoint step response of a generic process with open-loop Ziegler-Nichols tuning

The process oscillations follow a 10% setpoint change (from 60% to 50%), and take almost an hour to settle. Clearly, this is less-than-desirable behavior. The controller is much too aggressive, which is why the process oscillates so much following the setpoint step-change.

Looking closely at the PV and OUT waveforms shows that their phase relationship is nearly $180^o$, consistent with what is expected for a reverse-acting controller with strong proportional action. If integral or derivative action were primarily responsible for the oscillation, the influence of an additional $90^o$ phase shift characteristic to those actions would be seen (integral naturally produces a -$90^o$ phase shift, while derivative naturally produces a +$90^o$ phase shift, for any sinusoidal function).

Thus, if the oscillations were primarily the result of excessive integral action, expect the OUT wave to lead the PV wave by nearly $90^o$ ($180^o$ from reverse action $-$ $90^o$ from integral action $= 90^o$). If excessive derivative action were primarily responsible for the oscillations, expect the OUT wave to lag the PV wave by nearly $90^o$ ($180^o$ from reverse action $+$ $90^o$ from derivative action $= 270^o = -90^o$). Since there is a phase shift of the oscillations between OUT and PV very close to the $180^o$ predicted by proportional action, this controller's response is dominated by proportional action, which would be a good place to start "taming" this over-exuberant controller if by modifying the Ziegler-Nichols tuning recommendations.

Next, the closed-loop, or "Ultimate" tuning method of Ziegler and Nichols was applied to this process. Eliminating both integral and derivative control actions from the controller, and experimenting with different gain (proportional) values until self-sustaining oscillations of consistent amplitude[9] were obtained gave a gain value of 11. The system behavior under this proportional gain is shown in Figure 23.8.

Figure 23.8: Determination of ultimate values for Ziegler-Nichols tuning

From the trend in Figure 23.8, the ultimate period ($P_u$) is approximately 7 minutes in length. Following the Ziegler-Nichols recommendations for PID tuning based on these process characteristics:

[9]The astute observer will note the presence of some limiting (saturation) in the output waveform, as it attempts to go below zero percent. Normally, this is unacceptable while determining the ultimate gain of a process, but here it was virtually impossible to make the process oscillate at consistent amplitude without saturating on the output signal. The gain of this process falls off quite a bit at the ultimate frequency, such that a high controller gain is necessary to sustain oscillations, causing the output waveform to have a large amplitude.

$$K_p = 0.6K_u = (0.6)(11) = 6.6$$

$$\tau_i = \frac{P_u}{2} = \frac{7 \text{ min}}{2} = 3.5 \text{ min}$$

$$\tau_d = \frac{P_u}{8} = \frac{7 \text{ min}}{8} = 0.875 \text{ min}$$

It should be immediately apparent that these tuning parameters will yield poor control. While the integral and derivative values are close to those predicted by the open-loop (Reaction Rate) method, the gain value calculated here is even larger than what was calculated before. Since the proportional action was excessive in the last tuning attempt, and this one recommends an even higher gain value, expect the next trial to oscillate even worse.

Applying the PID values of 6.6 (gain), 3.5 minutes per repeat (integral), and 0.875 minute (derivative) gave the result shown in Figure 23.9 in automatic mode:

Figure 23.9: Closed-loop setpoint step response of a generic process with closed-loop Ziegler-Nichols tuning

The process oscillations follow a 10% setpoint change (from 60% to 50%), and are nowhere near settled after an hour's worth of time. Once again there is nearly $180^o$ phase shift between the OUT and the PV waves, indicating proportional reverse-action controller response. As expected, this is far too much gain for good control!

From the initial open-loop (manual output step-change) test, this process can be seen to contain multiple lags in addition to about 2 minutes of dead time. Both of these factors tend to limit the amount of gain that can be used in the controller before the process oscillates. Both Ziegler-Nichols tuning attempts confirmed this fact, which led to much lower gain values. Given the self-regulating nature of the process, the controller needed integral action, but once again the aggressiveness of this action would be necessarily limited by the lag and dead times. Derivative action, however, would prove to be useful in its ability to help "cancel" lags, so the PID was suspected to need relatively tame proportional and integral values, with a relatively aggressive derivative value.

After some experimenting, the values arrived at were 1.5 (gain), 10 minutes (integral), and 5 minutes (derivative). These tuning values represent a proportional action only one-third as aggressive as the least-aggressive Ziegler-Nichols recommendation, an integral action less than half as aggressive as the Ziegler-Nichols recommendations, and a derivative action *five times* more aggressive than the most aggressive Ziegler-Nichols recommendation. The results of these tuning values in automatic mode are shown in Figure 23.10.

Figure 23.10: Closed-loop setpoint step response of a generic process with better tuning

With this PID tuning, the process responded well to not just one, but *two* 10% setpoint step-changes within the span of an hour.

### 23.3.2 Tuning a liquid level process

The next simulated process attempted was a liquid level-control process. Performing an open-loop test (one 10% increasing output step-change, followed by a 10% decreasing output step-change, both

made in manual mode) on this process resulted in the behavior shown in Figure 23.11.

Figure 23.11: Open-loop output step response of a liquid-level process

From the trend, the process appears to be purely integrating, as though the control valve were throttling the flow of liquid into a vessel with a constant out-flow. The reaction rate ($R$) on the first step-change is 50% over 10 minutes, or 5 percent per minute. Dead time ($L$) appears virtually nonexistent, estimated to be 0.1 minutes simply for the sake of having a dead-time value to use in the Ziegler-Nichols equations. Following the Ziegler-Nichols recommendations for PID tuning based on these process characteristics (also including the 10% step-change magnitude $\Delta m$):

$$K_p = 1.2\frac{\Delta m}{RL} = 1.2\frac{10\%}{\frac{50\%}{10 \text{ min}}0.1 \text{ min}} = 24$$

$$\tau_i = 2L = (2)(0.1 \text{ min}) = 0.2 \text{ min}$$

$$\tau_d = 0.5L = (0.5)(0.1 \text{ min}) = 0.05 \text{ min}$$

Applying the PID values of 24 (gain), 0.2 minutes per repeat (integral), and 0.05 minutes (derivative) gave the result shown in Figure 23.12 when in automatic mode:

Figure 23.12: Closed-loop setpoint step response of a liquid-level process with open-loop Ziegler-Nichols tuning

The process variable certainly responds rapidly to the five increasing setpoint changes and also to the one large decreasing setpoint change, but the valve action is hopelessly chaotic. Not only would this "jittery" valve motion prematurely wear out the stem packing, but it would also result in vast over-consumption of compressed air to continually stroke the valve from one extreme to the other. Furthermore, there is evidence of "overshoot" at every setpoint change, most likely from excessive integral action.

The valve's wild behavior during periods when the process variable is holding at setpoint indicates that the problem is not a loop oscillation, but rather the effects of process noise on the controller. The extremely high gain value of 24 is amplifying PV noise by that factor, and reproducing it on the output signal.

Next, a closed-loop "Ultimate" gain test was attempted on this process, but it was not successful. Even the controller's maximum possible gain value would not generate oscillations, due to the extremely crisp response of the process (minimal lag and dead times) and its integrating nature (constant phase shift of -90$^o$).

From the initial open-loop (manual output step-change) test, it can be seen this process was purely integrating. This means it can be controlled primarily by proportional action, with very little integral action required, and no derivative action whatsoever. The presence of some process noise is the only factor limiting the aggressiveness of proportional action. With this in mind, increasingly aggressive gain values were applied until the output signal noise was at a maximum acceptable

limit for the control valve (qualitatively determined). Then, integral action was added to ensure reasonable elimination of offset.

After some experimenting, the values arrived at were 3 (gain), 10 minutes per repeat (integral), and 0 minutes (derivative). These tuning values represent a proportional action only one-eighth as aggressive as the Ziegler-Nichols recommendation, and an integral action *fifty times* less aggressive than the Ziegler-Nichols recommendation. The results of these tuning values in automatic mode are shown in Figure 23.13.

Figure 23.13: Closed-loop setpoint step response of a liquid-level process with better tuning

The trend shows five 10% increasing setpoint value changes, with crisp response every time, followed by a single 50% decreasing setpoint step-change. In all cases, the process response clearly meets the criteria of rapid attainment of new setpoint values and no overshoot or oscillation.

If it was decided that the noise in the output signal was too much for the valve to handle over time, the gain value may be reduced further and (possibly) compensating for slow offset recovery with more aggressive integral action. A damping constant could also be added into either the level transmitter or the controller itself, so long as this added lag did not cause oscillation problems in the loop[10]. The best solution would be to find a way to isolate the level transmitter from noise, so that the process variable signal was much "quieter." Whether or not this is possible depends on the

[10]Be *very* careful with the addition of damping, since the oscillations it could create may not appear on the trend. Remember that the insertion of damping (low-pass filtering) in the PV signal is essentially an act of "lying" to the controller: telling the controller something that differs from the real, measured signal. If the PV trend shows this damped signal and not the "raw" signal from the transmitter, it is possible for the process to oscillate and the PV trend to be deceptively stable!

process and on the particular transmitter used.

### 23.3.3 Tuning a temperature process

This next simulated process is a temperature control process. Performing an open-loop test (two 10% increasing output step-changes, both made in manual mode) on this process resulted in the behavior in Figure 23.14.

Figure 23.14: Open-loop output step response of a temperature process

From the trend, the process appears to be self-regulating with a slow time constant (lag) and a substantial dead time. The reaction rate ($R$) on the first step-change is 30% over 30 minutes, or 1 percent per minute. Dead time ($L$) looks to be approximately 1.25 minutes. Following the Ziegler-Nichols recommendations for PID tuning based on these process characteristics (also including the 10% step-change magnitude $\Delta m$):

$$K_p = 1.2\frac{\Delta m}{RL} = 1.2\frac{10\%}{\frac{30\%}{30 \text{ min}}1.25 \text{ min}} = 9.6$$

$$\tau_i = 2L = (2)(1.25 \text{ min}) = 2.5 \text{ min}$$

$$\tau_d = 0.5L = (0.5)(1.25 \text{ min}) = 0.625 \text{ min}$$

Applying the PID values of 9.6 (gain), 2.5 minutes per repeat (integral), and 0.625 minutes (derivative) gave the result in Figure 23.15 when in automatic mode.

Figure 23.15: Closed-loop setpoint step response of a temperature process with open-loop Ziegler-Nichols tuning

As can be seen, the results are quite poor. The PV is still oscillating with a peak-to-peak amplitude of almost 20% from the last process upset at the time of the 10% downward SP change. Additionally, the output trend is rather noisy, indicating excessive amplification of process noise by the controller.

Next, the closed-loop, or "Ultimate" tuning method of Ziegler and Nichols was applied to this process. Eliminating both integral and derivative control actions from the controller, and experimenting with different gain (proportional) values until self-sustaining oscillations of consistent amplitude were obtained as shown in Figure 23.16 gave a gain value of 15.

Figure 23.16: Determination of ultimate values for Ziegler-Nichols tuning of a temperature process

From the trend in Figure 23.16, the ultimate period ($P_u$) is approximately 5.2 minutes in length. Following the Ziegler-Nichols recommendations for PID tuning based on these process characteristics:

$$K_p = 0.6K_u = (0.6)(15) = 9$$

$$\tau_i = \frac{P_u}{2} = \frac{5.2 \text{ min}}{2} = 2.6 \text{ min}$$

$$\tau_d = \frac{P_u}{8} = \frac{5.2 \text{ min}}{8} = 0.65 \text{ min}$$

These PID tuning values are quite similar to those predicted by the open loop ("Reaction Rate") method, so very similar results were seen as shown in Figure 23.17.

Figure 23.17: Closed-loop setpoint step response of a temperature process with closed-loop Ziegler-Nichols tuning

As expected, there is still excessive oscillation following a 10% setpoint change, as well as excessive "noise" in the output trend.

From the initial open-loop (manual output step-change) test, this process was self-regulating with a slow lag and substantial dead time. The self-regulating nature of the process demands at least some integral control action to eliminate offset, but too much will cause oscillation given the long lag and dead times. The existence of over 1 minute of process dead time also prohibits the use of aggressive proportional action. Derivative action, which is generally useful in overcoming lag times, will cause problems here by amplifying process noise. In summary, then, expect to use mild proportional, integral, *and* derivative tuning values in order to achieve good control with this process. Anything too aggressive will cause problems for this process.

After some experimenting, the values arrived at were 3 (gain), 5 minutes per repeat(integral), and 0.5 minutes (derivative). These tuning values represent a proportional action only one-third as aggressive as the Ziegler-Nichols recommendation, and an integral action about half as aggressive as the Ziegler-Nichols recommendation. The results of these tuning values in automatic mode are shown in Figure 23.18.

Figure 23.18: Closed-loop setpoint step response of a temperature process with closed-loop Ziegler-Nichols tuning

As can be seen, the system's response has almost no overshoot (with either a 10% setpoint change or a 15% setpoint change) and very little "noise" on the output trend. Response to setpoint changes is relatively crisp considering the naturally slow nature of the process: each new setpoint is achieved within about 7.5 minutes of the step-change.

# References


[1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," Transactions of the ASME, vol. 64, pp. 759–768, 1942. [Online]. Available: http://www.driedger.ca/Z-N/Ziegler.pdf

[2] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, Design of Feedback Control Systems, 4th ed. New York, NY: Oxford University Press, 2002.

[3] C. L. Phillips and R. D. Harbor, Feedback Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[4] J. R. Rowland, Linear Control Systems: Modeling, Analysis, and Design. New York, NY: John Wiley & Sons, 1986.

[5] R. C. Dorf and R. H. Bishop, Modern Control Systems, 10th ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2005.

[6] F. J. Hale, Introduction to Control System Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[7] A. P. Sage, Linear Systems Control. Champaign, IL: Matrix Publishers, 1978.

[8] M. Driels, Linear Control Systems Engineering. New York, NY: McGraw-Hill, 1995.

[9] W. J. Palm, Control Systems Engineering. New York, NY: John Wiley and Sons, 1986.

[10] S. C. Gupta and L. Hasdorff, Fundamentals of Automatic Control. New York, NY: John Wiley and Sons, 1970.

[11] A. F. D'Souza, Design of Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[12] T. E. Marlin, Process Control, 2nd ed. New York, NY: McGraw-Hill, 2000.

[13] B. A. Ogunnaike and W. H. Ray, Process Dynamics, Modeling, and Control. New York, NY: Oxford University Press, 1994.

[14] A. Tewari, Modern Control Design with MATLAB and Simulink. New York, NY: John Wiley and Sons, 2002.

[15] C. A. Smith and A. Corripio, Principles and Practice of Automatic Process Control, 3rd ed. New York, NY: John Wiley and Sons, 2006.

[16] M. Gopal, Control Systems: Principles and Design. Boston, MA: McGraw-Hill, 2008.

[17] A. O'Dwyer, "PI and PID controller tuning rules for time delay processes: a summary," Dublin Institute of Technology, Tech. Rep. AOD-00-01, 2000. [Online]. Available: http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.33.7556&rep=rep1&type=pdf

# Chapter 24

# Programmable Logic Controllers

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms:
2. Describe the construction of various types of PLCs
3. Identify PLC programs using the following languages: ladder logic, function block, structured text, and instruction list
4. Discuss the function and operation of Human-Machine Intefaces (HMIs)

Electromechanical relays are versatile and reliable devices, and it is possible to perform virtually any control function given enough relays and wiring. However, they are subject to wear due to their moving parts, and re-configuring a relay-based control system entails disconnecting and re-connecting wires: a tedious task at best. Solid-state logic gates may replace electromechanical relays to avoid problems of wear (as well as vastly increase speed of response), but once again re-configuration would require disconnecting and re-connecting many wires between the gates.

A very practical alternative to relay- or gate-based control logic circuits is to use a *programmable* computer to perform the same functions. A computer equipped with the necessary input and output peripheral circuits, configured to be easily programmed by technical personnel for virtually any logical task, is called a *programmable logic controller* or *PLC*.

## 24.1 PLC examples

Programmable logic controllers are essentially nothing more than special-purpose, industrial computers. As such, they are built far more ruggedly than an ordinary personal computer (PC), and designed to run extremely reliable operating system software[1]. PLCs as a rule do not contain hard disk drives, cooling fans, or any other components with moving parts. This is an intentional design decision, intended to maximize the reliability of the hardware in harsh industrial environments where the PLC chassis may be subjected to temperature extremes, vibration, humidity, and airborne particulates (dust, fibers, and/or fumes).

Large PLC systems consist of a rack into which circuit "cards" are plugged. These cards include processors, input and output (I/O) points, communications ports, and other functions necessary to the operation of a complete PLC system. Such "modular" PLCs may be configured differently according to the specific needs of the application. Individual card failures are also easier to repair in a modular system, since only the failed card need be replaced, not all the cards or the whole card rack.

Small PLC systems consist of a monolithic "brick" containing all processor, I/O, and communication functions. These PLCs are typically far less expensive than their modular cousins, but are also more limited in I/O capability and must be replaced as a whole in the event of failure.

The following photographs show several examples of real PLC systems, some modular and some monolithic. These selections are not comprehensive by any means, as there are many more manufacturers and models of PLC than those pictured. They do, however, represent some of the more common brands and models in current industrial use.

Figure 24.1 is a Siemens (Texas Instruments) 505 series PLC installed in a control panel of a municipal wastewater treatment plant. This is an example of a modular PLC, with individual processor, I/O, and communication cards plugged into a rack. Three racks appear in this photograph (two completely filled with cards, and the third only partially filled).

[1] There are such things as *soft PLCs*, which consist of special-purpose software running on an ordinary PC with some common operating system. Soft PLCs enjoy the high speed and immense memory capacity of modern personal computers, but do not possess the same ruggedness either in hardware construction or in operating system design. Their applications should be limited to non-critical controls where neither main process production nor safety would be jeopardized by a control system failure.

Figure 24.1: A Siemens 505 Series PLC

Figure 24.2 shows an Allen-Bradley (Rockwell) PLC-5 system used to monitor and control the operation of a large natural gas compressor. Two racks appear in this first photograph, with different types of I/O cards plugged into each rack.

Figure 24.2: An Allen-Bradley PLC-5 PLC

Both the Siemens (formerly Texas Instruments) 505 and Allen-Bradley (Rockwell) PLC-5 systems are considered "legacy" PLC systems by modern standards, the two systems in the previous photographs being about 20 years old each. It is not uncommon to find "obsolete" PLCs still

in operation, though. Given their extremely rugged construction and reliable design, these control systems may continue to operate without significant trouble for decades.

A newer model of PLC manufactured by Allen-Bradley is the SLC 500 series (often verbally referred to as the "Slick 500" series), which is also modular in design like the older PLC-5 system, although the racks and modules of the SLC 500 design are quite compact by comparison. The SLC 500 rack shown in Figure 24.3 has 7 "slots" for processor and I/O cards.

Figure 24.3: An Allen-Bradley SLC 500 7-slot rack with power supply, CPU, and I/O

The first three slots of this SLC 500 rack acre occupied by the processor card, an analog input card, and a discrete input card, from left to right. The next two slots are empty (revealing the circuit board and connectors for accepting new cards). The last two slots hold discrete output and analog output cards, respectively.

A nine-slot SLC 500 system used for controlling a high-purity water treatment system for a biopharmaceuticals manufacturing facility is shown in Figure 24.4.

Figure 24.4: An Allen-Bradley SLC 500 9-slot rack with power supply, CPU, and I/O

A modern PLC manufactured by Siemens appears in Figure 24.5, an S7-300, which is a different design of modular PLC. Instead of individual cards plugging into a rack, this modular PLC design uses individual modules plugging into each other on their sides to form a wider unit.

Figure 24.5: A Siemens S7-300 PLC

A modern PLC manufactured by Allen-Bradley (Rockwell Automation) is the ControlLogix system shown in Figure 24.6 used to control a cereal manufacturing process. The modular design of the ControlLogix system follows the more traditional scheme of individual cards plugged into a rack of fixed size[2].

[2]These modern versions of programmable controllers like the ControlLogix are often called *Programmable Automation Controllers (PACs)*. A PAC has the ability to perform combinational and sequential logic control, process control, and motion control, where a PLC is usually limited to logic control and a minimal amount of process control.

Figure 24.6: An Allen-Bradley ControlLogix rack

While the Siemens S7 and Allen-Bradley ControlLogix platforms represent large-scale, modular systems, there exist much smaller PLCs available for a fraction of the cost. One of the least expensive PLC on the market is the Koyo "CLICK" PLC series sold by AutomationDirect. The processor module (with eight discrete input and six discrete output channels built in) in Figure 24.7 is shown fitting in a person's hand.

Figure 24.7: A Koyo "Click" PLC

This is a semi-modular PLC design, with a minimum of input/output (I/O) channels built into the processor module, but having the capacity to accept multiple I/O modules plugged in to the side, much like the Siemens S7-300 PLC.

Other semi-modular PLCs expand using I/O cards that plug in to the base unit not unlike traditional rack-based PLC systems. The Koyo DirectLogic DL06 is a good example of this type of semi-modular PLC. Figure 24.8 shows the DL06 base unit with 20 discrete input channels and 16 discrete output channels accepting an analog input card (this particular card is designed to input signals from thermocouples to measure up to four channels of temperature) in one of its four available card slots.

Figure 24.8: A Koyo DL06 PLC

.

Some low-end PLCs are strictly monolithic, with no ability to accept additional I/O modules. This General Electric Series One PLC shown in Figure 24.9 was used to monitor a small-scale hydroelectric power generating station and is an example of a purely monolithic design, having no "expansion" slots to accept I/O cards.

Figure 24.9: A General Electric Series One PLC

## 24.2 Input/Output (I/O) capabilities

Every programmable logic controller must have some means of receiving and interpreting signals from real-world sensors such as switches, and encoders, and also be able to effect control over real-world control elements such as solenoids, valves, and motors. This is generally known as *input/output*, or *I/O*, capability. Monolithic ("brick") PLCs have a fixed amount of I/O capability built into the unit, while modular ("rack") PLCs use individual circuit board "cards" to provide customized I/O capability.

Some PLCs have the ability to connect to processor-less remote racks filled with additional I/O cards or modules, thus providing a way to increase the number of I/O channels beyond the capacity of the base unit. The connection from the main (host) PLC to remote I/O racks usually takes the form of a special digital network, which may span a great physical distance. An example of this architecture is shown in Figure 24.10.

Figure 24.10: Diagram of a PLC system consisting of a main rack and a remote I/O rack

Input/output capability for programmable logic controllers comes in three basic varieties: *discrete*, *analog*, and *network*; each type discussed in a following subsection.

### 24.2.1 Discrete I/O

A "discrete" data point is one with only two states *on* and *off*. Process switches, pushbutton switches, limit switches, and proximity switches are all examples of discrete sensing devices. In order for a PLC to be aware of a discrete sensor's state, it must receive a signal from the sensor through a *discrete input (DI)* channel. Inside the discrete input module is (typically) a light-emitting diode (LED) which will be energized when the corresponding sensing device turns on. Light from this LED shines on a photo-sensitive device such as a phototransistor inside the module, which in turn activates a *bit* (a single element of digital data) inside the PLC's memory. This opto-coupled arrangement makes each input channel of a PLC rather rugged, capable of isolating the sensitive computer circuitry of the PLC from transient voltage "spikes" and other electrical phenomena capable of causing damage.

Indicator lamps, solenoid valves, and motor starters (assemblies consisting of contactors and overload protection devices) are all examples of discrete control devices. In a manner similar to discrete inputs, a PLC connects to any number of different discrete final control devices through a *discrete output (DO)* channel[3]. Discrete output modules typically use the same form of opto-isolation to allow the PLC's computer circuitry to send electrical power to loads: the internal PLC circuitry drives an LED which then activates some form of photosensitive switching device. Alternatively, small electromechanical relays may be used to interface the PLC's output bits to real-world electrical control devices.

An important concept to master when working with DC discrete I/O is the distinction between *current-sourcing* and *current-sinking* devices. The terms "sourcing" and "sinking" refer to the direction of current (as denoted by conventional flow notation) into or out of a device's control wire[4]. A device sending (conventional flow) current out of its control terminal to some other device(s) is said to be *sourcing* current, while a device accepting (conventional flow) current into its control terminal is said to be *sinking* current.

To illustrate, Figure 24.11 shows a PLC output channel is *sourcing* current to an indicator lamp, which is *sinking* current to ground.

[3] I/O "channels" are often referred to as "points" in industry lingo. Thus, a "32-point input card" refers to an input circuit with 32 separate channels through which to receive signals from on/off switches and sensors.

[4] "Control wire" means the single conductor connecting switch and load as opposed to conductors directly common with either the positive or negative lead of the voltage source. The task of determining whether a device is sourcing or sinking current becomes much simpler by focusing attention on this one wire, noting the direction of conventional-flow current through it.

Figure 24.11: Diagram illustrating sinking vs. sourcing

These terms really only make sense when electric current is viewed from the perspective of conventional flow, where the positive terminal of the DC power supply is envisioned to be the "source" of the current, with current finding its way "down" to ground (the negative terminal of the DC power supply). In every circuit formed by the output channel of a PLC driving a discrete control device, or by a discrete sensing device driving an input channel on a PLC, one element in the circuit must be sourcing current while the other is sinking current.

One engineer has a charming way to describe sourcing and sinking: *blowing* and *sucking*. A device that sources current to another "blows" current toward the other device. A device that sinks current "sucks" current from the other device. These terms may be helpful in first mastering the distinction between sourcing and sinking despite (or perhaps because of!) their informal nature.

If the discrete device connecting to the PLC is not polarity-sensitive, either type of PLC I/O module will suffice. For example, the following diagrams show a mechanical limit switch connecting to a sinking PLC input (Figure 24.12) and to a sourcing PLC input (Figure 24.13), respectively.

Figure 24.12: Diagram of a PLC with a sinking input connected to a field device that is not polarity sensitive

Figure 24.13: Diagram of a PLC with a sourcing input connected to a field device that is not polarity sensitive

Note the differences in polarity and labeling between the sinking card's common terminal and the sourcing card's common terminal. On the "sinking" card, the input channel terminal is positive while the common ("Com") terminal is negative. On the "sourcing" card, the input channel terminal is negative while the common ("VDC") terminal is positive.

Some discrete sensing devices *are* polarity-sensitive, such as electronic proximity sensors containing transistor outputs. A "sourcing" proximity switch can only interface with a "sinking"

PLC input channel, and visa-versa, as shown in Figure 24.14.

Figure 24.14: Diagrams of PLC inputs connected to field devices that are polarity sensitive

In all cases, the "sourcing" device sends current *out of* its signal terminal while the "sinking" device takes current *into* its signal terminal.

Two photographs of a DC (sinking) discrete input module for an Allen-Bradley model SLC 500

PLC are shown in Figure 24.15: one with the plastic cover closed over the connection terminals, and the other with the plastic cover opened up for viewing the terminals. A legend on the inside of the cover shows the purpose of each screw terminal: eight input channels (numbered 0 through 7) and two redundant "DC Com" terminals for the negative pole of the DC power supply to connect:

(a) Door closed

(b) Door open showing terminals

Figure 24.15: A sinking SLC 500 DI card

A standard feature found on practically every PLC discrete I/O module is a set of LED indicators visually indicating the status of each bit (discrete channel). On the SLC 500 module, the LEDs appear as a cluster of eight numbered squares near the top of the module.

A photograph of discrete output terminals on another brand of PLC (a Koyo model DL06) in Figure 24.16 shows somewhat different labeling.

Figure 24.16: The DO terminals on a Koyo DL06 PLC

Here, each output channel terminal is designated with a letter/number code beginning with the letter "Y". Several "common" terminals labeled with "C" codes service clusters of output channels. In this particular case, each "common" terminal is common only to four output channels. With sixteen total output channels on this PLC, this means there are four different "common" terminals. While this may seem somewhat strange (why not just have one "common" terminal for all sixteen output channels?), it more readily permits different DC power supplies to service different sets of output channels.

Electrical polarity is not an issue with AC discrete I/O, since the polarity of AC reverses periodically anyway. However, there is still the matter of whether the "common" terminal on a discrete PLC module will connect to the *neutral* (grounded) or *hot* (ungrounded) AC power conductor.

Figure 24.17 shows a discrete AC output module for an Allen-Bradley SLC 500 PLC. PLC AC DO modules typically use bidirectional TRIACs as power switching devices rather than the unidirectional transistors customarily used by DC discrete output modules.

Figure 24.17: An AC DI module for a SLC 500 PLC

This particular eight-channel module provides two sets of TRIACs for switching power to AC loads, each set of four TRIACs receiving AC power from a "hot" terminal (VAC 1 or VAC 2), the other side of the load device being connected to the "neutral" (grounded) conductor of the AC power source.

Fortunately, the hardware reference manual supplied by the manufacturer of every PLC shows diagrams illustrating how to connect discrete input and output channels to field devices. Additionally, there will be current limitations for each channel (and often per group of channels or per card) that must be observed. The designer must always consult the manufacturer's information before designing circuits to connect to the I/O points of a PLC to ensure no damage occurs!

### 24.2.2 Analog I/O

In the early days of programmable logic controllers, processor speed and memory were too limited to support anything but discrete (on/off) control functions. Consequently, the only I/O capability found on early PLCs were discrete in nature[5]. Modern PLC technology, though, is powerful enough

[5]Some modern PLCs such as the Koyo "CLICK" are also discrete-only. Analog I/O and processing is significantly more complex to engineer and more expensive to manufacture than discrete control, and so low-end PLCs are more likely to lack analog capability.

to support the measurement, processing, and output of analog (continuously variable) signals.

All PLCs are digital devices at heart. Thus, in order to interface with an analog sensor or control device, some "translation" is necessary between the analog and digital worlds. Inside every analog input module is an *ADC*, or *Analog-to-Digital Converter*, circuit designed to convert an analog electrical signal into a multi-bit binary word. Conversely, every analog output module contains a *DAC*, or *Digital-to-Analog Converter*, circuit to convert the PLC's digital command words into analog electrical quantities.

Analog I/O (AI or AO) is commonly available for modular PLCs for many different analog signal types, including:

- Voltage (0 to 10 volt, 0 to 5 volt)
- Current (0 to 20 mA, 4 to 20 mA)
- Thermocouple (millivoltage)
- RTD (millivoltage)
- Strain gauge (millivoltage)

The following photographs in Figure 24.18 show two analog I/O cards for an Allen-Bradley SLC 500 modular PLC system, an analog input card (Figure 24.18a) and an analog output card (Figure 24.18b). Labels on the terminal cover doors indicate screw terminal assignments.

(a) An AI card (b) Door open showing terminals

Figure 24.18: SLC 500 analog I/O cards

### 24.2.3 Network I/O

Many different digital network standards exist for PLC communication, from PLC to PLC and between PLCs and field devices. One of the earliest digital protocols developed for PLC communication was *Modbus*, originally for the Modicon brand of PLC. Modbus was adopted by other PLC and industrial device manufacturers as a *de facto* standard[6], and remains perhaps the most universal digital protocol available for industrial digital devices today.

Another digital network standard developed by a particular manufacturer and later adopted as a *de facto* standard is *Profibus*, originally developed by Siemens.

For more information on networking standards used in PLC systems, refer to Chapter 4, specifically the sections on specific network standards such as Modbus and Profibus.

## 24.3 Logic programming

Although it seems each model of PLC has its own idiosyncratic standard for programming, there does exist an international standard for controller programming that most PLC manufacturers at least attempt to conform to. This is the IEC 61131-3 standard, which will be the standard presented in this chapter.

One should take solace in the fact that despite differences in the details of PLC programming from one manufacturer to another and from one model to another, the basic principles are largely the same. There exist much greater disparities between different general-purpose programming languages (e.g., C/C++, BASIC, FORTRAN, Pascal, Java, Ada, etc.) than between the programming languages supported by different PLCs, and this fact does not prevent computer programmers from being "multilingual." One engineer has personally written and/or analyzed programs for over a half-dozen different manufacturers of PLCs (Allen-Bradley, Siemens, SquareD, Koyo, Fanuc, Moore Products APACS and QUADLOG, and Modicon), with multiple PLC models within most of those brands, and has said the differences in programming conventions are insignificant. After learning how to program one model of PLC, it is quite easy to adapt to programming other makes and models of PLC. Even if learning to program a particular PLC that does not exactly conform to the IEC 61131-3 standard, every single principle discussed in this chapter is still applicable – the fundamental concepts are truly that universal.

The IEC 61131-3 standard specifies four distinct forms of programming language for industrial controllers and one programming construct:

- Ladder Diagram (LD), a graphical language that resembles relay ladder logic diagrams
- Structured Text (ST), a high-level text language similar to C
- Instruction List (IL), a low-level text language similar to assembly language
- Function Block Diagram (FBD), a graphical language similar to LabVIEW

[6] A "de facto" standard is one arising naturally out of legacy rather than by an premeditated agreement between parties. Modbus and Profibus networks are considered "de facto" standards because those networks were designed, built, and marketed by pioneering firms prior to their acceptance as standards for others to conform to. In Latin, *de facto* means "from the fact," which in this case refers to the fact of pre-existence: a standard agreed upon to conform to something already in popular use. By contrast, a standard intentionally agreed upon before its physical realization is a *de jure* standard (Latin for "from the law"). FOUNDATION Fieldbus is an example of a *de jure* standard, where a committee arrives at a consensus for a network design and specifications prior to that network being built and marketed by any firm.

- Sequential Function Chart (SFC), a program organizational construct used for sequencing events that have transitions

Not all programmable logic controllers support all language types, but nearly all of them support Ladder Diagram (LD), which will be the primary focus in this chapter.

Programming languages for many industrial devices are limited by design. One reason for this is *simplicity*: any programming language simple enough in structure for someone with no formal computer programming knowledge to understand is going to be limited in its capabilities. Another reason for programming limitations is *safety*: the more flexible and unbounded a programming language is, the more potential there will be to unintentionally create complicated "run-time" errors when programming. The ISA safety standard number 84 classifies industrial programming languages as either *Fixed Programming Languages* (FPL), *Limited Variability Languages* (LVL), or *Full Variability Languages* (FVL). Ladder Diagram and Function Block Diagram programming are both considered to be "limited variability" languages, whereas Instruction List (and traditional computer programming languages such as C/C++, FORTRAN, BASIC, etc.) are considered "full variability" languages with all the attendant potential for complex errors.

### 24.3.1 Relating I/O status to virtual elements

Perhaps the most important yet elusive concept to grasp when learning to program PLCs is the relationship between the electrical status of the PLC's I/O points and the status of variables and other "elements" in its programming. This is especially true for Ladder Diagram (LD) programming, where the program itself resembles an electrical diagram. Making the mental connection between the "real" world of the switches, contactors, and other electrical devices connected to the PLC and the "imaginary" world of the PLC's program consisting of virtual contacts and relay "coils" is most fundamental.

The first fundamental rule one should keep in mind when examining a Ladder Diagram PLC program is that **each virtual contact shown in the program *actuates* whenever it reads a "1" state in its respective bit and will be *at rest* whenever it reads a "0" state in its respective bit** (in the PLC's memory). If the contact is a normally-open (NO) type, it will open when its bit is 0 and close when its bit is 1. If the contact is a normally-closed (NC) type, it will close when its bit is 0 and open when its bit is 1. A 0 bit state causes the contact to be in its "normal" (resting) condition, while a 1 bit state *actuates* the contact, forcing it into its non-normal (actuated) state.

Another rule to remember when examining a Ladder Diagram PLC program is that the programming software offers *color highlighting*[7] to display the virtual status of each program element: **a colored contact is *closed*, while an un-colored contact is *open*.** While the presence or absence of a "slash" symbol marks the *normal* status of a contact, its live color highlighting shown by PLC programming software reveals the "conductive" status of the elements *in real time*.

Figure 24.19 shows how the two types of contacts in a PLC's Ladder Diagram program respond to bit states, using red coloring to signify each contact's virtual conductivity.

[7] It should be noted that in some situations the programming software will fail to color the contacts properly, especially if their status changes too quickly for the software communication link to keep up, and/or if the bit(s) change state multiple times within one scan of the program. However, for simple programs and situations, this rule holds true and is a great help to beginning programmers as they learn the relationship between real-world conditions and conditions within the PLC's "virtual" world.

Figure 24.19: PLC virtual switches and the corresponding bit states

A PLC's virtual contact is actuated by a high *bit* condition (1). In the context of any switch, an *actuated* condition is the opposite of its *normal* (resting) condition.

The simplified[8] illustration in Figure 24.20 shows a small PLC with two of its discrete input channels electrically energized, causing those two bits to have "1" statuses. The color-highlighted contacts in the programming editor software's display shows a collection of contacts addressed to those input bits in various states (colored = closed ; un-colored = open). As you can see, every contact addressed to a "set" bit (1) is in its actuated state, while every contact addressed to a "cleared" bit (0) is in its normal state.

[8]The electrical wiring shown in this diagram is incomplete, with the "Common" terminal shown unconnected for simplicity's sake.

Figure 24.20: A PLC and its corresponding program illustrating switch states and bit values in the data table

Remember that a *colored* contact is a *closed* contact. The contacts appearing as colored are either normally-closed contacts with "0" bit states, or normally-open contacts with "1" bit states. It is the combination of bit state and contact type (NO vs. NC) that determines whether the virtual contact will be open (un-colored) or closed (colored) at any given time. Correspondingly, it is a combination of colored highlighting and virtual contact type that indicates the real-world energization status of a particular PLC input at any given time.

The main problem students have comprehending PLC Ladder Diagram programs is that they over-simplify and try to directly associate real-world switches connected to the PLC with their respective contact instructions inside the PLC program. Students mistakenly think the real-world switch connecting to the PLC and the respective virtual switch contact inside the PLC program are one and the same, when this is not the case at all. Rather, the real-world switch sends power to the PLC input, *which in turn controls the state of the virtual contact(s) programmed into the PLC.* Specifically, students routinely fall into the following misconceptions:

- Students mistakenly think the contact instruction type (NO vs. NC) needs to match that of

its associated real-world switch

- Students mistakenly think color highlighting of a contact instruction is equivalent to the electrical status of its associated real-world PLC input
- Students mistakenly think a closed real-world switch must result in a closed contact instruction in the live PLC program

To clarify, here are the fundamental rules one should keep in mind when interpreting contact instructions in Ladder Diagram PLC programs:

- **Each input bit in the PLC's memory will be a "1" when its input channel is powered, and will be a "0" when its input channel is unpowered**
- **Each virtual contact shown in the program *actuates* whenever it reads a "1" state in its respective bit, and will be *at rest* whenever it reads a "0" state in its respective bit**
- **A colored contact is *closed* (passes virtual power in the PLC program), while an un-colored contact is *open* (blocks virtual power in the PLC program)**

In trying to understand PLC Ladder Diagram programs, the importance of these rules cannot be overemphasized. The truth of the matter is a causal chain – rather than a direct equivalence – between the real-world switch and the contact instruction status. The real-world switch controls whether or not electrical power reaches the PLC input channel, which in turn controls whether the input register bit will be a "1" or a "0", which in turn controls whether the contact instruction will actuated or at rest. Virtual contacts inside the PLC program are thus *controlled* by their corresponding real-world switches, rather than simply being *identical* to their real-world counterparts as novices tend to assume. Following these rules, we see that normally-open (NO) contact instructions will mimic what their real-world switches are doing, while normally-closed (NC) contact instructions will act opposite of their real-world counterparts.

The color highlighting of *coil* instructions in a Ladder Diagram PLC program follows similar rules. A coil will be "on" (colored) when all contact instructions prior to it are closed (colored). A colored coil writes a "1" to its respective bit in memory, while an un-colored coil instruction writes a "0" to its respective bit in memory. If these bits are associated with real-world discrete output channels on the PLC, their states will control the real-world energization of devices electrically connected to those channels.

To illustrate, examine the operation of a simple PLC system designed to illuminate a warning lamp in the event that a process vessel experiences a high fluid pressure. The PLC's task is to energize a warning lamp if the process vessel pressure ever exceeds 270 PSI, and keep that warning lamp energized even if the pressure falls below the trip point of 270 PSI. This way, operators will be alerted to both *past* and *present* process vessel over-pressure events. The controller, field devices, and program are shown in Figure 24.21.

The 120 volt AC "line" power (L1 and L2) provides electrical energy for the PLC to operate, as well as signal potential for the input switches and power for the warning lamp. Two switches connect to the input of this PLC: one normally-open pushbutton switch acting as the alarm reset (pressing this switch "unlatches" the alarm lamp) and one normally-open pressure switch acting as the sensing element for high process vessel pressure.

Figure 24.21: A PLC, field devices, and ladder diagram illustrating the program

The reset pushbutton connects to discrete input X1 of the PLC, while the pressure switch connects to discrete input X4. The warning lamp connects to discrete output Y5. Red indicator LEDs next to each I/O terminal visually indicate the electrical status of the I/O points, while red-shaded highlighting shows the *virtual power*[9] status of the "contacts" and "coils" in the PLC's program, displayed on the screen of a personal computer connected to the PLC through a programming cable.

With no one pressing the reset pushbutton, that switch will be in its normal status, which for a "normally-open" switch is open. Likewise with the pressure switch: with process pressure less than the trip point of 270 PSI, the pressure switch will also be in its normal status, which for a "normally-open" switch is open. Since neither switch is conducting electricity right now no voltage is present on either of the discrete inputs, so neither discrete input X1 nor X4 will be energized. This means the "virtual" contacts inside the PLC program will likewise be in their own normal states.

[9]For a PLC program contact, the shading represents virtual "conductivity." For a PLC program coil, the shading represents virtual "power" to "energize" that coil.

Thus, any virtual contact drawn as a normally-open will be open (not passing virtual power), and any virtual contact drawn as a normally-closed (a diagonal slash mark through the contact symbol) will be closed. This is why the two normally-open virtual contacts X4 and Y5 have no highlighting, but the normally-closed virtual contact X1 does – the colored highlighting representing the ability to pass virtual power.

If the process vessel experiences a high pressure (> 270 PSI), the pressure switch will actuate, closing its normally-open contact. This will energize input X4 on the PLC, which will "close" the virtual contact X4 in the ladder program. This sends virtual power to the virtual "coil" Y5, which in turn latches itself on through virtual contact Y5 and also energizes the real discrete output Y5 to energize the warning lamp as shown in Figure 24.22.

Figure 24.22: The PLC system with a high pressure condition

If now the process pressure falls below 270 PSI, the pressure switch will return to its normal state (open), thus de-energizing discrete input X4 on the PLC. Because of the latching contact Y5 in the PLC's program, however, output Y5 remains on to keep the warning lamp in its energized

state as seen in Figure 24.23.

Figure 24.23: The PLC system with the high pressure alarm latched

The only way for a human operator to re-set the warning lamp is to press the pushbutton. This will have the effect of energizing input X1 on the PLC, thus opening virtual contact X1 (normally-closed) in the program, thus interrupting virtual power to the virtual coil Y5, thus powering down the warning lamp and un-latching virtual power in the program as seen in Figure 24.24.

Figure 24.24: The PLC system with the high pressure alarm reset

### 24.3.2 The PLC scan

The central processor unit (CPU) in a PLC operates continuously in a loop-like fashion unless a major fault causes it to halt in a process known as *scanning*. The basic scan contains multiple typical steps: read inputs and update the input data table, execute the logic, update the output data table and turn on outputs, send and receive communications, and execute some "housekeeping" programming. It then goes back to the beginning and begins again.

Here are some examples of scans for various types of PLCs:

Allen-Bradley SLC 500 & Micrologix:

1. Read inputs
2. Execute program and store the resulting outputs for each rung in memory
3. Write outputs

4. Communications
5. Overhead tasks (memory management)

Siemens S7-200:
1. Read inputs
2. Execute program and store the resulting outputs for each rung in memory
3. Communications
4. CPU self-diagnostics
5. Write outputs

Koyo "CLICK":
1. Read inputs
2. Service peripherals
3. Update special registers
4. Execute program and store the resulting outputs for each rung in memory
5. Write outputs
6. CPU self-diagnostics

Note, however, that many modern PLCs have moved to asynchronous updating of I/O and may require buffering I/O data to prevent issues with I/O values changing during a single scan[10]. A good example is the Allen-Bradley ControlLogix platform, which has a separate backplane CPU that handles I/O asynchronously from the scan performed by the logic CPU. Also, even legacy PLCs have "immediate" contact and coil (read and write) instructions that will update input and output data tables as soon as they are executed without waiting for the proper read or write portions of the scan.

### 24.3.3 Memory maps and I/O addressing

Every discrete input and output channel on a PLC is linked to a corresponding *bit* in the PLC's digital memory. Similarly, every analog input and output point on a PLC is linked to a corresponding *word* (a block of bits) in the PLC's memory. The association between I/O points and memory locations differs between manufacturers and models of PLC, sometimes radically. Learning how the memory of any particular computer is organized – for I/O as well as for other purposes – is one of the first things any programmer should become familiar with when learning how to work with any computer system, PLCs included.

Every PLC manufacturer publishes reference manuals providing this information, usually in the form of a chart known as a *memory map*. A "memory map" shows the organization of digital memory into different sections, reserved for different uses. The following table shows a partial memory map for an Allen-Bradley SLC 500 PLC[11]:

[10] The speed of modern CPUs used in PLCs make buffering unnecessary for most programs, but it may be needed from time to time

[11] Also called the *data table*, this map shows the addressing of memory areas reserved for programs entered by the user. Other areas of memory exist within the SLC 500 processor, but these other areas are inaccessible to the technician or engineer writing PLC programs.

| File number | File type | Logical address range |
|---|---|---|
| 0 | Output image | `O:0` to `O:30` |
| 1 | Input image | `I:0` to `I:30` |
| 2 | Status | `S:0` to `S:`$n$ |
| 3 | Binary | `B3:0` to `B3:255` |
| 4 | Timers | `T4:0` to `T4:255` |
| 5 | Counters | `C5:0` to `C5:255` |
| 6 | Control | `R6:0` to `R6:255` |
| 7 | Integer | `N7:0` to `N7:255` |
| 8 | Floating-point | `F8:0` to `F8:255` |
| 9 | Network | `x9:0` to `x9:255` |
| 10 through 255 | User defined | `x10:0` to `x255:255` |

Note that Allen-Bradley's use of the word "file" differs from modern parlance. In the SLC 500 controller, a "file" is a block of random-access memory used to store a particular type of data. By contrast, a "file" according to modern convention is a contiguous collection of data bits with collective meaning (e.g., a word processing file or a spreadsheet file). Within each of the Allen-Bradley PLC's "files" are multiple "elements," each element consisting of a set of bits (8, 16, 24, or 32) representing data. Elements are addressed by number following the colon after the file designator, and individual bits within each element addressed by a number following a slash mark. For example, the first bit (bit 0) of the second element in file 3 (Binary) would be addressed as `B3:2/0`.

In Allen-Bradley PLCs such as the SLC 500 and PLC-5 models, files 0, 1, and 2 are exclusively reserved for discrete outputs, discrete inputs, and status bits, respectively. Thus, the letter designators O, I, and S (file types) are redundant to the numbers 0, 1, and 2 (file numbers). Other file types such as B (binary), T (timers), C (counters), and others have their own default file numbers (3, 4, and 5, respectively), but may also be used in some of the user-defined file numbers (10 and above). For example, file 7 in an Allen-Bradley controller is reserved for data of the "integer" type (N), but integer data may also be stored in file 15 at the user's discretion. Thus, file numbers and file type letters for data types other than output (O), input (I), and status (S) always appear together. You would not typically see an integer word addressed as `N:30` (integer word 30 in the PLC's memory) for example, but rather as `N7:30` (integer word 30 *in file 7* of the PLC's memory) to distinguish it from other integer word 30's that may exist in other files of the PLC's memory.

A hallmark of the SLC 500's addressing scheme common to many legacy PLC systems is that the address labels for input and output bits explicitly reference the physical locations of the I/O channels. For instance, if an 8-channel discrete input card were plugged into slot 4 of an Allen-Bradley SLC 500 PLC, and you wished to specify the second bit (bit 1 out of a 0 to 7 range), you would address it with the following label: `I:4/1`. Addressing the seventh bit (bit number 6) on a discrete output card plugged into slot 3 would require the label `O:3/6`. In either case, the numerical structure of that label tells you exactly where the real-world input signal connects to the PLC.

To illustrate the relationship between physical I/O and bits in the PLC's memory, consider the example of an Allen-Bradley SLC 500 PLC in Figure 24.25, which shows one of its discrete input channels energized (the switch being used as a "Start" switch for an electric motor).

Figure 24.25: A DI energized on an Allen-Bradley SLC 500

If an input or output card possesses more than 16 bits – as in the case of the 32-bit discrete output card shown in slot 3 of the example SLC 500 rack – the addressing scheme further subdivides each element into *words* and bits (each "word" being 16 bits in length). Thus, the address for bit number 27 of a 32-bit input module plugged into slot 3 would be `I:3.1/11` (since bit 27 is equivalent to bit 11 of word 1 – word 0 addressing bits 0 through 15 and word 1 addressing bits 16 through 31):

Figure 24.26: A DO energized on an Allen-Bradley SLC 500

Legacy PLC systems typically reference each one of the I/O channels by labels such as "I:1/3" (or equivalent[12]) indicating the actual location of the input channel terminal on the PLC unit. The IEC 61131-3 programming standard refers to this channel-based addressing of I/O data points as *direct addressing*. A synonym for direct addressing is *absolute addressing*.

Addressing I/O bits directly by their card, slot, and/or terminal labels may seem simple and elegant, but it becomes very cumbersome for large PLC systems and complex programs. Every time

[12]Some systems such as the Texas Instruments 505 series used "X" labels to indicate discrete input channels and "Y" labels to indicate discrete output channels (e.g., input `X9` and output `Y14`). This same labeling convention is still used by Koyo in its DirectLogic and "CLICK" PLC models. Siemens continues a similar tradition of I/O addressing by using the letter "I" to indicate discrete inputs and the letter "Q" to indicate discrete outputs (e.g., input channel `I0.5` and output `Q4.1`).

a programmer views the program, they must "translate" each of these I/O labels to some real-world device (e.g., "Input `I:1/3` is actually the *Start* pushbutton for the middle tank mixer motor") in order to understand the function of that bit. A later effort to enhance the clarity of PLC programming was the concept of addressing variables in a PLC's memory by arbitrary names rather than fixed codes. The IEC 61131-3 programming standard refers to this as *symbolic addressing* in contrast to "direct" (channel-based) addressing, allowing programmers arbitrarily name I/O channels in ways that are meaningful to the system as a whole. To use our simple motor "Start" switch example, it is now possible for the programmer to designate input `I:1/3` (an example of a *direct address*) as "`Motor_start_switch`" (an example of a *symbolic address*) within the program, thus greatly enhancing the readability of the PLC program. Initial implementations of this concept maintained direct addresses for I/O data points, with symbolic names appearing in addition to those absolute addresses as mere supplements for program readability. One PLC manufacturer (Koyo) still does this, calling the symbolic labels *nicknames*.

The modern trend in PLC addressing is to avoid the use of direct addresses such as `I:1/3` altogether, so they do not appear anywhere in the programming code. The Allen-Bradley "Logix" series of programmable logic controllers is the most prominent example of this new convention at the time of this writing. Each I/O point, regardless of type or physical location, is assigned a *tag name* which is meaningful in a real-world sense, and these tag names (or *symbols* as they are alternatively called) are referenced to absolute I/O channel locations by a database file. An important requirement of tag names is that they contain no space characters between words (e.g., instead of "`Motor start switch`", a tag name should use hyphens or underscore marks as spacing characters: "`Motor_start_switch`"), since spaces are generally assumed by computer programming languages to be delimiters (separators between different variables).

This modern convention of symbolic addressing will be continued for the remainder of the chapter in the sections on PLC programming. Each data point within the PLC programs will bear its own tag name rather than a direct (channel-based) address code.

## 24.4 Ladder Diagram (LD) programming

In the United States, the most common language used to program PLCs is *Ladder Diagram* (LD), also known as *Relay Ladder Logic* (RLL). This is a graphical language showing the logical relationships between inputs and outputs as though they were contacts and coils in a hard-wired electromechanical relay circuit. This language was invented for the express purpose of making PLC programming feel "natural" to electricians familiar with relay-based logic and control circuits. While Ladder Diagram programming has many shortcomings, it remains extremely popular and so will be the primary focus of this chapter.

Every Ladder Diagram program is arranged to resemble an electrical diagram, making this a graphical (rather than text-based) programming language. Ladder diagrams are to be thought of as *virtual circuits*, where virtual "power" flows through virtual "contacts" (when closed) to energize virtual "relay coils" to perform logical functions. None of the contacts or coils seen in a Ladder Diagram PLC program are real; rather, they act on bits in the PLC's memory, the logical interrelationships between those bits expressed in the form of a diagram *resembling* a circuit.

The computer screenshot in Figure 24.27 shows a typical Ladder Diagram program[13] being edited

[13] This particular program and editor is for the Koyo "CLICK" series of micro-PLCs.

on a personal computer.

Figure 24.27: A DI energized on an Allen-Bradley SLC 500

Contacts appear just as they would in an electrical relay logic diagram – as short vertical line segments separated by a horizontal space. Normally-open contacts are empty within the space between the line segments, while normally-closed contacts have a diagonal line crossing through that space. Coils are somewhat different, appearing as either circles or pairs of parentheses. Other instructions appear as rectangular boxes.

Each horizontal line is referred to as a *rung*, just as each horizontal step on a stepladder is called a "rung." A common feature among Ladder Diagram program editors, as seen on this screenshot, is the ability to color-highlight those virtual "components" in the virtual "circuit" ready to "conduct" virtual "power." In this particular editor, the color used to indicate "conduction" is light blue. Another form of status indication seen in this PLC program is the values of certain variables in the PLC's memory, shown in red text.

For example, you can see coil `T2` energized at the upper-right corner of the screen (filled with light blue coloring), while coil `T3` is not. Correspondingly, each normally-open `T2` contact appears colored, indicating its "closed" status, while each normally-closed `T2` contact is uncolored. By contrast, each

normally-open `T3` contact is uncolored (since coil `T3` is unpowered) while each normally-closed `T3` contact is shown colored to indicate its conductive status. Likewise, the current count values of timers `T2` and `T3` are shown as 193 and 0, respectively. The output value of the math instruction box happens to be 2400, also shown in red text.

Color-highlighting of Ladder Diagram components only works, of course, when the computer running the program editing software is connected to the PLC and the PLC is in the "run" mode (and the "show status" feature of the editing software is enabled). Otherwise, the Ladder Diagram is nothing more than black symbols on a white background. Not only is status highlighting very useful in de-bugging PLC programs, but it also serves an invaluable diagnostic purpose when an engineer or technician analyzes a PLC program to check the status of real-world input and output devices connected to the PLC. This is especially true when the program's status is viewed remotely over a computer network, allowing maintenance staff to investigate system problems without even being near the PLC!

### 24.4.1 Contacts and coils

The most elementary objects in Ladder Diagram programming are *contacts* and *coils*, intended to mimic the contacts and coils of electromechanical relays. Contacts and coils are *discrete* programming elements, dealing with Boolean (1 and 0; on and off; true and false) variable states. Each contact in a Ladder Diagram PLC program represents the *reading* of a single bit in memory, while each coil represents the *writing* of a single bit in memory.

Discrete input signals to the PLC from real-world switches are read by a Ladder Diagram program by contacts referenced to those input channels. In legacy PLC systems, each discrete input channel has a specific address which must be applied to the contact(s) within that program. In modern PLC systems, each discrete input channel has a tag name created by the programmer which is applied to the contact(s) within the program. Similarly, discrete output channels – referenced by coil symbols in the Ladder Diagram – must also bear some form of address or tag name label.

To illustrate, imagine the construction and programming of a redundant flame-sensing system to monitor the status of a burner flame using three sensors. The purpose of this system will be to indicate a "lit" burner if at least two out of the three sensors indicate flame. If only one sensor indicates flame (or if no sensors indicate flame), the system will declare the burner to be un-lit[14]. The burner's status will be visibly indicated by a lamp that human operators can readily see inside the control room area.

The example system's wiring is shown in Figure 24.28.

[14]This is known as a two-out-of-three (2oo3) voting system. A single failure will neither cause, nor prevent, an action from occuring.

Figure 24.28: A flame-sensing system example

Each flame sensor outputs a DC voltage signal indicating the detection of flame at the burner, either on (24 volts DC) or off (0 volts DC). These three discrete DC voltage signals are sensed by the first three channels of the PLC's discrete input card. The indicator lamp is a 120 volt light bulb, and so must be powered by an AC discrete output card, shown here in the PLC's last slot.

To make the ladder program more readable, tag names (symbolic addresses) will be assigned to each input and output bit in the PLC, describing its real-world device in an easily-interpreted format[15]. The first three discrete input channels will be tagged as `IN_sensor_A`, `IN_sensor_B`, and `IN_sensor_C`, and the output as `OUT_burner_lit`.

[15] If this were a legacy Allen-Bradley PLC system using absolute addressing, the three sensor inputs would be forced to be addressed as `I:1/0`, `I:1/1`, and `I:1/2` (slot 1, channels 0 through 2), and the indicator lamp output as `O:2/0` (slot 2, channel 0). If this were a newer Logix5000 Allen-Bradley PLC, the default tag names would be `Local:1:I.Data.0`, `Local:1:I.Data.1`, and `Local:1:I.Data.2` for the three inputs, and `Local:2:O.Data.0` for the output. However, both systems allow symbolic addresses to be assigned so there a way to reference the I/O channels without having to rely on these cumbersome labels. The programs showing in this book exclusively use tag names rather than absolute addresses, since this is the more modern programming convention.

A ladder program to determine if at least two out of the three sensors detect flame is shown in Figure 24.29, with the tag names referencing each contact and coil.

Figure 24.29: The LD program implementing 2oo3 voting logic for a flame-sensing system

Series-connected contacts in a Ladder Diagram perform the logical `AND` function, while parallel contacts perform the logical `OR` function. Thus, this two-out-of-three flame-sensing program could be verbally described as:

"Burner is lit if either `A` *and* `B`, *or* either `B` *and* `C`, `or` `A` *and* `C` are TRUE"

An alternate way to express this is to use the notation of *Boolean algebra*, where multiplication represents the `AND` function and addition represents the `OR` function:

$$\text{Burner_lit} = AB + BC + AC$$

To illustrate how this program would work, we will consider a case where flame sensors B and C detect flame, but sensor A does not. This represents a two-out-of-three condition, and so we would expect the PLC to turn on the "Burner lit" indicator light as programmed. From the perspective of the PLC's rack, we would see the indicator LEDs for sensors B and C lit up on the discrete input card, as well as the indicator LED for the lamp's output channel, as shown in Figure 24.30.

Figure 24.30: The results of 2oo3 voting logic for a flame-sensing system with two sensors activated

Those two energized input channels "set" bits (logic 1 status) in the PLC's memory representing the status of flame sensors B and C. Flame sensor A's bit will be "clear" (logic 0 status) because its corresponding input channel is de-energized. The fact that the output channel LED is energized (and the "Burner lit" indicator lamp is energized) tells us the PLC program has "set" that corresponding bit in the PLC's output memory register to a "1" state.

A display of input and output register bits shows the "set" and "reset" states for the PLC at this moment in time as seen in Figure 24.31.

Figure 24.31: The memory registers showing the results of 2oo3 voting logic for a flame-sensing system with two sensors activated

Examining the Ladder Diagram program with status indication enabled in Figure 24.32 shows how just one of the series-connected contact pairs are passing "virtual power" to the output coil.

Figure 24.32: The ladder diagram showing the results of 2oo3 voting logic for a flame-sensing system with two sensors activated

Recall that the purpose of a contact in a PLC program is to *read* the status of a bit in the PLC's memory. These six "virtual contacts" read the three input bits corresponding to the three flame sensors. Each normally-open "contact" will "close" when its corresponding bit has a logic value of 1, and will "open" (go to the normal state) when its corresponding bit has a logic value of 0. Thus, we see here that the two contacts corresponding to sensor A appear without highlighting (representing no "conductivity" in the virtual relay circuit) because the bit for that input is reset (0). The two contacts corresponding to sensor B and the two contacts corresponding to sensor C all appear highlighted (representing "conductivity" in the virtual circuit) because their bits are both set (logic 1).

Recall also that the purpose of a coil in a PLC program is to *write* the status of a bit in the

PLC's memory. Here, the "energized" coil sets the bit for the PLC Output 0 to a logic 1 state, thus activating the real-world output and sending electrical power to the "Burner lit" lamp.

Note that the color highlighting does *not* indicate a virtual contact is *conducting* virtual power, but merely that it is *able* to conduct power. Color highlighting around a virtual coil, however, *does* indicate the presence of virtual "power" at that coil.

Contacts and relays are not just useful for implementing simple logic functions, but they may also perform *latching* functions as well. A very common application of this in industrial PLC systems is a latching start/stop program for controlling electric motors by means of momentary-contact pushbutton switches. As before, this functionality will be illustrated by means of an hypothetical example circuit shown in Figure 24.33.

Figure 24.33: A motor control circuit with a PLC used instead of hard-wired relay ladder logic

In this system, two pushbutton switches (a NO "Start" and a NC "Stop") are connected to discrete inputs on a PLC, and the PLC in turn energizes the coil of a motor contactor relay by means of one of its discrete outputs[16]. An overload contact is wired directly in series with the contactor coil to provide motor overcurrent protection, even in the event of a PLC failure where the

[16]The particular input and output channels chosen for this example are completely arbitrary. There is no particular reason to choose input channels 6 and 7, or output channel 2, as I have shown in the wiring diagram. Any available I/O channels will serve the purpose quite adequately.

discrete output channel remains energized[17].

The ladder diagram program for this motor control system is shown in Figure 24.34.

Figure 24.34: A motor control circuit PLC program

Pressing the "Start" pushbutton energizes discrete input channel 6 on the PLC, which "closes" the virtual contact in the PLC program labeled `IN_switch_Start`. The normally-closed virtual contact for input channel 7 (the "Stop" pushbutton) is already closed by default when the "Stop" button is not being pressed, and so the virtual coil will receive "power" when the "Start" pushbutton is pressed and the "Stop" pushbutton is not.

Note the *seal-in* contact bearing the exact same label as the coil: `OUT_contactor`. At first it may seem strange to have both a contact and a coil in a PLC program labeled identically[18], since contacts are most commonly associated with inputs and coils with outputs, but this makes perfect sense if you realize the true meaning of contacts and coils in a PLC program: as *read* and *write* operations on bits in the PLC's memory. The coil labeled `OUT_contactor` *writes* the status of that bit, while the NO contact labeled `OUT_contactor` *reads* the status of that same bit. The purpose of this contact, of course, is to latch the motor in the "on" state after a human operator has released his or her finger from the "Start" pushbutton.

This programming technique is known as *feedback*, where an output variable of a function (in this case, the feedback variable is `OUT_contactor`) is also an input to that same function. The path of feedback is *implicit* rather than *explicit* in Ladder Diagram programming, with the only indication of feedback being the common name shared by coil and contact. Other graphical programming languages (such as Function Block) have the ability to show feedback paths as connecting lines between function outputs and inputs, but this capacity does not exist in Ladder Diagram.

A step-by-step sequence showing the operation and status of this simple program illustrates how the seal-in contact functions through a start-up and shut-down cycle of the motor is shown in Figure 24.35.

[17]While it is possible to wire the overload contact to one of the PLC's discrete input channels and then program a *virtual* overload contact in series with the output coil to stop the motor in the event of a thermal overload, this strategy would rely on the PLC to perform a safety function which is better performed by hard-wired circuitry (unless a special kind of PLC called a *safety PLC* is used.

[18]A very common misconception among students first learning PLC Ladder Diagram programming is to always associate contacts with PLC inputs and coils with PLC outputs, thus it seems weird to have a contact bear the same label as an output. However, this is a false association. In reality, contacts and coils are *read* and *write* instructions, and thus it is possible to have the PLC read one of its own output bits as a part of some logic function. What *would* be truly strange is to label a coil with an input bit address or tag name, since the PLC is not electrically capable of setting the real-world energization status of any input channels.

Figure 24.35: The sequence of events in the motor control PLC program

This sequence helps illustrate the *order of evaluation* or *scan order* of a Ladder Diagram program. The PLC reads a Ladder Diagram from left to right, top to bottom, in the same general order as a

human being reads sentences and paragraphs written in English. Step 2 indicates the `OUT_contactor` coil "immediately activates" after the operator pushes the "Start" switch. However, according to the IEC 61131-3 standard, a PLC program must evaluate (read) all inputs (contacts) to a function before determining the status of a function's output (coil or coils). In other words, the PLC does not make any decision on how to set the state of a coil until all contacts providing power to that coil have been read, and then it only stores the result of a rung in memory until the scan updates the output data table. First, the inputs are scanned and activate the `IN_switch_Start` bit in the input data table, then the scan executes the logic, and finally it updates the output data table to activate the `OUT_contactor` coil. However, note that the NO contact associated with the `OUT_contactor` coil does not activate in Step 2. This is correct, as the `OUT_contactor` contacts are located in the input data table, and the scan has not come back around to updating the input data table yet. Once a coil's status has been written to memory, any contacts bearing the same tag name will update with that status during the next scan. In Step 3, the scan has updated the input data table with the status of all internal and external inputs and the `OUT_contactor` NO contract becomes true to seal in around the `IN_switch_Start` NO contact.

Step 5 in the sequence is also illustrative of this point. When the human operator presses the "Stop" pushbutton, the input channel for `IN_switch_Stop` becomes activated, which "opens" the normally-closed virtual contact `IN_switch_Stop`. Upon the next scan of this program rung, the PLC evaluates all input contacts (`IN_switch_Start`, `IN_switch_Stop`, and `OUT_contactor`) to check their status before deciding what status to write to the `OUT_contactor` coil. Seeing that the `IN_switch_Stop` contact has been forced open by the activation of its respective discrete input channel, the PLC writes a "0" (or "False") state to the `OUT_contactor` coil. However, the `OUT_contactor` feedback contact does not update until the next scan, which is why you still see it highlighted in blue during step 5.

A potential problem with this system as it is designed is that the human operator loses control of the motor in the event of an "open" wiring failure in either pushbutton switch circuit. For instance, if a wire fell off a screw contact for the "Start" pushbutton switch circuit, the motor could not be started if it was already stopped. Similarly, if a wire fell off a screw contact for the "Stop" pushbutton switch circuit, the motor could not be stopped if it was already running. In either case, a broken wire connection acts the same as the pushbutton switch's "normal" status, which is to keep the motor in its present state. In some applications, this failure mode would not be a severe problem. In many applications, though, it is quite dangerous to have a running motor that cannot be stopped. For this reason, it is customary to design motor start/stop systems a bit differently from what has been shown here.

In order to build a "fail-stop" motor control system with our PLC, re-wire the the "Stop" pushbutton switch to use its normally-closed (NC) contact as shown in Figure 24.36.

Figure 24.36: The correct way to wire a failsafe motor control circuit with a PLC

This keeps discrete input channel 7 activated when the pushbutton is unpressed. When the operator presses the "Stop" pushbutton, the switch's contacts will be forced open, and input channel 7 will de-energize. If a wire happens to fall off a screw terminal in the "Stop" switch circuit, input channel 7 will de-energize just the same as if someone pressed the "Stop" pushbutton, which will automatically shut off the motor.

In order for the PLC program to work properly with this new switch wiring, the virtual contact for `IN_switch_Stop` must be changed from a normally-closed (NC) to a normally-open (NO) as in Figure 24.37.

Figure 24.37: The correct way to program a failsafe motor control circuit in a PLC

As before, the `IN_switch_Stop` virtual contact is in the "closed" state when no one presses the "Stop" switch, enabling the motor to start any time the "Start" switch is pressed. Similarly, the `IN_switch_Stop` virtual contact will open any time someone presses the "Stop" switch, thus stopping virtual "power" from flowing to the `OUT_contactor` coil.

Although this is a very common way to build PLC-controlled motor start/stop systems – with an NC pushbutton switch and an NO "Stop" virtual contact – students new to PLC programming often find this logical reversal confusing[19]. Perhaps the most common reason for this confusion is a misunderstanding of the "normal" concept for switch contacts, be they real or virtual. The `IN_switch_Stop` virtual contact is programmed to be normally-open (NO), but yet it is *typically* found in the closed state. Recall that the "normal" status of any switch is its status while in a resting condition of no stimulation, *not* necessarily its status while the process is in a "normal" operating mode. The "normally-open" virtual contact `IN_switch_Stop` is typically found in the closed state because its corresponding input channel is typically found energized, owing to the normally-closed pushbutton switch contacts, which pass voltage to the input channel while no one presses the switch. Just because a switch is configured as normally-open does not necessarily mean it will be *typically* found in the open state! The status of any switch contact, whether real or virtual, is a function of its configuration (NO versus NC) and the stimulus applied to it.

Another concern surrounding real-world wiring problems is what this system will do if the motor contactor coil circuit opens for any reason. An open circuit may develop as a result of a wire falling off a screw terminal, or it may occur because the thermal overload contact tripped open due to an over-temperature event. The problem with our motor start/stop system as designed is that it is not "aware" of the contactor's real status. In other words, the PLC "thinks" the contactor will be energized any time discrete output channel 2 is energized, but that may not actually be the case if

[19] In an effort to alleviate this confusion, the Allen-Bradley corporation (Rockwell Automation) uses the terms *examine if closed* (XIC) and *examine if open* (XIO) to describe "normally open" and "normally closed" virtual contacts, respectively, in their Ladder Diagram programming. The idea here is that a virtual contact drawn as a normally-open symbol will be "examined" (declared "true") by the PLC's processor if its corresponding input channel is energized (powered by a real-life contact in the *closed* state). Conversely, a virtual contact drawn as a normally-closed symbol (with a slash mark through the middle) will be "examined" by the PLC's processor if its corresponding input channel is de-energized (if the real-life contact sending power to that terminal is in the open state). This nomenclature is often even more confusing to students than simply calling these virtual contacts "normally open" and "normally closed" like other PLC manufacturers do. The foundational concept for students to grasp here is that *the virtual contact is not a direct representation of the real-life electrical switch contact – rather, it is a read instruction for the bit set by voltage coming from the real-life electrical switch contact.*

there is an open fault in the contactor's coil circuit.

This may lead to a dangerous condition if the open fault in the contactor's coil circuit is later cleared. Imagine an operator pressing the "Start" switch but noticing the motor does not actually start. Wondering why this may be, he or she goes to look at the overload relay to see if it is tripped. If it is tripped, and the operator presses the "Reset" button on the overload assembly, the motor will immediately start because the PLC's discrete output has remained energized all the time following the pressing of the "Start" switch. Having the motor start up as soon as the thermal overload is reset may come as a surprise to operations personnel, and this could be quite dangerous if anyone happens to be near the motor-powered machinery when it starts.

A motor control system that refuses to "latch" on unless the contactor actually energizes when the "Start" switch is pressed would be safer. For this to be possible, the PLC must have some way of sensing the contactor's status.

In order to make the PLC "aware" of the contactor's real status, connect the auxiliary switch contact to one of the unused discrete input channels on the PLC as shown in Figure 24.38.

Figure 24.38: The correct way to wire motor control circuit status feedback with a PLC

Now, the PLC has a way of sensing the contactor's actual status.

The PLC program may be modified to recognize this status by assigning a new tag name to this input (`IN_contactor_aux`) and using a normally-open virtual contact of this name as the seal-in contact instead of the `OUT_contactor` bit as shown in Figure 24.39.

Figure 24.39: The correct way to program motor control circuit status feedback in a PLC

Now, if the contactor fails to energize for any reason when the operator presses the "Start" switch, the PLC's output will fail to latch when the "Start" switch is released. When the open fault in the contactor's coil circuit is cleared, the motor will *not* immediately start up, but rather wait until the operator presses the "Start" switch again, which is a much safer operating characteristic than before. However, getting feedback from a motor contactor still does not indicate whether or not the motor itself is actually running and the motor load is turning. Often, a process variable is used for feedback instead of an auxiliary contact on a motor starter, such as motor current monitored with a current switch or differential pressure across a pump or fan monitored with a DP switch.

A special class of virtual "coil" used in PLC ladder programming that bears mentioning is the "latching" coil. These usually come in two forms: a *set* coil and a *reset* coil. Unlike a regular "output" coil that positively writes to a bit in the PLC's memory with every scan of the program, "set" and "reset" coils only write to a bit in memory when energized by virtual power. Otherwise, the bit is allowed to retain its last value.

A very simple motor start/stop program could be written with just two input contacts and two of these latching coils (both bearing the same tag name, writing to the same bit in memory) as seen in Figure 24.40.

Figure 24.40: An example using latching coils

Note the use of normally-open (NO) pushbutton switch contacts (again!), with no auxiliary contact providing status indication of the contactor to the PLC. This is a very minimal example program only, shown for the strict purpose of illustrating the use of "set" and "reset" latching coils in Ladder Diagram PLC programming, and should not be used as wired as it is not a failsafe design.

"Set" and "Reset" coils[20] are examples of what is known in the world of PLC programming as *retentive instructions*. A "retentive" instruction *retains* its value after being virtually "de-energized" in the Ladder Diagram "circuit." A standard output coil is *non-retentive*, which means it does not "latch" when de-energized. The concept of retentive and non-retentive instructions will appear again as we explore PLC programming, especially in the area of *timers*.

Ordinarily, multiple coils bearing the same label in a PLC Ladder Diagram program should never be used. With each coil representing a "write" instruction, multiple coils bearing the same name represents multiple "write" operations to the same bit in the PLC's memory. Here, with latching coils, there is no conflict because each of the coils only writes to the `OUT_contactor` bit when its respective contact is energized. So long as only one of the pushbutton switches is actuated at a time, there is no conflict between the identically-named coils.

This begs the question: what would happen if *both* pushbutton switches were simultaneously pressed? What would happen if *both* "Set" and "Reset" coils were "energized" at the same time? The result is that the `OUT_contactor` bit would first be "set" (written to a value of 1) then "reset" (written to a value of 0) in that order as the two rungs of the program were scanned from top to

[20] Referred to as "Latch" and "Unlatch" coils by Allen-Bradley.

bottom. As mentioned before, PLCs typically do not typically update their discrete I/O registers while executing the logic of the Ladder Diagram program (this operation takes place during the input or output data table updates of the scan as appropriate), so the real discrete output channel status will be whatever the *last* write operation told it to be, in this case "reset" (0, or off).

Multiple (non-retentive) output coils with the same memory address are almost always a programming *faux pax* for this reason. This is known as *multiple coil syndrome*, and some programming interfaces will not allow the program to compile if it occurs, but not all. It's best to program the output coil on a ladder rung first (and only use one for each output address), then add the input conditions that activate the output coil to the rung ahead of it. This violates the "left-to-right" workflow that we've been taught while learning how to write, but it prevents multiple coil syndrome. Multiple *contacts* with identical addresses are no problem whatsoever, because multiple "read" operations to the same bit in memory will never cause a conflict.

The IEC 61131-3 PLC programming standard specifies *transition-sensing* contacts as well as the more customary "static" contacts. A transition-sensing contact will "actuate" only for a duration of one program scan, even if its corresponding bit remains active. Two types of transition-sensing Ladder Diagram contacts are defined in the IEC standard: one for *positive* transitions and another for *negative* transitions. The example in Figure 24.41 shows a wiring diagram, Ladder Diagram program, and a timing diagram demonstrating how each type of transition-sensing contact functions when stimulated by a real (electrical) input signal to a discrete channel.

Figure 24.41: An example of the behavior of transition-sensing instructions

When the pushbutton switch is pressed and the discrete input energized, the first test lamp will blink "on" for exactly one scan of the PLC's program, then return to its off state. The positive-transition contact (with the letter "P" inside) activates the coil `OUT_test1` only during the scan it sees the status of `IN_test` transition from "false" to "true," even though the input remains energized for many scans after that transition. Conversely, when the pushbutton switch is released and the discrete input de-energizes, the second test lamp will blink "on" for exactly one scan of the PLC's program then return to its off state. The negative-transition contact (with the letter "N" inside) activates the coil `OUT_test2` only during the scan it sees the status of `IN_test` transition from "true"

to "false," even though the input remains de-energized for many scans after that transition:

It should be noted that the duration of a single PLC program scan is typically very short: measured in milliseconds. If this program were actually tested in a real PLC, you would probably not be able to see either test lamp light up, since each pulse is so short-lived. Transitional contacts are typically used any time it is desired to execute an instruction just one time following a "triggering" event, as opposed to executing that instruction over and over again so long as the event status is maintained "true."

Contacts and coils represent only the most basic of instructions in the Ladder Diagram PLC programming language. Many other instructions exist, which will be discussed in the following subsections.

### 24.4.2 Counters

A *counter* is a PLC instruction that either increments (counts up) or decrements (counts down) an integer number value when prompted by the transition of a bit from 0 to 1 ("false" to "true"). Counter instructions come in three basic types: *up* counters, *down* counters, and *up/down* counters. Both "up" and "down" counter instructions have single inputs for triggering counts, whereas "up/down" counters have two trigger inputs: one to make the counter increment and one to make the counter decrement.

To illustrate the use of a counter instruction, Figure 24.42 shows a PLC-based system designed to count objects as they pass down a conveyor belt.

Figure 24.42: An example wiring diagram for a system designed to count events

In this system, a continuous (unbroken) light beam causes the light sensor to close its output contact, energizing discrete channel IN4. When an object on the conveyor belt interrupts the light beam from source to sensor, the sensor's contact opens, interrupting power to input IN4. A pushbutton switch connected to activate discrete input IN5 when pressed will serve as a manual "reset" of the count value. An indicator lamp connected to one of the discrete output channels will serve as an indicator of when the object count value has exceeded some pre-set limit.

A simple Ladder Diagram program designed to increment a counter instruction each time the light beam breaks is shown in Figure 24.43.

Figure 24.43: The LD program for the system designed to count events

This particular counter instruction (CTU) is an incrementing counter, which means it counts "up" with each off-to-on transition input to its "CU" input. The normally-closed virtual contact (`IN_sensor_object`) is typically held in the "open" state when the light beam is continuous, by virtue of the fact the sensor holds that discrete input channel energized while the beam is continuous. When the beam is broken by a passing object on the conveyor belt, the input channel de-energizes, causing the virtual contact `IN_sensor_object` to "close" and send virtual power to the "CU" input of the counter instruction. This increments the counter just as the leading edge of the object breaks the beam. The second input of the counter instruction box ("R") is the *reset* input, receiving virtual power from the contact `IN_switch_reset` whenever the reset pushbutton is pressed. If this input is activated, the counter immediately resets its current value (CV) to zero.

Status indication is shown in this Ladder Diagram program, with the counter's preset value (PV) of 25 and the counter's current value (CV) of 0 shown highlighted in blue. The preset value is something programmed into the counter instruction before the system put into service, and it serves as a threshold for activating the counter's output (Q), which in this case turns on the count indicator lamp (the `OUT_counts_reached` coil). According to the IEC 61131-3 programming standard, this counter output should activate whenever the current value is equal to or greater than the preset value (Q is active if CV $\geq$ PV).

Figure 24.44 is the status of the same program after thirty objects have passed by the sensor on the conveyor belt. As can be seen, the current value of the counter has increased to 30, exceeding the preset value and activating the discrete output when the current value equalled 25.

Figure 24.44: The LD program for the system designed to count events after the output was activated

If maintaining an accurate total count of objects past 25 was not the goal – but rather the purpose of the program was to indicate when 25 or more objects had passed by – a *down* counter instruction preset to a value of 25 could also be used. This type of counter turns on an output coil when the count reaches zero as seen in Figure 24.45.

Figure 24.45: The LD program for the system designed to count events after the output was activated

Here, a "load" input causes the counter's current value to equal the preset value (25) when activated. With each sensor pulse received, the counter instruction decrements. When it reaches zero, the Q output activates.

A potential problem in either version of this object-counting system is that the PLC cannot discriminate between forward and reverse motion on the conveyor belt. If, for instance, the conveyor belt were ever reversed in direction, the sensor would continue to count objects that had already passed by before (in the forward direction) as those objects retreated on the belt. This would be a problem because the system would "think" more objects had passed along the belt (indicating greater production) than actually did.

One solution to this problem is to use an up/down counter, capable of both incrementing (counting up) and decrementing (counting down), and equip this counter with two light-beam sensors capable of determining direction of travel. If two light beams are oriented parallel to each other,

closer than the width of the narrowest object passing along the conveyor belt (half of the width is often used), there will be enough information to determine direction of object travel. This type of system is shown in Figure 24.46.

Figure 24.46: Diagram of a system with directional counting ability

This is called *quadrature* signal timing[21], because the two pulse waveforms are shifted apart in phase. These two phase-shifted signals can be used to increment or decrement an up/down counter instruction, depending on which pulse leads and which pulse lags.

A Ladder Diagram PLC program designed to interpret the quadrature pulse signals is shown in Figure 24.47. It makes use of negative-transition contacts as well as standard contacts.

[21] The two pulse waveforms are approximately $90^o$ (one-*quarter* of a period) apart in phase (especially if the sensor's are placed a half-width of the object apart) leading to the term *quadrature*.

Figure 24.47: The LD program for the system designed to count events after the output was activated

The counter will increment (count up) when sensor B de-energizes only if sensor A is already in the de-energized state (i.e., light beam A breaks before B). The counter will decrement (count down) when sensor A de-energizes only if sensor B is already in the de-energized state (i.e., light beam B breaks before A).

Note that the up/down counter has both a "reset" (R) input and a "load" input ("LD") to force the current value. Activating the reset input forces the counter's current value (CV) to zero, just as we saw with the "up" counter instruction. Activating the load input forces the counter's current value to the preset value (PV), just as we saw with the "down" counter instruction. In the case of an up/down counter, there are two Q outputs: a QU (output up) to indicate when the current value is equal to or greater than the preset value, and a QD (output down) to indicate when the current value is equal to or less than zero.

Note how the current value (CV) of each counter shown is associated with a tag name of its own, in this case `parts_counted`. The integer number of a counter's current value (CV) is a variable in the PLC's memory just like boolean values such as `IN_sensor_A` and `IN_switch_reset`, and may be associated with a tag name or symbolic address just the same[22]. This allows other instructions in a PLC program to read (and sometimes write!) values from and to that memory location.

[22]This represents the IEC 61131-3 standard, where each variable within an instruction may be "connected" to its own arbitrary tag name. Other programming conventions may differ somewhat. The Allen-Bradley Logix series of controllers (i.e., Controllogix, CompactLogix, and GuardLogix) is one of those that differs, following a convention reminiscent of structure element addressing in the C programming language: each counter is given a tag name, and variables in each counter are addressed as elements within that structure. For example, a Logix counter instruction might be named `parts_count`, with the accumulated count value (equivalent to the IEC's "current value") addressed as `parts_count.ACC` (each element within the counter specified as a suffix to the counter's tag name).

### 24.4.3 Timers

A *timer* is a PLC instruction measuring the amount of time elapsed following an event. Timer instructions come in two basic types: *on-delay* timers and *off-delay* timers. Both "on-delay" and "off-delay" timer instructions have single inputs triggering the timed function.

An "on-delay" timer activates an output only when the input has been active for a minimum amount of time. The PLC program in Figure 24.48 is designed to sound an audio alarm siren prior to starting a conveyor belt. To start the conveyor belt motor, the operator must press and hold the "Start" pushbutton for 10 seconds, during which time the siren sounds, warning people to clear away from the conveyor belt that is about to start. Only after this 10-second start delay does the motor actually start (and latch "on").

Figure 24.48: An example diagram of a system with a timer function

Similar to an "up" counter, the on-delay timer's elapsed time (ET) value increments once per second until the preset time (PT) is reached, at which time its output (Q) activates. In this program, the preset time value is 10 seconds, which means the Q output will not activate until the "Start" switch has been depressed for 10 seconds. The alarm siren output, which is not activated by the timer, energizes immediately when the "Start" pushbutton is pressed.

An important detail regarding this particular timer's operation is that it be *non-retentive*. This means the timer instruction should *not* retain its elapsed time value when the input is de-activated. Instead, the elapsed time value should reset back to zero every time the input de-activates. This ensures the timer resets itself when the operator releases the "Start" pushbutton. A *retentive* on-

delay timer, by contrast, maintains its elapsed time value even when the input is de-activated. This makes it useful for keeping "running total" times for some event.

Most PLCs provide retentive and non-retentive versions of on-delay timer instructions, such that the programmer may choose the proper form of on-delay timer for any particular application. The IEC 61131-3 programming standard, however, addresses the issue of retentive versus non-retentive timers a bit differently. According to the IEC 61131-3 standard, a timer instruction may be specified with an additional *enable* input (EN) that causes the timer instruction to behave non-retentively when activated, and retentively when de-activated. The general concept of the enable (EN) input is that the instruction behaves "normally" so long as the enable input is active (in this case, non-retentive timing action is considered "normal" according to the IEC 61131-3 standard), but the instruction "freezes" all execution whenever the enable input de-activates. This "freezing" of operation has the effect of retaining the current time (CT) value even if the input signal de-activates.

For example, an "enabled" IEC 61131-3 timer instruction like the one seen in Figure 24.49 could be used to add a retentive timer to the conveyor control system to record total run time for the conveyor motor.

Figure 24.49: An example diagram of a system with an IEC 61131-3 retentive timer function

When the motor's contactor bit (`OUT_contactor`) is active, the timer is enabled and allowed to time. However, when that bit de-activates (becomes "false"), the timer instruction as a whole is disabled, causing it to "freeze" and retain its current time (CT) value[23]. This allows the motor to be started and stopped, with the timer maintaining a tally of total motor run time.

To give the operator the ability to manually reset the total run time value to zero, an additional switch could be hard-wired to the PLC's discrete input card and "reset" contacts added to the program as seen in Figure 24.50.

[23]The "enable out" (ENO) signal on the timer instruction serves to indicate the instruction's status: it activates when the enable input (EN) activates and de-activates when either the enable input de-activates or the instruction generates an error condition (as determined by the PLC manufacturer's internal programming). The ENO output signal serves no useful purpose in this particular program, but it is available if there were any need for other rungs of the program to be "aware" of the run-time timer's status.

Figure 24.50: An example diagram of a system with a resettable timer function

Whenever the "Reset" switch is pressed, the timer is enabled (EN) but the timing input (IN) is disabled, forcing the timer to (non-retentively) reset its current time (CT) value to zero.

The other major type of PLC timer instruction is the *off-delay* timer. This timer instruction differs from the on-delay type in that the timing function begins as soon as the instruction is de-activated, not when it is activated. An application for an off-delay timer is a cooling fan motor control for a large industrial engine as shown in Figure 24.51. In this system, the PLC starts an electric cooling fan as soon as the engine is detected as rotating, and keeps that fan running for two minutes following the engine's shut-down to dissipate residual heat.

Figure 24.51: An example diagram of a system with an off-delay timer function

When the input (IN) to this timer instruction is activated, the output (Q) immediately activates (with no time delay at all) to turn on the cooling fan motor contactor. This provides the engine with cooling as soon as it begins to rotate (as detected by the speed switch connected to the PLC's discrete input). When the engine stops rotating, the speed switch returns to its normally-open position, de-activating the timer's input signal which starts the timing sequence. The Q output remains active while the timer counts from 0 seconds to 120 seconds. As soon as it reaches 120 seconds, the output de-activates (shutting off the cooling fan motor) and the elapsed time value remains at 120 seconds until the input re-activates, at which time it resets back to zero.

The timing diagrams in Figure 24.52 compare and contrast on-delay with off-delay timers.

Figure 24.52: A timing diagram comparing on-delay and off-delay timer functions

While it is common to find on-delay PLC instructions offered in both retentive and non-retentive forms within the instruction sets of nearly every PLC manufacturer and model, it is almost unheard of to find retentive off-delay timer instructions. Typically, off-delay timers are non-retentive only[24].

### 24.4.4 Data comparison instructions

As we have seen with counter and timers, some PLC instructions generate digital values other than simple Boolean (on/off) signals. Counters have current value (CV) registers and timers have elapsed time (ET) registers, both of which are typically integer number values. Many other PLC instructions are designed to receive and manipulate non-Boolean values such as these to perform useful control functions.

The IEC 61131-3 standard specifies a variety of *data comparison* instructions for comparing two non-Boolean values, and generating Boolean outputs. The basic comparative operations of "less than" ($<$), "greater than" ($>$), "less than or equal to" ($\leq$), "greater than or equal to" ($\geq$), "equal to" ($=$), and "not equal to" ($\neq$) may be found as a series of "box" instructions in the IEC standard as shown in Figure 24.53.

[24]The enable (EN) input signals specified in the IEC 61131-3 programming standard make retentive off-delay timers possible (by de-activating the enable input while maintaining the "IN" input in an inactive state), but bear in mind that most PLC implementations of timers do not have separate EN and IN inputs. This means (for most PLC timer instructions) the only input available to activate the timer is the "IN" input, in which case it is *impossible* to create a retentive off-delay timer (since such a timer's elapsed time value would be immediately re-set to zero each time the input re-activates).

Figure 24.53: Comparison functions

The Q output for each instruction "box" activates whenever the evaluated comparison function is "true" and the enable input (EN) is active. If the enable input remains active but the comparison function is false, the Q output de-activates. If the enable input de-de-activates, the Q output retains its last state.

A practical application for a comparative function is something called *alternating motor control*, where the run-times of two redundant electric motors[25] are monitored, with the PLC determining which motor to turn on next based on which motor has run the least. This method of control is shown in Figure 24.54.

[25]Perhaps two pumps performing the same pumping function, one serving as a backup to the other. Alternating motor control ensures the two motors' run times are matched as closely as possible. This method is controversial among maintenance engineers, with some liking it, and others worrying that if run time is equalized that the backup might fail nearly simultaneously. Some will set up an 80/20 or 75/25 split instead of a 50/50 split.

Figure 24.54: Examples LD program used to equalize run-time between motors

In this program, two retentive on-delay timers keep track of each electric motor's total run time, storing the run time values in two registers in the PLC's memory: `Motor_A_runtime` and `Motor_B_runtime`. These two integer values are input to the "greater than" instruction box for comparison. If motor A has run longer than motor B, motor B will be the one enabled to start up next time the "start" switch is pressed. If motor A has run less time or the same amount of time as motor B (the scenario shown by the blue-highlighted status indications), motor A will be

the one enabled to start. The two series-connected virtual contacts `OUT_motor_A` and `OUT_motor_B` ensure the comparison between motor run times is not made until both motors are stopped. If the comparison were continually made, a situation might arise where *both* motors would start if someone happened to press the Start pushbutton with one motor is already running.

### 24.4.5 Math instructions

The IEC 61131-3 standard specifies several dedicated ladder instructions for performing arithmetic calculations. Some of them are shown in Figure 24.55.

Figure 24.55: Examples of IEC 61131-3 math functions

As with the data comparison instructions, each of these math instructions must be enabled by an "energized" signal to the enable (EN) input. Input and output values are linked to each math instruction by tag name.

An example showing the use of such instructions is shown in Figure 24.56. This LD program converts a temperature measurement in units of degrees Fahrenheit to units of degrees Celsius. In this particular case, the program inputs a temperature measurement of 138 $^{o}$F and calculates the equivalent temperature of 58.89 $^{o}$C.

Figure 24.56: Examples of math functions used for temperature conversion

Note how two separate math instructions were required to perform this simple calculation, as well as a dedicated variable (`X`) used to store the intermediate calculation between the subtraction and the division "boxes." Ladder Diagram programming for mathematics requires a significant amount of effort on the part of the programmer (which is why some prefer Instruction List for complicated math calculations).

Although not specified in the IEC 61131-3 standard, many programmable logic controllers support Ladder Diagram math instructions allowing the direct entry of arbitrary equations. Rockwell (Allen-Bradley) Logix5000 programming, for example, has the "Compute" (`CPT`) function, which allows any typed expression to be computed in a single instruction as opposed to using several

dedicated math instructions such as "Add," "Subtract," etc. General-purpose math instructions dramatically shorten the length of a ladder program compared to the use of dedicated math instructions for any applications requiring non-trivial calculations.

For example, the same Fahrenheit-to-Celsius temperature conversion program implemented in Logix5000 programming for a Logix-series processor only requires a single math instruction and no declarations of intermediate variables as seen in Figure 24.57.

Figure 24.57: Example of "Compute" function used for temperature conversion

### 24.4.6 Sequencers

Many industrial processes require control actions to take place in certain, predefined sequences. Batch processes are perhaps the most striking example of this, where materials for making a batch must be loaded into the process vessels, parameters such as temperature and pressure controlled during the batch processing, and then discharge of the product monitored and controlled. Before the advent of reliable programmable logic devices, this form of sequenced control was usually managed by an electromechanical device known as a *drum sequencer*. This device worked on the principle of a rotating cylinder (drum) equipped with tabs to actuate switches as the drum rotated into certain positions. If the drum rotated at a constant speed (turned by a clock motor), those switches would actuate according to a timed schedule[26].

A typical application for a sequencer is to control a *Clean In Place* (*CIP*) system for a food processing vessel, where a process vessel must undergo a cleaning cycle to purge it of any biological matter between food processing cycles. The steps required to clean the vessel are well-defined and

[26]The operation of the drum is not unlike that of an old *player piano*, where a strip of paper punched with holes caused hammers in the piano to automatically strike their respective strings as the strip was moved along at a set speed, thus playing a pre-programmed song.

must always occur in the same sequence in order to ensure hygienic conditions. An example timing chart is shown in Figure 24.58.

Figure 24.58: Example timing chart for a sequenced operation

In this example, there are nine discrete outputs – one for each of the nine final control elements (pumps and valves) – and seventeen steps to the sequence, each one of them timed. In this particular sequence, the only input is the discrete signal to commence the CIP cycle. From the initiation of the CIP to its conclusion two and a half hours (150 minutes) later, the sequencer simply steps through the programmed routine.

Another practical application for a sequencer is to implement a *Burner Management System* (BMS), also called a *flame safety system*. Here, the sequencer manages the safe start-up of a

combustion burner: beginning by "purging" the combustion chamber with fresh air to sweep out any residual fuel vapors, waiting for the command to light the fire, energizing a spark ignition system on command, and then continuously monitoring for presence of good flame and proper fuel supply pressure once the burner is lit.

In a general sense, the operation of a drum sequencer is that of a *state machine*: the output of the system depends on the condition of the machine's internal state (the drum position), not just the conditions of the input signals. Digital computers are very adept at implementing state functions, and so the general function of a drum sequencer should be (and is) easy to implement in a PLC. Other PLC functions previously discussed ("latches" and timers in particular) are similar, in that the PLC's output at any given time is a function of both its present input condition(s) and its past input condition(s). Sequencing functions expand upon this concept to define a much larger number of possible states ("positions" of a "drum"), some of which may even be timed.

Unfortunately, despite the utility of drum sequence functions and their ease of implementation in digital form, there seems to be very little standardization between PLC manufacturers regarding sequencing functions. Sadly, the IEC 61131-3 standard (at least at the time of this writing, in 2009) does not specifically define a sequencing function suitable for Ladder Diagram programming. PLC manufacturers are left to invent sequencing instructions of their own design.

### Koyo "drum" instructions

The *drum* instruction offered in Koyo PLCs is a model of simplicity itself. This instruction is practically self-explanatory, as shown in the following example:

Figure 24.59: Example of a time-based drum sequencer instruction in a Koyo CLICK PLC

The three-by-three grid of squares represent steps in the sequence and bit states for each step. Rows represent steps, while columns represent output bits written by the drum instruction. In this particular example, a three-step sequence proceeds at the command of a single input (`X001`), and the drum instruction's advance from one step to the next proceeds strictly on the basis of elapsed time (a *time-based* orientation). When the input is active, the drum proceeds through its timed sequence. When the input is inactive, the drum halts wherever it left off, and resumes timing as soon as the input becomes active again.

Being based on time, each step in the drum instruction has a set time duration for completion. The first step in this particular example has a duration of 10 seconds, the second step 15 seconds, and the third step 18 seconds. At the first step, only output bit `Y001` is set. In the second step, only output bit `Y002` is set. In the third step, output bits `Y002` and `Y003` are set (1), while bit `Y001` is reset (0). The colored versus uncolored boxes reveal which output bits are set and reset with each step. The current step number is held in memory register `DS1`, while the elapsed time (in seconds) is stored in timer register `TD1`. A "complete" bit is set at the conclusion of the three-step sequence.

Koyo drum instructions may be expanded to include more than three steps and more than three output bits, with each of those step times independently adjustable and each of the output bits arbitrarily assigned to any writable bit addresses in the PLC's memory.

The example of a Koyo drum instruction in Figure 24.60 shows how it may be set up to trigger on *events* rather than on elapsed times. This setup is called an *event-based* configuration.

Figure 24.60: Example of an event-based drum sequencer instruction in a Koyo CLICK PLC

Here, a three-step sequence proceeds when enabled by a single input (`X001`), with the drum instruction's advance from one step to the next proceeding only as the different event condition bits become set. When the input is active, the drum proceeds through its sequence when each event condition is met. When the input is inactive, the drum halts wherever it left off regardless of the event bit states.

For example, during the first step (when only output bit `Y001` is set), the drum instruction waits for the first condition input bit `X002` to become set (1) before proceeding to step 2, with time being irrelevant. When this happens, the drum immediately advances to step 2 and waits for input bit `X003` to be set, and so forth. If all three event conditions were met simultaneously (`X002`, `X003`, and `X004` all set to 1), the drum would skip through all steps as fast as it could (one step per PLC program scan) with no appreciable time elapsed for each step. Conversely, the drum instruction will wait as long as it must for the right condition to be met before advancing, whether that event takes place in milliseconds or in days.

### Allen-Bradley sequencer instructions

Rockwell (Allen-Bradley) PLCs use a more sophisticated set of instructions to implement sequences. The closest equivalent to Koyo's *drum* instruction is the Allen-Bradley *SQO* (Sequencer Output) instruction, shown here:

Figure 24.61: Example of a sequencer output (*SQO*) instruction in a SLC 500 PLC

Notice there are no colored squares inside the SQO instruction box to specify when certain bits are set or reset throughout the sequence, in contrast to the simplicity of the Koyo PLC's drum instruction. Instead, the Allen-Bradley SQO instruction is told to read a set of 16-bit words beginning at a location in the PLC's memory arbitrarily specified by the programmer, one word at a time. It steps to the next word in that set of words with each new position (step) value. This means Allen-Bradley sequencer instructions rely on the programmer already having pre-loaded an area of the PLC's memory with the necessary 1's and 0's defining the sequence. This makes the Allen-Bradley sequencer instruction more challenging for a human programmer to interpret because the bit states are not explicitly shown inside the SQO instruction box, but it also makes the sequencer far more flexible in that these bits are not fixed parameters of the SQO instruction and therefore may be dynamically altered as the PLC runs. With the Koyo drum instruction, the assigned output states are part of the instruction itself, and are therefore fixed once the program is downloaded to the PLC (i.e., they cannot be altered without editing and re-loading the PLC's program). With the Allen-Bradley, the on-or-off bit states for the sequence may be freely altered[27] during run-time. This is a very useful feature in recipe-control applications, where the recipe is subject to change at the whim of production personnel, and they would rather not have to rely on a technician or an engineer to re-program the PLC for each new recipe.

[27] Perhaps the most practical way to give production personnel access to these bits without having them learn and use PLC programming software is to program an HMI panel to write to those memory areas of the PLC. This way, the operators may edit the sequence at any time simply by pressing "buttons" on the screen of the HMI panel, and the PLC need not have its program altered in any "hard" way by a technician or engineer.

The "Length" parameter tells the SQO instruction how many words will be read (i.e., how many steps are in the entire sequence). The sequencer advances to each new position when its enabling input transitions from inactive to active (from "false" to "true"), just like a count-up (CTU) instruction increments its accumulator value with each new false-to-true transition of the input. Here we see another important difference between the Allen-Bradley SQO instruction and the Koyo drum instruction: the Allen-Bradley instruction is fundamentally *event-driven*, and does not proceed on its own like the Koyo drum instruction is able to when configured for a *time* base.

Sequencer instructions in Allen-Bradley PLCs use a notation called *indexed addressing* to specify the locations in memory for the set of 16-bit words it will read. In the example shown above, we see the "File" parameter specified as `#B3:0`. The "#" symbol tells the instruction that this is a *starting* location in memory for the first 16-bit word, when the instruction's position value is zero. As the position value increments, the SQO instruction reads 16-bit words from successive addresses in the PLC's memory. If `B3:0` is the word referenced at position 0, then `B3:1` will be the memory address read at position 1, `B3:2` will be the memory address read at position 2, etc. Thus, the "position" value causes the SQO instruction to "point" or "index" to successive memory locations.

The bits read from each indexed word in the sequence are compared against a static mask[28] specifying which bits in the indexed word are relevant. At each position, only these bits are written to the destination address.

As with most other Allen-Bradley instructions, the sequencer requires the human programmer to declare a special area in memory reserved for the instruction's internal use. The "`R6`" file exists just for this purpose, each element in that file holding bit and integer values associated with a sequencer instruction (e.g., the "enable" and "done" bits, the array length, the current position, etc.).

Figure 24.62 shows how each row (element) of a set of bits held in the `B3` file of an Allen-Bradley SLC 500 PLC would be read by an SQO instruction as it stepped through its positions.

Figure 24.62: Indirect addressing for a sequencer output (*SQO*) instruction reading bits in a SLC 500 PLC

The sequencer's position number is added to the file reference address as an *offset*. Thus, if the data file is specified in the SQO instruction box as `#B3:0`, then `B3:1` will be the row of bits read

[28] In this particular example, the mask value is FFFF hexadecimal, which means all 1's in a 16-bit field. This mask value tells the sequencer instruction to regard *all* bits of each `B3` word that is read. To contrast, if the mask were set to a value of 000F hexadecimal instead, the sequencer would only pay attention to the four least-significant bits of each `B3` word that is read, while ignoring the 12 more-significant bits of each 16-bit word. The mask allows the SQO instruction to only write to selected bits of the destination word, rather than always writing all 16 bits of the indexed word to the destination word.

when the sequencer's position value is 1, `B3:2` will be the row of bits read when the position value is 2, and so on.

The *mask* value specified in the SQO instruction tells the instruction which bits out of each row will be copied to the destination address. A mask value of FFFFh (FFFF in *hexadecimal* format) means all 16 bits of each `B3` word will be read and written to the destination. A mask value of 0001h means only the first (least-significant) bit will be read and written, with the rest being ignored.

Let's see what would happen with an SQO instruction having a mask value of 000Fh, starting from file index `#B3:0`, and writing to a destination that is output register `O:0.0`, given the bit array values in file `B3` shown in Figure 24.63.

Figure 24.63: Example of a sequencer output (*SQO*) instruction reading bits in a SLC 500 PLC

When this SQO instruction is at position 2, it reads the bit values `0010` from `B3:2` and writes only those four bits to `O:0.0`. The "X" symbols shown in the illustration mean that all the other bits in that output register are untouched – the SQO instruction does not write to those bits because they are "masked off" from being written. You may think of the mask's zero bits inhibiting source bits from being written to the destination word in the same sense that *masking tape* prevents paint from being applied to a surface.

The following Allen-Bradley SLC 500 PLC program in Figure 24.64 shows how a pair of SQO instructions plus an on-delay timer instruction may be used to duplicate the exact same functionality as the "time base" Koyo drum instruction presented earlier.

Figure 24.64: Example of (*SQO*) instructions duplicating a time-based drum sequencer in a SLC 500 PLC

The first SQO instruction reads bits in the `B3` file array, sending only the three least-significant of them to the output register `O:0.0` (as specified by the 0007h mask value). The second SQO instruction reads integer number values from elements of the `N7` integer file and places them into the "preset" register of timer `T4:0`, so as to dynamically update the timer's preset value with each step of the sequence. The timer, in turn, counts off each of the time delays and then enables both sequencers to advance to the next position when the specified time has elapsed. This is the tremendous benefit of the SQO instruction's indexed memory addressing: the fact that the SQO instruction reads its bits from arbitrarily-specified memory addresses means SQO instructions may be used to sequence *any type of data existing in the PLC's memory!* Thus, the instruction is not limited to turning on and off individual bits as with the Koyo drum instruction, but rather it is free to index whole integer numbers, ASCII characters, or any other forms of binary data resident in the PLC's memory.

Data file windows appear on the computer screen showing the bit array held in the `B3` file as well as the timer values held in the `N7` file. In the live screenshot in Figure 24.64, both sequencer instructions are shown at position 2, with the second SQO instruction having loaded a value of 15 seconds from register `N7:2` to the timer's preset register `T4:0.PRE`.

Note how the enabling contact address for the second SQO instruction is the "enable" bit of the first instruction, ensuring both instructions are enabled simultaneously. This keeps the two separate sequencers synchronized (on the same step).

Event-based transitions may be implemented in Allen-Bradley PLCs using a complementary sequencing instruction called SQC (Sequencer Compare). The SQC instruction is set up very similar to the SQO instruction, with an indexed file reference address to read from, a reserved memory structure for internal use, a set length, and a position value. The purpose of the SQC instruction is to read a data register and compare it against another data register, setting a "found" (`FD`) bit

if the two match. Thus, the SQC instruction is ideally suited for detecting when certain conditions have been met, and thus may be used to enable an SQO instruction to proceed to the next step in its sequence.

The program example in Figure 24.65 shows an Allen-Bradley MicroLogix 1100 PLC programmed with both an SQO and an SQC instruction.

Figure 24.65: Example of (*SQO*) and (*SQC*) instructions duplicating an event-based drum sequencer in a SLC 500 PLC

The three-position SQO (Sequencer Output) instruction reads data from `B3:1`, `B3:2`, and `B3:3`, writing the four least-significant of those bits to output register `O:0.0`. The three-position SQC (Sequencer Compare) instruction reads data from `B3:6`, `B3:7`, and `B3:8`, comparing the four least-significant of those bits against input bits in register `I:0.0`. When the four input bit conditions match the selected bits in the `B3` file, the SQC instruction's `FD` bit is set, causing both the SQO instruction and the SQC instruction to advance to the next step.

Lastly, Allen-Bradley PLCs offer a third sequencing instruction called *Sequencer Load* (SQL), which performs the opposite function as the Sequencer Output (SQO). An SQL instruction takes data from a designated source and writes it into an indexed register according to a position count value, rather than reading data from an indexed register and sending it to a designated destination as does the SQO instruction. SQL instructions are useful for reading data from a live process and storing it in different registers within the PLC's memory at different times, such as when a PLC is used for *datalogging* (recording process data).

## 24.5 Function Block Diagram (FBD) programming

Function block programming is a programming environment where function blocks are dropped on a "page" and wires are used to connect graphical functions together.

### 24.5.1 A FBD programming example

Figure 24.66 is an example of function block programming in the RSLogix5000 environment for a Logix5000 series controller. The example shows a complete PID control loop, from temperature transmitter to the PID algorithm block to the I/P converter for the temperature valve.

Figure 24.66: A function block program in a PAC

### 24.5.2 A second FBD programming example

Figure 24.67 shows a motor control function programmed in ladder diagram for a generic tag-based controller.

Figure 24.67: A ladder diagram motor control circuit program in a PLC

The Boolean logic expression for this is:

```
(IN_switch_Start OR IN_contactor_aux) AND IN_switch_Stop = OUT_contactor
```

Figure 24.68 shows the equivalent FBD program for this functionality.

Figure 24.68: A function block motor control circuit program in a PAC

## 24.6 Structured Text (ST) programming

Structured Text programming is a text-based programming language that is similar to C or other high-level programming languages. Since most U.S. (current) maintenance workers (e.g., electricians) are not familiar with any programmming languages, engineers do not often use it for programming in the U.S. However, structured text is the easier of the two text-based programming languages to read and understand for someone who has programmed using a high-level language (most people programming PLCs have never seen assemby language). Structured text can contain *if-then-else*, *for-next*, *while*, and case *structures*, as well as assignment and standard mathematical operators (including an order of precedence).

### 24.6.1 A ST programming example

Temperature transmitters are reading the inlet and outlet water temperatures of a heat exchanger and a mass flow transmitter measures the water flow. The analog input values are stored in their proper engineering units in memory tags `Tin`, `Tout`, and `Flow`, respectively. The rate of heat transferred into the water is to be calculated according to $\dot{Q} = \dot{m}c_p(T_out - T_in)$ and stored in tag `Qdot` assuming a constant value of specific heat capacity stored in tag `c_p`[29]. If the outlet temperature exceeds a value hard-coded by high and low limits, the process downstream of the heat exchanger won't function properly and a high- or low-level alarm is triggered. The following structured text program solves this problem on an Allen-Bradley Logix series PAC:

```
c_p := 1; /*Assign specific heat capacity for water*/
Qdot := Flow*c_p*(Tout-Tin); /*Calculate rate of heat transfer into water*/
IF Tout > 100 THEN
Tout_AH = 1; /*Set high alarm if Tout exceeds 100 deg F*/
ELSEIF Tout < 80 THEN
Tout_AL = 1; /*Set low alarm if Tout below 80 deg F*/
END_IF
```

Note that the assignment operator is given by `:=`, a semicolon follows each assignment expression, and /*...*/ contains a comment. The order of mathematical precedence in the calculation of $\dot{Q}$ is maintained by the use of parentheses.

If the ST code will contain a function block like a timer or a counter (instead of a simple function like a math or assignment operator), the function block will have to be instantiated by creating an associated tag just like it does in ladder diagram programming.

## 24.7 Instruction List (IL) programming

Instruction List programming is a text-based programming that is similar to assembly language. It uses low-level mnemonics that stand for instructions. Again, since most U.S. maintenance workers (and these days, engineers) are not familiar with assembly programmming for microprocessors, it's not often used for programming in the U.S.

[29] Tag creation in Rockwell Automation's RSLogix5000 ST programs can occur before writing the code, or tags may be created on the fly as the code is written by pointing at the text variable and using the right-click menu selection of "New tag." For the purposes of this example, assume the tags have been previously created.

### 24.7.1 An IL programming example

Figure 24.69 shows a motor control function programmed in ladder diagram for a generic tag-based controller.

Figure 24.69: A ladder diagram motor control circuit program in a PLC

The Boolean logic expression for this is:

```
(IN_switch_Start OR IN_contactor_aux) AND IN_switch_Stop = OUT_contactor
```

The Siemens S7 IL program that is equivalent to this follows:

```
A(
O "IN_switch_Start" //Start button
O "IN_contactor_aux" //Aux contact seal
)
A "IN_switch_Stop" //Stop button
= "OUT_contactor" //Motor starter
```

where `A` is the AND operator, `O` is the OR operator, `=` is the assignment operator, and `//` indicates a comment. Each instruction gets its own line in the program. Note that the Start switch and Aux contact bits that are in the logic OR function both have the `O` command in front of them indicating that they belong to the same OR command. They are grouped in the parentheses indicating they are to be evaluated together first. The result of the OR logic and the Stop switch bit belong to the same AND command as indicated by the fact that they both have are associated with the `A` command. Interestingly enough, the IL is very similar to the Excel expression that would be created to solve the same logic: `AND(OR(IN_switch_Start,IN_contactor_aux),IN_switch_Stop)`.

## 24.8 Sequential Function Charts (SFC)

Sequential function charts are not really a programming language, but rather, a program organizational tool a lot like a flowchart. SFCs work especially well for any automation system that require sequencers to repeat events over and over again, with each event having a well-defined

transition into it. A *step* is a piece of code that executes as long as the transition below it is false. The code may be written in any of the languages described earlier that are supported by the SFC structure. For example, in an Allen-Bradley Logix controller the action in a step is either written directly as ST in the SFC environment, or the action may be located in a separate LD or FB routine that is executed when directed by a jump-to-subroutine instruction in the step. A *transition* is a piece of code that evaluates to a Boolean value of 1 (TRUE) or 0 (FALSE) that either allows or prevents the next step from executing. It may be as simple as a logic 1 assigned to the transition which means that it is always true, or it may be conditional logic or the result of a time delay function.

### 24.8.1 Sequencing

Figure 24.70 is an example of portion of a sequential function chart organizational structure. After the logic condition associated with Transition 1 becomes true, Step 2 will execute. Step 2 will continue to run until the logic condition associated with Transition 2 becomes true, then Step 3 will execute. This is a *sequencing* structure written in SFC.

Figure 24.70: A portion of a sequence organized in a sequential function chart

### 24.8.2 Looping

Figure 24.71 shows a SFC structure that repeats Steps 1, 2, and 3 indefinitely. This is a *loop* structure written in SFC.

Figure 24.71: A SFC sequence that repeats due to looping

### 24.8.3 Selection

Figure 24.72 shows a SFC structure that selects either Step 2 or 3 depending on the truth of the transition condition above each step. This is a *selection* structure written in SFC. By default, the SFC checks the transitions from left to right, meaning if both transitions are true simultaneously, Step 2 will be executed, not Step 3. However, many programming environments allow priorities to be assigned to the transitions to determine the order of execution.

Figure 24.72: A SFC sequence that selects a step to execute based on the transition conditions.

### Bypassing

Figure 24.73 shows a SFC structure that selects either Step 2 or skips it depending on the transition conditions. This is a *selection* structure written in SFC that shows a step being bypassed.

Figure 24.73: A SFC sequence that can bypass a step

### 24.8.4 Simultaneous execution

Figure 24.74 shows a SFC structure that executes both Steps 2 and 3. This is a *simultaneous sequence* structure written in SFC.

Figure 24.74: A SFC sequence that executes steps in parallel with each other

SFCs have definite advantages when used to control a sequencing system like a batch process, but may not be the best choice for a continuous process that doesn't sequence.

## 24.9 Human-Machine Interfaces

Many modern PLC systems are equipped with computer-based interface panels which human operators use to observe data gathered by the PLC and also enter data to the PLC. These panels are generally referred to as *Human*[30]*-Machine Interfaces*, or *HMI* panels.

HMIs may take the form of general-purpose ("personal") computers running special graphic software to interface with a PLC, or as special-purpose computers designed to be mounted in sheet metal panel fronts to perform no task but the operator-PLC interface. Figure 24.75 shows an example of the former, and Figure 24.76 an example of the latter:

[30] An older term for an operator interface panel was the "Man-Machine Interface" or "MMI." However, this fell out of favor due to its sexist tone.

Figure 24.75: An HMI running on a general-purpose desktop computer

Figure 24.76: An HMI running on an industrial operator interface computer

Modern HMI panels and software are almost exclusively tag-based, with each graphic object on the screen associated with at least one data tag name, which in turn is associated to data points (bits, or words) in the PLC by way of a tag name database file resident in the HMI. Graphic objects on the HMI screen either accept (read) data from the PLC to present useful information to the operator, send (write) data to the PLC from operator input, or both. The task of programming an HMI unit consists of building a tag name database and then drawing screens to illustrate the process to as good a level of detail as operators will need to run it.

An example screenshot of a tag name database table for a modern HMI is shown in Figure 24.77.

| No. | Tag Name | Data Type | PLC Address | Device Name | Attribute |
|---|---|---|---|---|---|
| 1 | START_PUSHBUTTON | Discrete | 10024 | PLC_01 | R |
| 2 | STOP_PUSHBUTTON | Discrete | 10031 | PLC_01 | R |
| 3 | MOTOR_RUN | Discrete | 00005 | PLC_01 | R/W |
| 4 | MOTOR_RUN_TIMER | Unsigned int 32 | 40010 | PLC_01 | R/W |
| 5 | START_COUNTER | Unsigned int 16 | 40242 | PLC_01 | R/W |
| 6 | MOTOR_TEMPERATURE | Floating PT 32 | 30008 | PLC_01 | R |
| 7 | ERROR_MESSAGE | Ascii String | 40560 | PLC_01 | R/W |
| 8 | MOTOR_SPEED | Floating PT 32 | 30017 | PLC_01 | R |

Figure 24.77: An HMI running on an industrial operator interface computer

The tag name database is accessed and edited using the same software to create graphic images in the HMI. In this particular example you can see several tag names (e.g., `START_PUSHBUTTON`, `MOTOR_RUN_TIMER`, `ERROR_MESSAGE`, `MOTOR_SPEED`) associated with data points within the PLC's memory (in this example, the PLC addresses are shown in Modbus register format). In many cases the tag name editor will be able to display corresponding PLC memory points in the same manner as they appear in the PLC programming editor software (e.g., `I:5/10`, `SM0.4`, `C11`, etc.).

When programming large, complex systems it is very beneficial to define all the necessary tag names *before* beginning to lay out graphics in an HMI. The same goes for PLC programming: it makes the whole project go faster with less confusion time is spent defining all the necessary I/O points (and tag names, if the PLC programming software supports tag names in the programming environment) before creating any code specifying how those inputs and outputs will relate to each other.

Maintaining a consistent convention for tag names is important, too. For example, some programmers begin the tag name of every hard-wired I/O point as either `INPUT` or `OUTPUT` (e.g., `INPUT_PRESSURE_SWITCH_HIGH`, `OUTPUT_SHAKER_MOTOR_RUN`, etc.). The reason for maintaining a strict naming convention is not obvious at first, since the whole point of tag names is to give the programmer the freedom to assign *arbitrary names* to data points in the system. However, most tag name editors list the tags in alphabetical order, which means a naming convention organized in this way will present all the input tags contiguously (adjacent) in the list, all the output tags contiguously in the list, and so on.

Another way to leverage the alphabetical listing of tag names is to begin each tag name with a word describing its association to a major piece of equipment. Figure 24.78 is an example of a process with several data points defined in a PLC control system and displayed in an HMI.

Figure 24.78: An HMI running on an industrial operator interface computer

Listing all these tags in alphabetical order, the association is immediately obvious:

- `Exchanger_effluent_pump`
- `Exchanger_effluent_temp_out`
- `Exchanger_preheat_pump`
- `Exchanger_preheat_temp_in`
- `Exchanger_preheat_valve`
- `Reactor_bed_temp`

- `Reactor_feed_flow`
- `Reactor_feed_temp`
- `Reactor_jacket_valve`

As can be seen from this tag name list, all the tags directly associated with the heat exchanger are located in one contiguous group, and all the tags directly associated with the reactor are located in the next contiguous group. In this way, judicious naming of tags serves to group them in hierarchical fashion, making them easy for the programmer to locate at any future time in the tag name database.

Note that all the tag names shown here lack space characters between words (e.g., instead of "`Reactor bed temp`", a tag name should use hyphens or underscore marks as spacing characters: "`Reactor_bed_temp`"), since spaces are generally assumed by computer programming languages to be delimiters (separators between different variables).

Like programmable logic controllers themselves, the capabilities of HMIs have been steadily increasing while their price decreases. Modern HMIs support graphic trending, data archival, advanced alarming, and even web server ability allowing other computers to easily access certain data over wide-area networks. The ability of HMIs to log data over long periods of time relieves the PLC of having to do this task, which is very memory-intensive. This way, the PLC merely "serves" current data to the HMI, and the HMI is able to keep a record of current and past data using its vastly larger memory reserves[31].

[31] If the HMI is based on a personal computer platform (e.g., Rockwell FactoryTalk, Wonderware, FIX/Intellution software), it may even be equipped with a hard disk drive for enormous amounts of historical data storage.

# References


[1] F. D. Petruzella, Programmable Logic Controllers, 3rd ed. Boston, MA: McGraw-Hill, 2005.

[2] K. T. Erickson, Programmable Logic Controllers: An Emphasis on Design and Application. Rolla, MO: Dogwood Valley Press, 2005.

[3] (2010) Logix5000 Controllers Sequential Function Charts Programming Manual. 1756-PM006D-EN-P. Rockwell Automation. Milwaukee, WI.

[4] International Standard, Programmable Controllers – Part 3: Programming Languages, International Electrotechnical Commission Std. IEC 61 131-3, 2003.

# Chapter 25

# Analytical Control Design

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: rise time, settling time, percent overshoot, steady-state error
2. Describe how a controller affects the response of a system
3. State different types of performance criteria used to evaluate controller performance
4. List the qualities desired in a tracking system and compare and contrast them to the desirable quantities in a regulating system
5. Design controllers using loop shaping
6. Implement controllers in analog or digital form

## 25.1 Controller design goals

Controllers are designed to make a system meet certain performance criteria or specifications in response to certain input signals. These performance criteria typically include:

- time domain criteria (e.g., rise time, overshoot, settling time, steady-state error, control effort)
- frequency domain stability criteria (i.e., gain margin and phase margin)

These goals are accomplished by adding negative feedback, and often, by adding poles, zeros, and gain into a control loop in the form of a controller or compensator. When a controller is functioning in a closed-loop negative feedback system, it changes the location of the closed loop poles, which changes the frequency response and manipulates the system's time-domain response. This chapter focuses on controller design methods for single-input, single-output (SISO) systems.

## 25.2 Performance evaluation

### 25.2.1 Tracking vs. regulating controllers

Chapter 23 dealt with tuning PID controllers. Evaluation of the quality of the tuning is often made by putting step changes into the setpoint of the controller. This type of evaluation makes sense if the controller's goal is to track a changing setpoint, hence this is the job of a *tracking controller*. Motion control systems and temperature control systems with ramp-and-soak cycles are examples of systems that require tracking controllers. However, in many processes, the goal of the controller is not to track a changing setpoint, but rather to keep the process variable at a constant setpoint by rejecting disturbances and noise. This is the job of a *regulating controller*. In order to properly evaluate the performance of a regulating controller, a controlled disturbance must be provided into the system. For example, if a DC-DC converter were regulating the output voltage, increasing the output current by adding a resistive load in parallel with the attached load will cause the output voltage to drop due to loading effects. The controller can then respond to the drop in voltage by adjusting the duty cycle of the converter to return the output voltage to its regulated value. However, since it is often very difficult to provide controlled disturbances into a system, the performance of regulating controllers is more difficult to judge than that of tracking controllers.

### 25.2.2 Time domain criteria

With the exception of steady-state error, all of the following time domain criteria apply only to step responses for self-regulating systems. MATLAB can provide all of these measurements except steady-state error by using the `stepinfo` command on a system object.

#### Rise time

Rise time ($t_r$) is defined by different authors in different way, however, for the purposes of this text, rise time will be defined as the time it takes the unit step response to rise from 10% to 90% of the steady state value of thre response (this happens to be the default MATLAB setting range for rise time measurement). For a first order system or first-order dominant system, this definition of rise time corresponds to:

$$t_r = 2.1972\tau$$

Consider the second order underdamped system with the transfer function:

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

For this transfer function with $0.3 \leq \zeta \leq 0.8$, the rise time corresponds to approximately [1]:

$$\frac{2.16\zeta + 0.6}{\omega_n}$$

The shortest rise time possible is a desirable characteristic.

**Settling time**

Settling time is also defined by different authors in different ways. For the purposes of this text, settling time will be defined as the time it takes the response to settle to within 2% of its final value (this happens to be the default MATLAB setting range for settling time measurement). For a first order system or first-order dominant system, this definition of settling time corresponds to:

$$t_s = 4\tau$$

For the second order underdamped system transfer function previously discussed, the settling time is approximated by:

$$t_s = \frac{4}{\zeta\omega_n}$$

The shortest settling time possible is a desirable characteristic.

**Percent overshoot**

Overshoot only occurs for systems that have a dominant second-order underdamped response. For these types of systems, the percent overshoot ($P.O.$) is defined by:

$$P.O. = 100\% \times \frac{M_p - M_f}{M_f}$$

where $M_p$ is the peak magnitude and $M_f$ is the final magnitude. Percent overshoot is only affected by the damping ratio:

$$P.O. = 100e^{-\zeta\pi/\sqrt{1-\zeta^2}}$$

The peak magnitude value can be found from:

$$M_p = 1 + e^{-\zeta\pi/\sqrt{1-\zeta^2}}$$

In many control systems, overshoot is not allowable at all. For example, there shouldn't be any overshoot if a controller is trying to land a passenger airplane, or the plane will slam into the ground as opposed to landing smoothly. For some control systems, overshoot is allowed but must be limited to a certain percentage. Usually, the smallest overshoot possible is a desirable characteristic, but this conflicts with the desire to minimize rise time and settling time since rise time and settling time are inversely related to overshoot.

#### Steady-state error

The steady-state error ($e_{ss}$) is defined as the difference between the desired final value of the response and the actual final value. It can also be converted to a percent steady-state error value ($pe_{ss}$):

$$pe_{ss} = 100 \times \frac{desired\ value - actual\ value}{desired\ value}$$

For stable systems, the actual final value of the response can often be found using the final value theorem from Laplace transform theory.

#### Control effort

The speed of response (e.g., rise time, settling time) is directly related to the amount of control effort (i.e., power) that must be put into the system. This means if a system's response must move rapidly, large amounts of control effort must be used. This can be a problem since final control elements saturate, e.g., a valve only has a range of 0-100% open, or an amplifier can only put out $\pm 10$ volts. It behooves a control engineer to pay attention to the value of the control effort when designing a controller for a system, especially if computer-assisted design and simulation are being used as the computer often does not limit control effort to the actual range available from a final control element.

#### Integral performance indices

Assume two controllers are designed that can meet the same performance specification for a system. How can the two controllers be compared to determine which is better? Various types of integral performance indices may be used to determine a single number that is a figure of merit for each controller. Integral performance indices work well if one of the following two conditions exist for each of the controller structures:

1. If the system response indicates that the closed-loop system has zero steady-state error, or
2. If system responses can be obtained by running the identical experiment or simulation for the same length of time

By minimizing an integral figure of merit obtained for each controller structure, the performance of each controller can be objectively compared. The following integral performance indices are often used:

- Integral of the absolute error - $IAE = \int_0^\infty |e| dt$
- Integral of the squared error - $ISE = \int_0^\infty e^2 dt$
- Integral of the time-weighted absolute error - $ITAE = \int_0^\infty |te| dt$
- Integral of the time-weighted squared error - $ITSE = \int_0^\infty te^2 dt$

The MATLAB `trapz` command can be used to perform the integration once the various signals are generated.

### 25.2.3 Frequency domain criteria

Since a stable system has its poles in the open left-half of the complex frequency plane, the point at which a system is first considered unstable is when the poles are located on the imaginary axis. In this condition, a finite amount of energy added to the system creates an output that oscillates in a

perfect sinusoidal fashion. This occurs in a negative feedback loop when the total gain around the loop is unity (0 dB) *and* when the feedback is in phase, that is, the total phase shift around the loop is ±360°. Since the negative sign involved in producing negative feedback provides a phase shift of -180°, if the remaining components produce exactly -180° of phase shift at unity gain, the system will become unstable. These two criteria for oscillation, known as the Barkhausen criteria, lead to definitions for gain margin and phase margin.

### Gain margin

If the open-loop system has -180°phase shift, its gain must not be allowed to reach or exceed 0 dB or the system will become unstable. Gain margin (GM) refers to the gain (in dB) that would need to be added to reach 0 dB at the frequency where the phase crosses through -180°. By designating the phase crossover frequency as $\omega_{pc}$, the gain margin is calculated by:

$$GM = -M|_{\omega_{pc}}$$

### Phase margin

If the open-loop system has unity gain, the phase shift must not be allowed to reach or exceed -180° or the system will become unstable. Phase margin (PM) refers to the phase (in degrees) that would need to be added to reach -180° at the frequency where the gain crosses through 0 dB. By designating the gain crossover frequency as $\omega_{gc}$, the phase margin is calculated by:

$$PM = -180^\circ - \phi|_{\omega_{gc}}$$

### Examples of gain and phase margin

If either gain margin or phase margin is negative, the system will not be stable when the loop is closed with negative feedback. Figure 25.1 shows the Bode plot of the system with open-loop transfer function:

$$G(s) = \frac{100}{s^3 + 3s^2 + 3s + 1}$$

Figure 25.1: Bode plot of the example third-order system

Figure 25.2 illustrates the concepts of gain margin and phase margin for that system.

Figure 25.2: Bode plot showing gain and phase margins

## 25.3 Controller design methods

### 25.3.1 Root locus design

The goal of root locus design is to place the closed-loop poles of the SISO system in the proper location to obtain the desired time domain response specifications. This is done by changing the gain of a proportional controller if the desired closed-loop pole locations lie on the existing root locus. If the root locus does not pass through the desired closed-loop pole locations, the shape of the root locus can be altered by adding poles and zeros to the controller so that it does.

### 25.3.2 Frequency response design using loop shaping

The output of a system $G$ subjected to a reference input $R$, an output disturbance $D$ and a measurement noise $N$ and acted upon by unity feedback control with controller $C$ is given in the frequency domain by:

$$Y = \frac{CG}{1+CG}R + \frac{1}{1+CG}D - \frac{CG}{1+CG}N$$

There are special names for the coefficients of each of the inputs. The coefficient of $D$ is called the sensitivity function $S$. The coefficient of $R$ and $N$ is called the complementary sensitivity function $T$. It follows that $S + T = 1$. Note that $S$ and $T$ are frequency-dependent functions, though not explicity written in terms of $s$. Thus, for good disturbance rejection, it is desirable to have the sensitivity function $S$ be close to zero for the range of frequencies where the disturbances are likely to occur (typically at low frequencies). For good tracking, it is desirable to have the complementary sensitivity function $T$ be close to one for the range of frequencies where the reference inputs are likely to occur (typically at low frequencies). For good noise rejection, it is desirable to have the complementary sensitivity function $T$ be close to zero for the range of frequencies where the noise is likely to occur (typically at high frequencies).

Let $S$ and $T$ be parameterized in terms of the product of $CG$ (which will be called the open-loop gain $L$, i.e., $L = CG$), which is also a function of complex frequency $s$. Thus, $L$ should be large at low frequencies to track changing reference inputs and reject disturbances. In order to keep adequate stability margins, $L$ should cross 0 dB on a Bode plot at a slope of no more than -40 dB/decade (though -20 dB/decade is preferred). In order to have a fast time-domain response, the frequency at which $L$ crosses through 0 dB should be large, in other words, $L$ should have a large bandwidth. Finally, $L$ must be small at high frequencies to reject noise, and should roll off as steeply as possible after crossing through 0 dB. Since the plant transfer function $G$ is typically unable to be manipulated, the controller transfer function $C$ must be chosen to shape $L$ as desired. This method of controller design is known as *loop-shaping*.

#### Loop-shaping steady-state error for non-integrating controllers

Assuming a unity feedback system, the transfer function from the reference input $R$ to the error $E$ can be shown to be:

$$
\begin{aligned}
T_{ER} &= \frac{1}{1+CG} \\
&= \frac{1}{1+L}
\end{aligned}
$$

For stable loops, the magnitude of the steady state error $|E_{ss}|$ to a unit step function can be determined using the final value theorem:

$$
\begin{aligned}
|E_{ss}| &= \lim_{s\to 0} s\frac{1}{1+L}\frac{1}{s} \\
&= \lim_{s\to 0} \frac{1}{1+L}
\end{aligned}
$$

As $s \to 0$, the magnitude of $L$ approaches the dc gain value. Therefore,

$$
|E_{ss}| = \frac{1}{1+|L|_{dc}}
$$

Rearranging to solve for the dc gain of $L$ that will result in a given steady-state error gives:

$$
|L|_{dc} = \frac{1-|E_{ss}|}{|E_{ss}|}
$$

**Example 25.28** A controller is being designed using loop-shaping. What should the DC gain of the open-loop gain $L$ be (in dB) to achieve a steady-state error of no more than 5% for a unit step input?

**Answer to Example 25.28** Using

$$
|L|_{dc} = \frac{1-|E_{ss}|}{|E_{ss}|}
$$

and substituting 0.05 for $|E_{ss}|$ gives an answer of $|L|_{dc} = 19$. Converting to decibels yields $|L|_{dc,dB}$ = 25.58 dB. Therefore, the open loop gain dc value must be 25.58 dB to achieve a steady-state error of 5%. Since the system transfer function is fixed, the controller transfer function will need to provide the required gain.

## 25.4 Controller implementation

Controller algorithms may be implemented in either analog or digital form, though digital is now more common in most industries.

### 25.4.1 Analog compensator implementation

Analog controller transfer functions are implemented as operational amplifier circuits with resistors and capacitors used to produce the poles and zeros.

| **Type** | **Advantages** | **Disadvantages** |
|---|---|---|
| Proportional (P) | Simplest, improves response speed by increasing bandwidth | Has steady-state error to step disturbance in input or load |
| Lead | Reduces steady-state error, improves stability margins, improves response speed by increasing bandwidth | Does not eliminate steady-state error, may produce large signals |
| Lag | Reduces steady-state error, improves noise rejection | Reduces stability margins |
| Lag-Lead | Reduces steady-state error, improves stability margins, improves response speed by increasing bandwidth, improves noise rejection | Does not eliminate steady-state error, more complex architecture |
| Proportional-Integral (PI) | Eliminates steady-state error to step disturbance in input or load | Reduces stability margins |
| Proportional-Integral-Derivative (PID) | Eliminates steady-state error to step disturbance, improves stability margins, works best with inverse response systems | Unless filtered, D term amplifies noise |

Table 25.1: Types of Simple Controllers

The analog active implementation for each type of simple controller transfer function listed in Table 25.1 is shown in Figure 25.3.

Generic Op-Amp Controller

| Controller | Input Impedance ($Z_i$) | Feedback Impedance ($Z_f$) | Transfer Function |
|---|---|---|---|
| Proportional | $R_1$ | $R_2$ | $K$<br><br>$K = -R_2/R_1$ |
| Lead | $R_1$, $R_2$, $C_1$ | $R_3$ | $\frac{K(s+a)}{(s+b)}$<br><br>$K = -R_3/R_1$<br>$b = 1/R_2 C_1$<br>$a = (1+R_2/R_1)b$ |
| Lag | $R_1$ | $R_2$, $R_3$, $C_1$ | $\frac{K(s+a)}{(s+b)}$<br><br>$K = -R_2/R_1$<br>$b = 1/R_3 C_1$<br>$a = (1+R_3/R_2)b$ |
| Lag-Lead | $R_1$, $R_2$, $C_1$ | $R_3$, $R_4$, $C_2$ | $\frac{K(s+a)(s+d)}{(s+b)(s+c)}$<br><br>$K = -R_3/R_1$<br>$b = 1/R_4 C_2$<br>$a = (1+R_4/R_3)b$<br>$d = 1/R_2/C_1$<br>$c = (1+R_2/R_1)d$<br>$a<d$ |
| Proportional-Integral | $R_1$ | $R_2$, $C_1$ | $\frac{K(s+a)}{s}$<br><br>$K = -R_2/R_1$<br>$a = 1/R_2 C_1$ |
| Proportional-Integral-Derivative (with Filter) | $R_1$, $C_1$, $R_2$ | $R_3$, $C_2$ | $\frac{K(s+a)(s+d)}{s(s+b)}$<br><br>$K = -R_3/R_2$<br>$a = 1/R_3 C_2$<br>$a = (R_1+R_2)/R_1 R_2 C_1$<br>$d = 1/R_1 C_1$ |

Figure 25.3: Analog controller implementations

### 25.4.2 Digital compensator implementation

Digital systems are both discretized in amplitude (quantized) and discretized in time (sampled). Fixed-point mathematical issues that used to exist in quantized systems are not as much of a problem anymore as most microcontrollers are now capable of floating point mathematics. However, sampling still presents issues in controller design. One common rule of thumb is to sample at least ten times faster than the highest natural frequency content of the system. Note that time constants have units of inverse frequency, so for systems that are to behave with a given dominant time constant $\tau$, the sampling period $T_s$ should be at least ten times smaller than $\tau$, i.e., the sampling frequency $\omega_s$ should be at least $\frac{10}{\tau}$.

Controllers designed for continuous systems can be easily converted to discrete-time forms in software packages such as MATLAB with the Control Systems Toolbox or Octave. If the system to be controlled will be controlled by a computer, the `c2d` instruction can be used to discretize the controller transfer function at the sampling rate required. As an example, consider a PI algorithm:

$$C(s) = \frac{2(s+4)}{s}$$

that is to be sampled every 5 ms. First, the controller is defined in MATLAB as a system object using either the `tf` or `zpk` commands, e.g., `Co = tf(2*[1 4],[1 0])`. The command `Cod = c2d(Co, 0.005)` will then discretize the controller with the required sampling rate, resulting in:

$$Cod = \frac{U(z)}{E(z)} = \frac{2z - 1.96}{z - 1}$$

This is a sampled-data transfer function, hence the discrete-time notation in $z$, the forward shift operator. To convert this to a backward shift operator $z^{-1}$, simply divide both numerator and denominator by $z$, which produces:

$$Cod = \frac{U(z^{-1})}{E(z^{-1})} = \frac{2 - 1.96z^{-1}}{1 - z^{-1}}$$

Cross-multiplying each side of the equation gives:

$$U(1 - z^{-1}) = E(2 - 1.96z^{-1})$$

Using the distributive property results in:

$$U - z^{-1}U = 2E - 1.96z^{-1}E$$

Finally, solving for the $U$ term that is by itself produces:

$$U = z^{-1}U + 2E - 1.96z^{-1}E$$

Now, using the inverse $z$-transform where $z^{-1}$ is a unit delay operator results in the following difference equation[1]:

$$U(n) = U(n-1) + 2E(n) - 1.96E(n-1)$$

[1] To find the difference equation, simply replace every $z^{-k}X$ with $X(n-k)$

In words, this equation says "the controller output at the current time step is the previous controller output plus twice the current error minus 1.96 times the previous error." Since this equation is simply composed of multiplication, addition, and subtraction operations, and requires the use of a previous value of controller output and previous value of error from memory, a computer easily handles it. At the end of each computation, the computer then stores the current value of controller output and error into the respective "previous" memory locations and waits for the next clock cycle.

The previous equation could be implemented as written, but may have issues due to actuator saturation. For example, if the controller output hits 100% in response to a nonzero error, the actuator saturates (e.g., the valve reaches 100% open). It this occurs before the error is driven to zero, the process variable will still not be able to change because there is no way to change the manipulated variable. The nonzero error continues to integrate in the controller designed previously, resulting in a larger and larger control effort above the 100% level. This is known as *reset wind-up*, and occurs in any controller that has an integral term. Should the error change signs, the actuator will still be stuck at its limit value until the controller output drops below the 100% level, and the process will not respond until the actuator can move. Therefore, reset wind-up results in poor control. Most modern digital controllers implement anti-reset wind-up techniques to avoid the problem, but many require the user to configure the controller to engage this option. One simple method of anti-reset wind-up simply stops adding the error to the integral term when the controller output reaches either 0% or 100%.

# References


[1] R. C. Dorf and R. H. Bishop, Modern Control Systems, 10th ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2005.

[2] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, Design of Feedback Control Systems, 4th ed. New York, NY: Oxford University Press, 2002.

[3] C. L. Phillips and R. D. Harbor, Feedback Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[4] J. R. Rowland, Linear Control Systems: Modeling, Analysis, and Design. New York, NY: John Wiley & Sons, 1986.

[5] F. J. Hale, Introduction to Control System Analysis and Design. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[6] A. P. Sage, Linear Systems Control. Champaign, IL: Matrix Publishers, 1978.

[7] M. Driels, Linear Control Systems Engineering. New York, NY: McGraw-Hill, 1995.

[8] W. J. Palm, Control Systems Engineering. New York, NY: John Wiley and Sons, 1986.

[9] S. C. Gupta and L. Hasdorff, Fundamentals of Automatic Control. New York, NY: John Wiley and Sons, 1970.

[10] A. F. D'Souza, Design of Control Systems. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[11] T. E. Marlin, Process Control, 2nd ed. New York, NY: McGraw-Hill, 2000.

[12] B. A. Ogunnaike and W. H. Ray, Process Dynamics, Modeling, and Control. New York, NY: Oxford University Press, 1994.

[13] A. Tewari, Modern Control Design with MATLAB and Simulink. New York, NY: John Wiley and Sons, 2002.

[14] C. A. Smith and A. Corripio, Principles and Practice of Automatic Process Control, 3rd ed. New York, NY: John Wiley and Sons, 2006.

[15] M. Gopal, Control Systems: Principles and Design. Boston, MA: McGraw-Hill, 2008.

# Chapter 26

# Multivariable Process Control Strategies

## Objectives

Upon completion of this chapter, the student shall be able to:

1. Define the following terms: multivariable (MIMO) systems, cascade control, feedforward control, ratio control, override control, split range control, relative gain array, closed-loop gain, open-loop gain, loop interaction, decoupling, remote set point, inner loop, outer loop, master controller, slave controller, static feedforward, dynamic feedforward, ratio control, override (constraint) control, selective control
2. Sketch block diagrams or instrumentation diagrams for the following compensator types: cascade, feedforward, ratio, override, split range
3. Identify opportunities to use multivariable control techniques
4. Design compensators for servo and regulator systems using cascade, feedforward, ratio, override, and split range controllers to achieve performance criteria
5. Explain why feedforward control cannot be effectively used without feedback control
6. Calculate the relative gain array (RGA) of a MIMO system and determine controlled-manipulated variable pairings
7. Explain the importance of decoupling as it applies to compensator design and design decouplers

In a simple control system, a process variable (PV) is measured and compared with a setpoint value (SP). A manipulated variable (MV, or output) signal is generated by the controller and sent to a final control element, which then influences the process variable to achieve stable control. The algorithm by which the controller develops its output signal is typically PID (Proportional-Integral-Derivative), but other algorithms may be used as well. A generic diagram for a feedback control loop is shown in Figure 26.1.

Figure 26.1: The standard feedback control loop

This form of simple control may be improved upon and expanded for a greater range of process applications by interconnecting multiple controllers and/or redirecting measurement and control signals in more complex arrangements. This chapter explores some of the more common multivariable process control system configurations.

## 26.1 Supervisory control

In a manually-controlled process, a human operator directly actuates some form of final control element (usually a valve) to influence a process variable. Simple automatic ("regulatory") control relieves human operators of the need to continually adjust final control elements by hand, replacing this task with the occasional adjustment of setpoint values. The controller then manipulates the final control element to hold the process variable at the setpoint value determined by the operator.

The next step in complexity after simple automatic control is to automate the adjustment of the setpoint for a process controller. A common implementation of this concept is the automatic cycling of setpoint values according to a timed schedule. An example of this is a temperature controller for

a heat-treatment furnace used to temper metal samples, an applicatuion shown in Figure 26.2.

Figure 26.2: Diagram showing feedback control of an arc furnace with an external setpoint adjustment

Here, a computer communicates setpoint values to the temperature indicating controller (TIC) over a digital network interface such as Ethernet. From the temperature controller's perspective, this is a *remote* setpoint signal, as opposed to a *local* setpoint value which would be set by a human operator at the controller faceplate. Since the heat-treatment of metals requires particular temperature ranges and rates of change over time, this control system relieves the human operator of having to manually adjust setpoint values again and again during heat-treatment cycles. Instead, the computer schedules different setpoint values at different times (even setpoint values that change steadily at a certain rate over a period of time) according to the needs of the particular metal type and treatment type. Such a control scheme is quite common for heat-treating processes, and it is referred to as *ramp and soak*[1].

Supervisory setpoint control is also used in the chemical processing industries to optimize production by having a powerful computer provide setpoint adjustments to regulatory controls based on mathematical models of the process and optimization constraints. Dividing the control system into two layers – the upper-layer supervisory control computer and the lower-layer process controllers – provides a flexible means of achieving basic control and advanced (optimization) controls without burdening any one piece of control hardware with too much responsibility over the process. If the supervisory computer fails for whatever reason, for example, the regulatory controls (either panel-mounted controllers or more likely a distributed control system) may default to local setpoint control.

Such optimization control systems are usually built over a digital network for reasons of convenience. A single network cable not only is able to handle the frequent setpoint changes from

[1] In honor of the system's ability to slowly "ramp" temperature up or down at a specified rate, then "soak" the metal at a constant temperature for set periods of time. Many single-loop process controllers have the ability to perform ramp-and-soak setpoint scheduling without the need of an external "supervisory" computer.

the supervisory computer to the multitude of process loop controllers, but it may also carry process variable information from those controllers back to the supervisory computer so it has data for its optimization algorithms to operate on, as shown in Figure 26.3.

Figure 26.3: Diagram showing a SCADA application used for external setpoint adjustment for multiple controllers

The complexity of these optimization algorithms is limited only by the computational power of the supervisory computer and the creativity of the programmers and engineers who implement it. A modern trend in process optimization for industries able to produce varying proportions of different products from the same raw material feed is to have computer algorithms select and optimize production not only for maximum cost efficiency, but also for maximum market sales and minimum storage of volatile product[2].

[2] A meeting of industry representatives occured where one person talked at length about a highly-automated lumber mill where logs were cut into lumber not only according to minimum waste, but also according to the real-time market value of different board types and stored inventory. The joke was, if the market value of wooden toothpicks suddenly spiked up, the control system would shred every log into toothpicks in an attempt to maximize profit!

## 26.2 Split-range control

There are many process control applications in industry where it is desirable to have multiple control valves respond to the output of a common controller. Control valves configured to follow the command of the same controller are said to be *split-ranged*, or *sequenced*.

Split-ranged control valves may take different forms of sequencing. A few different modes of control valve sequencing are commonly seen in industry: *complementary*, *exclusive*, and *progressive*[3].

### 26.2.1 Complementary valve sequencing

The first is a mode where two valves serve to proportion a mixture of two fluid streams, such as the example in Figure 26.4 where base and pigment liquids are mixed together to form colored paint.

Figure 26.4: A complementary valve control scheme

Both base and pigment valves operate from the same 3 to 15 psig pneumatic signal output by the I/P transducer (AY), but one of the valves is Air-To-Open while the other is Air-To-Close. The following table shows the relationship between valve opening for each control valve and the controller's output:

[3]Tony Kuphaldt searched in vain for standardized names to categorize different forms of control valve sequencing. The names "complementary," "exclusive," and "progressive" are his own invention. If he missed someone else's categorization of split-ranging in his research, he sincerely apologizes.

| Controller output (%) | I/P output (psig) | Pigment valve (stem position) | Base valve (stem position) |
|---|---|---|---|
| 0 % | 3 psig | fully shut | fully open |
| 25 % | 6 psig | 25% open | 75% open |
| 50 % | 9 psig | half-open | half-open |
| 75 % | 12 psig | 75% open | 25% open |
| 100 % | 15 psig | fully open | fully shut |

An alternative expression for this split-range valve behavior is a graph showing each valve opening as a colored stripe of varying width (wider representing further open). For this particular mode of split-ranging, the graph would look like this:

Figure 26.5: Illustration of valve position using a complementary scheme

With this form of split-ranging, there is never a condition in the controller's output range where both valves are fully open or fully shut. Rather, each valve *complements* the other's position[4].

### 26.2.2 Exclusive valve sequencing

Other applications for split-ranged control valves call for a form of valve sequencing where both valves are fully closed at a 50% controller output signal, with one valve opening fully as the controller output drives toward 100% and the other valve opening fully as the controller output goes to 0%. The nature of this valve sequencing is to have an "either-or" throttled path for process fluid. That is, *either* process fluid flows through one valve *or* through the other, but never through both at the same time.

A practical example of this form of split-ranging is in reagent feed to a pH neutralization process as shown in Figure 26.6, where the pH value of process liquid is brought closer to neutral by the addition of either acid or caustic.

[4]In mathematics, a "complement" is a value whose sum with another quantity always results in a fixed total. Complementary angles, for instance, always add to $90^o$ (a right angle).

Figure 26.6: An exclusive split-range valve control scheme

Here, a pH analyzer monitors the pH value of the mixture and a single pH controller commands two reagent valves to open when needed. If the process pH begins to increase, the controller output signal increases as well (direct action) to open up the acid valve. The addition of acid to the mixture will have the effect of lowering the mixture's pH value. Conversely, if the process pH begins to decrease, the controller output signal will decrease as well, closing the acid valve and opening the caustic valve. The addition of caustic to the mixture will have the effect of raising the mixture's pH value.

Both reagent control valves operate from the same 3 to 15 psig pneumatic signal output by the I/P transducer (AY), but the two valves' calibrated ranges are not the same. The Air-To-Open acid valve has an operating range of 9 to 15 psig, while the Air-To-Close caustic valve has an operating range of 9 to 3 psig. The following table shows the relationship between valve opening for each control valve and the controller's output:

| Controller output (%) | I/P output (psig) | Acid valve (stem position) | Caustic valve (stem position) |
|---|---|---|---|
| 0 % | 3 psig | fully shut | fully open |
| 25 % | 6 psig | fully shut | half-open |
| 50 % | 9 psig | fully shut | fully shut |
| 75 % | 12 psig | half-open | fully shut |
| 100 % | 15 psig | fully open | fully shut |

Again, the two valves' exclusive relationship may be expressed in the form of a graph as seen in Figure 26.7, with colored stripes representing valve opening.

Figure 26.7: Illustration of valve position using an exclusive split-range scheme

Exclusive-sequenced control valves are used in applications where it would be undesirable to have both valves open simultaneously. In the example given of a pH neutralization process, the goal here is for the controller to be able to call forth either acid reagent or caustic reagent to "push" the pH value either direction as needed. However, simultaneously adding both acid and caustic to the process would be wasteful, as one reagent would simply neutralize the other with no benefit to the process liquid itself.

### 26.2.3 Progressive valve sequencing

A third form of control valve sequencing is used to expand the operating range of flow control for some fluid beyond that which a single control valve could muster. Once again pH control provides a suitable example to illustrate an application of this form of sequencing.

pH is an especially challenging application of process control because the dynamic range of the process is enormous. Each unit of pH value change represents a *ten-fold* change in hydrogen ion concentration within the process liquid. This means the difference in ion concentration between a process liquid having a value of 10 pH and a process liquid having a value of 7 pH is a factor of *one thousand!* Consequently, the flow rate of reagent necessary to neutralize a process liquid stream may vary widely. It is quite possible that a control valve sized to handle minimum flow will simply be too small to meet the demands of high flow when needed. Yet, a control valve sized large enough to meet the maximum flow rate may be too large to precisely "turn down" when just a trickle of reagent is needed.

This general control problem was encountered by automotive engineers in the days when *carburetors* were used to mix gasoline with air prior to combustion in an engine. A carburetor is a mechanical air flow control device using a "butterfly" valve element to throttle air flow into the engine, and a venturi element producing vacuum to aspirate fuel droplets into the air stream to create an air-fuel mixture. A carburetor with a butterfly valve and flow tube sized to idle well and respond to the needs of in-town driving would not flow enough air to provide high-end performance. Conversely, a large carburetor suitable for performance driving would be almost uncontrollable for low-speed and idling operation. Their solution to this problem was the *progressive carburetor*, having two butterfly valves to throttle the flow of air into the engine. One butterfly valve handled low amounts of air flow only, while a larger butterfly valve opened up only when the accelerator

pedal was nearly at its maximum position. The combination of two differently-sized butterfly valves – progressively opened – gave drivers the best of both worlds. Now, an automobile engine could perform well both at low power levels and at high power levels.

On a fundamental level, the problem faced in pH control as well as by early automotive engineers is the same thing: insufficient *rangeability*. Some processes demand a greater range of control than any single valve can deliver, and it is within these processes that a pair of progressively-sequenced control valves is a valid solution[5].

Figure 26.8 shows this solution applied to a pH control process where the incoming liquid always has a high pH value and must be neutralized with acid.

Figure 26.8: Neutralization using a progressive scheme

Assume linear valve characteristics and that the small valve has a $C_v$ of 10 and the large valve has a $C_v$ of 100. Proper sequencing of the small and large acid control valves is shown in the table and the graph:

[5]This scheme is sometimes called a 1/3-2/3 scheme, where the smaller valve handles up to 1/3 of the total flow and the larger valve handles the remaining 2/3 (whether or not the split is at 1/3, the scheme usually retains the name).

| Controller output (%) | I/P output (psig) | Small acid valve (stem position) | Large acid valve (stem position) |
|---|---|---|---|
| 0 % | 3 psig | fully shut | fully shut |
| 5 % | 3.6 psig | half-open | fully shut |
| 10 % | 4.2 psig | fully open | fully shut |
| 60 % | 10.2 psig | fully open | half-open |
| 100 % | 15 psig | fully open | fully open |

Figure 26.9: Illustration of valve position using a progressive scheme

With the two acid control valves sequenced progressively, the control system will have sufficient rangeability to handle widely varying process conditions. [6]

### 26.2.4 Valve sequencing methods

In all previous control valve sequencing examples shown, both control valves received the same pneumatic signal from a common I/P (current-to-pressure) converter. Sequencing between the two valves was a matter of proper bench-set pressure ranges. The common pneumatic signal approach (one controller, one I/P transducer) is simple but suffers from the disadvantage of slow response, since one I/P transducer must drive two pneumatic actuators. Response time may be improved by adding a pneumatic *volume booster* between the I/P and the valve actuators, or by adding a positioner to at least one of the valves. Either of these solutions works by the same principle: reducing the air volume demand on the one I/P transducer.

Several alternative methods exist for control valve sequencing, as shown in Figure 26.10.

[6] Note that the sequencing scheme shown above applies to valves with a linear characteristic; for proper progressive sequencing of equal percentage valves, consult Béla Lipták's "Instrument Engineers Handbook Volume 2: Process Control and Optimization."

Figure 26.10: Methods of sequencing control valves

Wiring two I/P transducers in series so they share the exact same current is another way to split-range two control valves. This approach does not suffer from slow response, since each valve has its own dedicated I/P transducer to supply it with actuating air. There is now a choice where to implement the split ranges: it can be done in each of the I/P transducers (with non-standard I/P calibrations) *or* in the valve bench-set ranges as before. Since it is generally easier to re-range an I/P than it is to rebuild a control valve with a different spring (to give it a different actuating pressure range), this approach has the advantage of convenient configuration. A disadvantage of this approach is the extra demand placed on the controller's output signal circuitry: one must be careful to ensure the two series-connected I/P converters do not drop too much voltage at full current or else the controller may have difficulty driving both in series. Another (potential) disadvantage of series-connected valve devices in one current loop is the inability to install "smart" instruments communicating with the HART protocol, since multiple devices on the same loop will experience address conflicts[7].

A very common way to implement split-ranging is to use a controller with multiple 4-20 mA outputs. This is very easy to do if the controller is part of a large system (e.g., a DCS or a PLC with multiple output channels). Now, each valve has its own dedicated wire pair for control. A further advantage of dual controller outputs is the ability to perform the split-range sequencing within the controller itself, which is often easier than re-ranging an I/P or calibrating a valve positioner.

Dual controllers are an option only for specialized applications requiring different degrees of responsiveness for each valve, usually for exclusive or progressive split-ranging applications only. Care must be taken to ensure the controllers' output signals do not wander outside of their intended ranges, or that the controllers do not begin to "fight" each other in trying to control the same process variable[8].

An important consideration – and one that is easily overlooked – in split-range valve systems is *fail-safe mode*. As discussed in a previous section of this chapter (on page 679), the basis of fail-safe control system design is that the control valve(s) must be chosen to fail in the mode that is safest for the process in the event of actuating power loss or control signal loss. The actions of all other instruments in the loop should then be selected to complement the valves' natural operating mode.

In control systems where valves are split-ranged in either complementary or exclusive fashion, one control valve will be fully closed and the other will be fully open at each extreme end of the signal range (e.g., at 4 mA and at 20 mA). If the control valves are driven by the same controller signal, the failure modes of the two valves must likewise be opposite each other: one will fail open while the other fails closed if the signal goes dead or if air pressure is lost. However, if it is deemed safer for the process to have the two valves fail in the same state – for example, to both fail closed in the event of air pressure or signal loss – it is still possible to sequence them for complementary or exclusive control action by driving the two valves with different output signals. In other words, split-ranging two control valves so they normally behave in opposite fashion does *not* necessarily mean the two valves must fail in opposite states.

[7] Although the HART standard does support "multidrop" mode where multiple devices exist on the same current loop, this mode is digital-only with no analog signal support. Not only do many host systems not support HART multidrop mode, but the relatively slow data communication rate of HART makes this choice unwise for most process control applications. If analog control of multiple HART valve positioner devices from the same 4-20 mA signal is desired, the address conflict problem may be resolved through the use of one or more *isolator* devices, allowing all devices to share the same analog current signal but isolating each other from HART signals.

[8] Both controllers should be equipped with provisions for reset windup control (or have no integral action at all), such that the output signal values are predictable enough that they behave as a synchronized pair rather than as two separate controllers.

As an example, consider the temperature control system supplying either hot water or chilled water to a "jacket" surrounding a biofuel reactor vessel shown in Figure 26.11. The purpose of this system is to control temperature within the reactor to a constant setpoint value, regardless of the chemical reaction's thermal properties. If the reaction inside the vessel is *exothermic* (releasing heat), the control system will respond by sending chilled water to the jacket to remove that heat. If the reaction inside the vessel is *endothermic* (absorbing heat), the control system will respond with hot water to the jacket to add heat. Chemical piping in and out of the reactor vessel has been omitted from this P&ID for simplicity to allow focus just on the reactor's temperature control system:

Figure 26.11: An implementation of split-range exclusive sequencing

Here, the controller has been configured for dual-output operation, where the output value drives two identical 4-20 mA signals to the control valve positioners, which directly input the current signals from the controller without the need for I/P transducers in between. The hot water valve (TV-37a) is fail-closed (FC) while the cold water valve (TV-37b) is fail-open (FO). Half-range positioner calibrations provide the exclusive sequencing necessary to ensure the two valves are never open simultaneously – TV-37b operates on the lower half of the 4-20 mA signal range (4-12 mA), while TV-37a operates on the upper half (12-20 mA)[9].

[9]Instead of using half-range positioner calibrations, an alternative approach is to scale the output of the controller

Consider the effects from the controller (TIC-37) losing power. Both 4-20 mA signals will go dead, driving both valves to their fail-safe modes: hot water valve TV-37a will fully close, while cold water valve TV-37b will fully open. Now consider the effects of air pressure loss to both valves. With no air pressure to operate, the actuators will spring-return to their fail-safe modes: once again hot water valve TV-37a will fully close, while cold water valve TV-37b will fully open. In both failure events, the two control valves assume consistent states, ensuring the reactor will cool down rather than heat up.

Now imagine someone reconfigures the system, using identical control valves (signal-to-open, fail-closed) for both hot and cold water supply, and a different program in the controller to exclusively sequence two different 4-20 mA current signals as seen in Figure 26.12.

Figure 26.12: A different implementation of split-range exclusive sequencing

Consider the effects from the controller (TIC-37) losing power. Both 4-20 mA signals will go dead, driving both valves to their fail-safe modes: fully closed. Now consider the effects of air pressure loss to both valves. With no air pressure to operate, the actuators will spring-return to their fail-safe modes: once again both control valves fully close. In both failure events, the two control valves assume consistent states where the reactor is neither heated nor cooled, but rather

such that for 0% to 50% output it sends a 4-20 mA signal to the chilled water valve and for 50% to 100% output it sends a 4-20 mA signal to the hot water valve. This means that the control computer executes the split-range, not the pneumatic positioners, which is a much more accurate and drift-free approach.

left to assume its own temperature. The failure modes of both valves are still consistent regardless of the nature of the fault, but note how this scheme allows both valves to fail in the same mode if that is what is deemed safest for the process. However, if the reaction is exothermic, temperature may increase until a fire or explosion occurs.

As with all fail-safe system designs, begin by choosing the proper fail-safe mode for each control valve *as determined by the nature of the process*, not by what is considered the simplest or easiest-to-understand instrument configurations. Only after each valve's failure mode has been chosen is the rest of the system designed. This includes split-range sequencing: where and how to sequence the valve operation is a decision to be made only after the valves' natural fail-safe states are chosen based on the needs of process safety.

## 26.3 Cascade control

A simple control system drawn in block diagram form looks Figure 26.13.

Figure 26.13: The standard feedback control loop

Information from the measuring device goes to the controller, which then manipulates the final control device to influence the process which is sensed again by the measuring device. The controller's task is to inject the proper amount of negative feedback such that the process variable stabilizes over time. This flow of information is collectively referred to as a feedback control *loop*.

External disturbance influences can affect the performance of a system. If the disturbances affect the control effort that is influencing the process, the process does not behave as it should. As an example, consider the heat exchanger in Figure 26.14.

Figure 26.14: A temperature control loop on a heat exchanger

The sensed temperature is fed to a controller which adjusts the valve, thereby controlling the outlet temperature by manipulating the rate of energy provided to the heat exchanger by adjusting the steam flow. The formula for this (all variables on the right side of the equation pertain to the steam) is:

$$\dot{Q} = \dot{m}c_p \left(T_{in} - T_{out}\right)$$

Thus, if the inlet steam temperature changes or if the mass flow rate of the steam changes (e.g., if the steam system pressure changes so do the system flows), the rate of heat energy provided by the steam to the heat exchanger changes. Therefore, a disturbance load occurs on the control effort to the heat exchanger when either of these changes occur. Figure 26.15 illustrates this case in block diagram form.

Figure 26.15: The standard feedback control loop with disturbance on the control effort

If the disturbance on the control effort can be rapidly detected and corrected, the process output may not be affected very much. This calls for a *cascade control* scheme.

To *cascade* controllers means to connect the output signal of one controller to the setpoint of another controller, with each controller sensing a different aspect of the same process[10]. The first controller (called the *primary*, or *master*) essentially "gives orders" to the second controller (called the *secondary* or *slave*) via a *remote setpoint* signal.

Thus, a cascade control system consists of two feedback control loops, one nested inside the other, as shown in Figure 26.16.

[10]This is the process control definition of "cascade," which includes a second measurement function - the definition in the mechanical or electrical engineering control books of the term "cascade" is to simply connect two function blocks in series.

Figure 26.16: A cascade control system diagram

A more useful diagram is shown in Figure 26.17, which shows the master and slave loops and the remote setpoint being sent from the master controller to the slave controller.

Figure 26.17: A cascade control system block diagram

A very common example of cascade control is a *valve positioner*, which receives a command signal from a regular process controller, and in turn works to ensure the valve stem position precisely matches that command signal. The control valve's stem position is the process variable (PV) for the positioner, just as the command signal is the positioner's setpoint (SP). Valve positioners therefore act as "slave" controllers to "master" process controllers controlling pressure, temperature, flow, or some other process variable.

The purpose of cascade control is to achieve greater stability of the primary process variable by regulating a secondary process variable in accordance with the needs of the first. An essential requirement of cascaded control is that the secondary process variable be faster-responding (i.e., less lag time) than the primary process variable. A typical rule of thumb is that the secondary process dynamics must be at least five times faster than the primary process dynamics in order for cascade control to work effectively.

For example, consider dryer system shown in Figure 26.18 where heated air is used to force-evaporate water from a granular solid. The primary process variable is the outlet air exiting the dryer, which should be maintained at a high enough temperature to ensure water will not remain in the upper layers of the solid material. This outlet temperature is fairly slow to react, as the solid material mass creates a large lag time.

Figure 26.18: An attempt to use simple feedback control on a drying process

There are several parameters influencing the temperature of the outlet air other than the moisture content of the drying material. These include air flow, ambient air temperature, and variations in steam temperature. Each one of these variables is a *load* on the process variable (outlet air temperature). If any of these parameters were to suddenly change, the effect would be slow to

register at the outlet temperature even though there would be immediate impact at the bottom of the dryer where the heated air enters. Correspondingly, the control system would be slow to correct for any of these changing loads.

One way to help gain better control over this dryer system is to install a second temperature transmitter at the inlet duct of the dryer, with its own controller to adjust steam flow at the command of the primary controller, as shown in Figure 26.19.

Figure 26.19: Use of cascade control on a drying process

Now, if any of the loads related to incoming air flow or temperature were to change, the secondary controller (TC-1b) would *immediately* compensate by adjusting steam flow through the heat exchanger to maintain a constant air temperature entering the dryer. Thus, the "slave" control loop (1b) helps stabilize the "master" control loop (1a) by reacting to changes in one of the variables influencing it.

A helpful way to think of this is to consider the slave controller as *shielding* the master controller from the loads previously mentioned (incoming air flow, ambient temperature, and steam temperature). Of course, these variables still act as loads to the slave controller, as it must continuously adjust the steam valve to compensate for changes in air flow, ambient air temperature, and steam temperature. However, so long as the slave controller does a good job of stabilizing the air temperature entering the dryer, the master controller will never "see" the effects of those load changes. Responsibility for incoming air temperature has been delegated to the slave controller, and as a result the master controller is conveniently isolated from the loads impacting that loop.

A common implementation of cascade control is where a flow controller receives a setpoint from some other process controller (pressure, temperature, level, analytical, etc.), fluid flow being one of the fastest-responding process types in existence. A feedwater control system for a steam boiler – shown in Figure 26.20 in pneumatic form – is a good example.

Figure 26.20: Use of cascade control on a boiler water level control process

The "secondary" flow controller works to maintain feedwater flow to the boiler at whatever flow rate desired by the level controller. If feedwater pressure happens to increase or decrease, any resulting changes in flow will be quickly countered by the flow controller without the level controller having to act from a consequent change in steam drum water level. Thus, cascade control works to guard against steam drum level instability resulting from changes in the feedwater flow caused by factors outside of the system.

It is worth noting that the inclusion of a flow control "slave" loop to this boiler water level control system helps to overcome a potential problem of the control valve: nonlinear behavior. In Chapter 16, the phenomenon of of *installed valve characteristics* was explored (Section 16.3.11 on page 711), specifically noting how changes in pressure drop across a control valve influences its throttling behavior. The result of these pressure changes is a nonlinear valve response such that the valve tends to be more responsive near its closed position and less responsive near its open

position. One of the benefits of cascaded flow control is that this problem becomes confined to the secondary (flow control) loop, and is effectively removed from the primary control loop. To phrase it simply, distorted valve response becomes the flow controller's problem rather than something the level controller must manage. The result is a level control system with more predictable response.

An analogy for considering cascade control is that of *delegation* in a work environment. If a supervisor delegates some task to a subordinate, and that subordinate performs the task without further need of guidance or assistance from the supervisor, the supervisor's job is made easier. The subordinate takes care of all the little details that would otherwise burden the supervisor.

A necessary step in implementing cascade control is to ensure the secondary ("slave") controller is well-tuned *before* any attempt is made to tune the primary ("master") controller. Just a moment's thought is all that is needed to understand why this precedence in tuning must be: it is a simple matter of dependence. The slave controller does not depend on good tuning in the master controller in order to control the slave loop. If the master controller were placed in manual (effectively turning off its automatic response), the slave controller would simply control to a constant setpoint. However, the master controller most definitely depends on the slave controller being well-tuned in order to fulfill the master's "expectations." If the slave controller were placed in manual mode, the master controller would not be able to exert any control over its process variable whatsoever. Clearly then, the slave controller's response is essential to the master controller being able to control its process variable, therefore the slave controller must be the first one to be tuned.

## 26.4 Feedforward control

*Feedback* control works on the principle of information from the outlet of a process being "fed back" to the input of that process for corrective action. A block diagram of feedback control looks like a loop as shown in Figure 26.21.

Figure 26.21: The standard feedback control loop

The reason any control system is necessary at all[11] to maintain a process variable at some stable value is the existence of something called a *load* or *disturbance* – the terms will be used interchangeably. A "load" is some variable influencing a process that is not itself under direct control. It may be represented in the block diagram as an arrow entering the process, but it is not within the control loop, as shown in Figure 26.22.

[11] This statement is true only for self-regulating processes. Integrating and unstable processes require control systems to achieve stability even in the complete absence of any loads. However, since self-regulation typifies the vast majority of industrial processes, the fundamental purpose of most control systems is to counteract the effects of loads.

Figure 26.22: The standard feedback control loop showing loads influencing the process

For example, consider the problem of controlling the speed of an automobile. In this scenario, vehicle speed is the process variable being measured and controlled, while the final control device is the accelerator pedal controlling engine power output. If it were not for the existence of hills and valleys, head-winds and tail-winds, air temperature changes, road surface variations, and a host of other load variables affecting car speed, it would be an elementary matter to drive at a constant speed: simply hold the accelerator position constant.

However, the presence of these load variables makes it necessary for the human driver of the automobile (or a *cruise control* system) to continually adjust engine power in order to maintain constant speed regardless of load variations. Using the car's measured speed as feedback, the driver (or cruise control) adjusts the accelerator pedal position as necessary based on whether or not the car's speed matches the desired setpoint value.

An inherent weakness of any feedback control system is that it can never be *pro-active*. The best any feedback control system can ever do is make adjustments to a process *after* some disturbance in process variable is detected. This makes deviations from setpoint inevitable, if only for short periods of time. In the context of our automobile cruise control system, this means the car cannot maintain perfectly constant speed because the control system does not have the ability to anticipate hills, wind gusts, changes in air temperature, or changes in road surface.

*Feedforward* control addresses this weakness by taking a fundamentally different approach, basing final control decisions on the states of load variables rather than the process variable. In other words, a feedforward control system monitors all the factors influencing a process and decides how to compensate for these factors *ahead of time* before they have the opportunity to affect the process

variable. If all loads are accurately measured, and the control algorithm realistic enough to predict process response for these known load values, the process variable does not even need to be measured at all, as shown in Figure 26.23.

Figure 26.23: The standard feedforward control loop

While cascade control is used to counteract disturbances that affect the control effort, feedforward control can be used to counteract loads that do not affect the control effort. As an example, consider the heat exchanger in Figure 26.24.

Figure 26.24: A temperature control loop on a heat exchanger

As previously discussed, if the inlet steam temperature changes or if the mass flow rate of the steam changes (e.g., if the steam system pressure changes so do the system flows), the rate of heat energy provided by the steam to the heat exchanger changes. Since this is a disturbance on the control effort, a cascade control scheme is appropriate. However, a disturbance may occur on a variable that isn't related to the control effort (e.g., the inlet water temperature or flow changes, or the setpoint changes), and this disturbance may affect the process variable. Figure 26.25 illustrates this in block diagram form. In this case, a feedforward control scheme is needed.

Figure 26.25: The standard feedback control loop with disturbance that is not on the control effort

A purely feedforward automobile cruise control system is an open-loop control and would be interfaced with topographical maps, real-time weather monitoring stations, and road surface sensors to decide how much engine power is necessary at any given time to attain the desired speed[12]. Assuming all relevant load variables are accounted for, the cruise control would be able to maintain constant speed regardless of conditions, and without the need to even monitor the car's speed.

In practice, it is nearly impossible to accurately account for *all* loads influencing a process and to both anticipate and counter-act their combined effects, and so pure feedforward control systems are rare. Instead, the feedforward principle finds use as an augment to normal feedback control. To understand feedforward control better, its pure application will be considered before exploring how it may be combined with feedback control.

First, consider a liquid level control system on an open tank, where three different fluid ingredients (shown in the following P&ID simply as A, B, and C) are mixed to produce a final product. A level transmitter (LT) measures liquid level, while a level controller (LC) compares this level to a setpoint value, and outputs a signal calling for a certain amount of discharge flow. A cascaded (slave) flow controller (FC) senses outgoing flow via a flow transmitter (FT) and works to maintain whatever rate of flow is "asked" for by the level controller, as shown in Figure 26.26.

[12]The load variables mentioned that influence a car's speed are an incomplete list at best. Many other variables come into play, such as fuel quality, engine tuning, and tire pressure. In order for a purely feedforward (i.e., no feedback monitoring of the process variable) control system to work, *every single load variable* must be accurately monitored and factored into the system's output signal. This is impractical for most applications, which is why feedforward control is usually used in conjunction with feedback control.

Figure 26.26: A cascade control loop for controlling tank level

The level control system acts to keep liquid level constant in the vessel, ensuring adequate mixing of the three ingredients[13]. Being a feedback level control system, it adjusts the discharge flow rate in response to measured changes in liquid level. Like all feedback control systems, this one is *reactive* in nature: it can only take corrective action *after* a deviation between process variable (level) and setpoint is detected. As a result, temporary deviations from setpoint are guaranteed to occur with this control system every time the combined flow rate of the three ingredients increases or decreases.

Let us now change the control system strategy from feedback to feedforward. It is clear what the loads are in this process: the three ingredient flows entering the vessel. If these three flow rates[14] are

[13]If the liquid level drops too low, there will be insufficient retention time in the vessel for the fluids to mix before they exit the product line at the bottom. If liquid level is too high, the mixing action will be damped to the point where it ceases to be effective.

[14]The device or computer function performing the summation is shown in the P&ID as a bubble with "FY" as the label. The letter "F" denotes *Flow*, while the letter "Y" denotes a signal relay or transducer.

measured and summed, then the total incoming flow signal is used as a setpoint for the discharge flow controller, the outlet flow should (ideally) match the inlet flow, resulting in a constant liquid level. Being a purely feedforward control system, there is no level transmitter (LT) any more, just flow transmitters to measure the three loads, as shown in Figure 26.27.

Figure 26.27: A feedforward control loop for controlling tank level

If all flow transmitter calibrations are perfect, the summing of flow rates flawless, and the flow controller's tuning robust, this level control system should control liquid level in the vessel by pro-action rather than by reaction. Any change in the flow rate of ingredients A, B, and/or C will quickly result in a matching adjustment to the discharge flow rate. So long as total volumetric flow out of the vessel is held equal to total volumetric flow into the vessel, the liquid level inside the vessel *cannot* change[15].

[15]Incidentally, this is a good example of an *integrating* process, where the rate of process variable change over time

An interesting property of feedforward control systems is that they cannot generate oscillations as is the case with an over-tuned (excessive gain) feedback system. Since a feedforward system never monitors the effects of its actions, it will never react to something it did to the process, which is the foundation of feedback oscillation. While it is entirely possible for a feedforward control system to be configured with too much gain, the effect of this will be *overcompensation* for a load change rather than oscillation. In the case of the mixing tank feedforward level control process, improper instrument scaling and/or offsets will cause the discharge and inlet flows to mis-match, resulting in a liquid level that either continues to increase or decrease over time ("integrate"). However, no amount of poor adjustment can cause this feedforward system to produce oscillations.

In reality, this pure feedforward control system is impractical even if all calibrations and calculations within are perfect. There are still loads unaccounted for: evaporation of liquid from the vessel, for example, change in liquid density (often due to liquid temperature changes), or the occasional pipe fitting leak. Furthermore, since the control system has no "knowledge" of the actual liquid level, it cannot make adjustments to that level. If an operator, for instance, desired to decrease the liquid level to achieve a more vigorous mixing action, he or she would have to manually drain liquid out of the vessel, or temporarily place the discharge flow controller in "manual" mode and increase the flow there (then place back into "cascade" mode where it follows the remote setpoint signal again). The advantage of pro-active control and minimum deviation from setpoint over time comes at a fairly high price of practicality and convenience.

For these reasons, feedforward control is most often found in conjunction with feedback control. This would work in the liquid level control system by incorporating a level transmitter and level controller back into the system, the output of that level controller being summed with the feedforward flow signal (by the LY summing relay) before going to the cascaded setpoint input of the discharge flow controller. This is shown in Figure 26.28.

is proportional to the imbalance of flow rates in and out of the process. Stated another way, total accumulated (or lost) mass in a mass-balance system such as this is the time-integral of the difference between incoming and outgoing flow rates: $m = \int_0^T (Q_{in} - Q_{out})\, dt$.

Figure 26.28: A feedforward with trim control loop for controlling tank level

This hybrid control strategy is sometimes called *feedforward with trim*. In this context, "trim" refers to the level controller's (LC) output signal contributing to the discharge flow setpoint, helping to compensate for any unaccounted loads (evaporation, leaks) and provide for level setpoint changes. This "trim" signal should do very little of the control work in this system, the bulk of the liquid level stability coming from the feedforward signals provided by the incoming flow transmitters.

A very similar control strategy commonly used on large steam boilers for the precise control of steam drum water level goes by the name of *three-element feedwater control*. Figure 26.29 shows an example of this control strategy implemented with pneumatic (3-15 psig signal) instruments.

Figure 26.29: A three-element control loop for controlling boiler level

Such a control system is called "three-element" because it makes use of three process measurements:

1. Feedwater flow rate
2. Steam drum water level
3. Steam flow rate

Feedwater flow is controlled by a dedicated flow controller (FIC), receiving a remote setpoint signal from a summing relay (LY). The summer receives two inputs: a mass steam flow signal and the output signal (trim) from the level controller (LIC). The feedforward portion of this system (mass steam flow feeding forward to mass water flow) is intended to match the mass flow rates of water into the boiler with steam flow out of the boiler. If steam demand suddenly increases, this feedforward portion of the system immediately calls for a commensurate increase in water flow, since every molecule of steam must come from one molecule of water. The level controller and transmitter act as a feedback control loop, supplementing the feedforward signal to the cascaded water flow

controller to make up for ("trim") any shortcomings of the feedforward loop.

A three-element boiler feedwater control system is a good example of a feedforward strategy designed to ensure *mass balance*, defined as a state of equality between all incoming mass flow rates and all outgoing mass flow rates. The steam flow transmitter measures outgoing mass flow, its signal being used to adjust incoming water mass rate. Since mass cannot be created or destroyed (the Law of Mass Conservation), every unit of steam mass leaving the boiler must be accounted for as an equivalent unit of water mass entering the boiler. If the control system perfectly balances these mass flow rates, water level inside the boiler *cannot* change.

In processes where the process variable is affected by energy flow rates rather than mass, the balance maintained by a feedforward control system will be *energy balance* rather than mass balance. Like mass, energy cannot be created or destroyed, but must be accounted for. A feedforward control system monitoring all incoming energy flows into a process and adjusting the outgoing energy flow rate (or visa-versa) will ensure no energy is depleted from or accumulated within the process, thus ensuring the stability of the processes' internal energy state.

An example of an energy-balance feedforward control system is found applied to the control of the heat exchanger in Figure 26.30.

Figure 26.30: A feedforward with trim control loop for controlling oil temperature

The two transmitters on the incoming (cold oil) line measure oil temperature and oil flow rate, respectively. The quotient of these two variables represents the *energy demand* of the incoming oil

(i.e., how much energy will be required to elevate the oil flow's temperature to some higher value). This "energy demand" signal is summed with the temperature controller's output signal to set the steam valve position (adding energy to the process).

A block diagram of the regulating type of feedforward control method is shown in Figure 26.17.

Figure 26.31: A regulating feedforward control system block diagram that corrects for process disturbances

This diagram shows that the total control effort ($u$) is the sum of the control efforts from the feedback controller ($u1$) and from the feedforward controller ($u2$). This feedforward architecture is used to correct for process disturbances. A tracking type of feedforward structure is used to correct for setpoint disturbances, and is shown in Figure 26.32.

Figure 26.32: A tracking feedforward control system block diagram that corrects for changing setpoints

### 26.4.1 Feedforward with dynamic compensation

As discussed previously, feedforward control is a way to improve the stability of a feedback control system in the face of changing loads. Rather than rely on feedback to make corrective changes to a process only *after* some load change has driven the process variable away from setpoint, feedforward systems monitor the relevant load(s) and use that information to preemptively make stabilizing changes to the final control element such that the process variable will not be affected. In this way, the feedback loop's role is to merely "trim" the process for factors lying outside the realm of the feedforward system.

At least, this is how feedforward control is *supposed* to work. One way feedforward controls commonly fail to live up to their promise is if the effects of load changes and of manipulated variable changes possess different time lags in their respective effects on the process variable. This is a problem in feedforward control systems because it means the corrective action called for in response to a change in load will not affect the process variable at the same time, or in the same way over time, as the load will. In order to correct this problem, time lags must be intelligently inserted into the control system to equalize the time lags of load and feedforward correction. This is called *dynamic compensation*.

The following subsections will explore illustrative examples to make both the problem and the solution(s) clear.

### Dead time compensation

Examine the following control system P&ID showing the addition of *flocculant* (a chemical compound used in water treatment to help suspended solids clump together for easier removal by filtering and/or gravity clarification) and *lime* for pH balance. Flocculant is necessary to expedite the removal of impurities from the water, but some flocculation compounds have the unfortunate effect of decreasing the pH value of the water (turning it more acidic). If the water's pH value is too low, the flocculant ironically loses its ability to function. Thus, lime (an alkaline substance) must be added to the water to counter-act the flocculant's effect on pH to ensure efficient flocculation. Both substances are powders in this water pre-treatment system, metered by variable-speed screw conveyors and carried to the mixing tank by belt-style conveyors:

Figure 26.33: A control system for controlling pH in a water treatment process

The control system shown in this P&ID consists of a pH analyzer (AT) transmitting a signal to a pH indicating controller (AIC) which adjusts the speed of the lime screw conveyor. The flocculant screw conveyor speed is manually set by a *hand indicating controller* (HIC) – sometimes known as a *manual loading station* – adjusted when necessary by experienced water treatment operators who periodically monitor the effectiveness of flocculation in the system.

This simple feedback control system will work fine in steady-state conditions, but if the operator suddenly changes flocculant flow rate into the mixing vessel, there will be a temporary deviation of pH from setpoint before the pH controller is able to find the correct lime flow rate into the vessel to compensate for the change in flocculant flow. In other words, flocculant feed rate into the mixing

tank is a *load* for which the pH control loop must compensate.

Dynamic response could be greatly improved with the addition of feedforward control to this system as shown in Figure 26.34.

Figure 26.34: A feedforward with trim control system for controlling pH in a water treatment process

Here, the hand controller's signal goes to a signal summing relay (or function block in a digital control system) in addition to acting as a control signal to the flocculant screw conveyor motor drive. If an operator changes the flocculant feed rate, the lime feed rate will immediately adjust to compensate, *before* any change in pH value takes place in the water. Ideally, the pH controller need only make minor "trim" adjustments to lime feed rate, while the feedforward signal does most of the work in maintaining a steady pH value[16].

Even if all components in the feedforward system have been calibrated and configured properly, however, a potential problem still lurks in this system which can cause the pH value to temporarily deviate from setpoint following flocculant feed rate changes. This problem is the *transport delay* –

[16]Much of the complexity which would normally accompany a feedforward control scheme in a pH system has been deliberately omitted from this diagram, including gain and bias relays (function blocks), nonlinear functions, limits, and other signal-manipulation algorithms likely necessary to balance lime addition with flocculant addition. pH is a very nonlinear process to control, and as such a simple 1:1 ratio system like this would certainly yield poor results. However, the basic concept will still serve to illustrate the problem of differential time lags in a feedforward system, which still have not been directly revealed at this point!

otherwise known as *dead time* – inherent to the two belt conveyors transporting both flocculant and lime powder from their respective screw conveyors to the mixing vessel. If the rotational speed of a screw conveyor changes, the flow rate of powder exiting that screw conveyor will immediately and proportionately change. However, the belt conveyor imposes a time delay before the new powder feed rate makes it into the mixing vessel. In other words, the water in the vessel will not "see" the effects of a change in flocculant or lime feed rate until after this time delay has elapsed. This is not a problem if the dead times of both belt conveyors are exactly equal, since this means any compensatory change in lime feed rate initiated by the feedforward system will reach the water at exactly the same time the new flocculant rate reaches the water. So long as flocculant and lime feed rates are precisely balanced with one another at the point in time they reach the mixing vessel, pH should remain stable. But what if their arrival times are not coordinated – what will happen to pH then?

Here is a "thought experiment" to explore the consequences of the flocculant conveyor belt moving much slower than, and/or being much longer than, the lime conveyor belt. Suppose the flocculant belt imposed a dead time of 60 seconds on flocculant powder making it to the vessel, while the lime belt only delayed lime powder transit by 5 seconds from screw conveyor to mixing tank. This would mean changes in flocculant flow (set by the hand controller) would compensate with changes in lime flow *55 seconds too soon*. Now imagine the human operator making a sudden increase to the flocculant powder feed rate. The lime feed rate would immediately increase due to the efforts of the feedforward system. However, since the increased flow rate of lime powder will reach the mixing vessel 55 seconds before the increased flow rate of flocculant powder, the effect will be a temporary increase in pH value beginning about 5 seconds after the operator's change, and then a settling of pH value back to setpoint[17], as shown in the timing diagram in Figure 26.35.

Figure 26.35: A timing diagram for a system with a long dead time for flocculant feed

The obvious solution to this problem is to mechanically alter the belt conveyor systems for

[17]This "thought experiment" assumes no compensating action on the part of the feedback pH controller for the sake of simplicity. However, even the pH controller's efforts are included, the problem does not go away. As pH rises due to the premature addition of extra lime, the controller will try to reduce the lime feed rate. This will initially reduce the degree to which pH deviates from setpoint, but then the reverse problem will occur when the increased flocculant enters the vessel 55 seconds later. Now, the pH will drop below setpoint, and the feedback controller will have to ramp up lime addition (to the amount it was before the additional lime reached the vessel) to achieve setpoint.

equal transport times of flocculant and lime powders. If this is impractical, a similar result may be achieved by incorporating another signal relay (or digital function block) inserting dead time into the feedforward control system. In other words, the control system can be modified in such a way to emulate what would be impractical to modify in the process itself.

This new function will add a dead time of 55 seconds to the feedforward signal before it enters the summer, thus delaying the lime feed rate's response to feedforward effect by just the right amount of time such that any lime feed rate changes called for by feedforward action will arrive at the vessel *simultaneously* with the changed flocculant feed rate as shown in Figure 26.36.

Figure 26.36: A P&ID showing lime feed dead time compensation in a pH control process

This solution to different time lags in a physical system, of purposely adding time lag functions to the signal path(s) of a control system, is called *dynamic compensation*. The proper selection and placement of such dynamic compensation elements in a control system is best done after all the time delays of the process are understood, which is why this scenario began with a detailed explanation of delay time between the two conveyor belts. Only after understanding time lags intrinsic to the process, and understanding the effect these differing time lags will have on the efficacy of feedforward control, can a dynamic compensation element be intelligently placed in a control system for beneficial effect.

Note how the feedback pH controller's loop was purposely spared the effects of the added dead time function, by placing the function outside of that controller's feedback loop. This is important,

as dead time in any form is the bane of feedback control. The more dead time within a feedback loop, the easier that loop will tend to oscillate. By strategically placing the dead time function before the summing relay rather than after (between the summer and the lime screw conveyor motor drive), the feedback control system still achieves minimum response time while only the feedforward signal gets delayed.

Now consider the same flocculant and lime powder control system, this time with transport delays reversed between the two belt conveyors. If the flocculant conveyor belt is now the fast one (5 seconds dead time) and the lime belt slow (60 seconds), the effects of flocculant feed rate changes will be reversed. An increase in flocculant powder feed rate to the vessel will result in a drop in pH beginning 5 seconds after the HIC setting change, followed by a rise in pH value after the additional lime feed rate finally reaches the vessel. This is shown in Figure 26.37.

Figure 26.37: A timing diagram for a system with a long dead time for lime feed

It would be possible to compensate for the difference in conveyor belt transport times using a special relay in the same location of the feedforward signal path as before, if only there was such a thing as a relay that could *predict* the future exactly 55 seconds in advance! Since no such device exists (or ever will exist), dynamic compensation must be applied elsewhere in the feedforward control system.

If a time delay is the only type of compensation function at our disposal, then the only thing that can be delayed in this system to make the two dead times equal is the flocculation feed rate. Thus, a 55-second dead time relay should be placed in the signal path between the hand indicating controller (HIC) and the flocculant screw conveyor motor drive.

Figure 26.38 shows the proper placement of the dead time function.

Figure 26.38: A P&ID showing flocculant feed dead time compensation in a pH control process

With this dead time relay in place, any change in flocculation feed rate initiated by a human operator will immediately adjust the feed rate of lime powder, but delay an adjustment to flocculant powder feed rate by 55 seconds, so the two powders' feed rate changes arrive at the mixing vessel simultaneously.

### Lag time compensation

Process time delays characterized by pure transport delay (dead time) are less common in industry than other forms of time delays, most notably *lag times* A simple "lag" time is the characteristic exhibited by a low-pass RC filter circuit, where a step-change in input voltage results in an output voltage asymptotically rising to the new voltage value over time as shown in Figure 26.39.

Figure 26.39: An RC circuit that exemplifies a first-order lag function

The *time constant* ($\tau$) of such a system – be it an RC circuit or some other physical process – is the time required for the output to move 63.2% of the way to its final value ($1 - e^{-1}$). For an RC circuit such as the one shown, $\tau = RC$ (assuming $R_{load} >> R$ so the load resistance will have negligible effect on timing).

Lag times differ fundamentally from dead times. With a dead time, the effect is simply time-delayed by a finite amount from the cause, like an echo. With a lag time, the effect begins at the exact same time as the cause, but does not follow the same rapid change over time as the cause. Like dead times in a feedforward system, it is quite possible (and in fact usually the case) for loads and final control variables to have differing lag times regarding their respective effects on the process variable. This presents another form of the same problem seen in the two-conveyor water pretreatment system, where an attempt at feedforward control was not completely successful because the corrective feedforward action does not happen with the same amount of *lag* as the load.

One example is a heat exchanger used to preheat fuel oil before being sent to a combustion furnace. Hot steam is the heating fluid used to pre-heat the oil in the heat exchanger. As steam gives up its thermal energy to the oil through the walls of the heat exchanger tubes, it undergoes a phase change to liquid form (water), where it exits the shell of the exchanger as "condensate" ready to be re-boiled back into steam.

A simple feedback control system regulates steam flow to the heat exchanger, maintaining the discharge temperature of the oil at a constant setpoint value, as shown in Figure 26.40.

Figure 26.40: A heat exchanger serving as an example of a first-order lag function

Once again, it should come as no surprise that the outlet temperature will suffer temporary deviations from setpoint if load conditions happen to change. The feedback control system may be able to *eventually* bring the exiting oil's temperature back to setpoint, but it cannot begin corrective action until *after* a load has driven the oil temperature off setpoint. What is needed for improved control is *feedforward* action in addition to feedback action. This way, the control system can take corrective action in response to load changes *before* the process variable gets affected.

Suppose the dominant load in this system is known to be the oil flow rate[18], caused by changes in demand at the combustion furnace where this oil is being used as fuel. Adapting this control system to include feedforward is as simple as installing an oil flow transmitter and a summing relay (or summing function block) as shown in Figure 26.41.

[18]Knowing this allows the measurement of the incoming cold oil temperature to be avoided and just measure incoming cold oil flow rate as the feedforward variable. If the incoming oil's temperature were known to vary substantially over time, it would need to be measured as well as flow rate, combining the two variables together to calculate the *energy demand* and using this abstract variable as the feedforward variable.

Figure 26.41: A feedforward with trim compensation for a first-order lag function

With feedforward control action in place, the steam flow rate will immediately change with oil flow rate, preemptively compensating for the increased or decreased heat demand of the oil. In other words, the feedforward system acts to maintain precise *energy balance* in the process, so heat energy never accumulates in the exchanger or bleeds away from the exchanger, causing changes in temperature.

There is a problem of time delay in this system, however: a change in oil flow rate has a *faster* effect on outlet temperature than a proportional change in steam flow rate. This is due to the relative masses impacting the temperature of each fluid. The oil's temperature is primarily coupled to the temperature of the tubes, whereas the steam's temperature is coupled to both the tubes and the shell of the heat exchanger. So, the steam has a greater mass to heat than the oil has to cool, giving the steam a larger thermal time constant than the oil.

For the sake of illustration, assume transport delays are short enough to ignore[19], so there is a difference in *lag* times between the oil flow's effect on temperature and the steam flow's effect on temperature.

Figure 26.42 is what would happen to the heated oil temperature if steam flow were held constant and oil flow were suddenly increased.

[19]Transport delay (dead time) in heat exchanger systems can be a thorny problem to overcome, as they they tend to change with flow rate! For reasons of simplicity in our illustration, this process will be treated as if it only possessed lag times, not dead times.

Figure 26.42: A step change in oil flow causes the temperature transient shown

Increased oil flow convects heat away from the steam at a faster rate than before, resulting in decreased oil temperature. This drop in temperature is fairly quick, and is self-regulating.

For contrast, Figure 26.43 illustrates what would happen to the heated oil temperature if oil flow were held constant and steam flow were suddenly increased.

Figure 26.43: A step change in steam flow causes the temperature transient shown

Increased steam flow convects heat into the oil at a faster rate than before, resulting in increased oil temperature. This rise in temperature is also self-regulating, but much slower than the temperature change resulting from a proportional adjustment in oil flow. In other words, the time constant ($\tau$) of the process with regard to steam flow changes is greater than the time constant of the process with regard to oil flow changes ($\tau_{steam} > \tau_{oil}$).

If these two effectsare superimposed, as will be the case when the feedforward system is working (without the benefit of feedback "trim" control), when the oil flow suddenly increases there will be a "fight" between the cooling effect of the increased oil flow and the heating effect of the increased steam flow. However, it will not be a fair fight initially: the oil flow's effect will temporarily win over the steam's effect because of the oil's faster time constant. The result will be a momentary dip in outlet temperature before the system achieves equilibrium again as shown in Figure 26.44.

Figure 26.44: A step change in steam flow causes the temperature transient shown

The solution to this problem is not unlike the solution applied to the water treatment system: these two time delays must be made more equal so their superimposed effects will directly cancel, resulting in an undisturbed process variable. An approximate solution for equalizing two different lag times is to cascade two lags together in order to emulate one larger lag time[20]. This may be done by inserting a lag time relay or function block in the feedforward system.

When looking at the P&ID, though, a problem is immediately evident. The lag time that needs to slow down is the lag time of the oil flow's effect on temperature. In this system, oil flow is a wild variable, not something that is controllable. The feedforward control system can only manipulate the steam valve position in response to oil flow, not influence oil flow in order to give the steam time to "catch up."

If the time constant inherent to the wild variable (oil flow) cannot be slowed down, then the best thing to do is speed up the time constant of the variable that can be influenced (steam flow). The solution is to insert something called a *lead function* into the feedforward signal driving the steam valve. A "lead" is the mathematical inverse of a lag. If a lag is modeled by an RC low-pass filter circuit, then a "lead" is modeled by an RC high-pass filter circuit as shown in Figure 26.45.

Figure 26.45: An RC circuit that exemplifies a first-order lead function

Being mathematical inverses of each other, a lead function should perfectly cancel a lag function when the output of one is fed to the input of the other, and when the time constants of each are equal. If the time constants of lead and lag are not equal, their cascaded effect will be a partial cancellation. In the heat exchanger control application, this is what must be done: partially cancel the steam valve's slow time constant so it will be more equal with the oil flow's time constant. Therefore, a lead function must be inserted into the feedforward signal path.

A lead function will take the form of either a physical signal relay or (more likely with modern technology) a function block executed inside a digital control system. The proper place for the lead function is between the oil flow transmitter and the summation function, as shown in Figure 26.46.

[20]Technically, two cascaded lag times is not the same as one large lag time, not matter what the time constant values. Two first-order lags in series with one another create a *second-order lag*, which is a different effect. However imperfect as the added lag solution is, it is still better than nothing at all!

Figure 26.46: Proper application of a lead function for feedforward control

Now, when the oil flow rate to this heat exchanger suddenly changes, the lead function will add a "surge" to the feedforward signal before it goes to the summing function, quickly opening the steam valve further than usual and sending a surge of steam to the exchanger to help overcome the naturally sluggish response of the oil temperature to changes in steam flow. The dynamic compensation will not be perfect, but it will be substantially better than no dynamic compensation at all.

### Lead/Lag and dead time function blocks

The addition of dynamic compensation in a feedforward control system may require a lag function, a lead function, and/or a dead time function, depending on the nature of the time delay differences between the relevant process load and the system's corrective action. Modern control systems provide all these functions as digital *function blocks*. In the past, these functions could only be implemented in the form of individual instruments with these time characteristics, called *relays*. As already seen, lead and lag functions may be rather easily implemented as simple RC filter circuits. Pneumatic equivalents also exist, which were the only practical solution in the days of pneumatic transmitters and controllers. Dead time is notoriously difficult to emulate using analog components of any kind, and so it was common to use lag-time elements (sometimes more than one connected in series) to provide an approximation of dead time.

With digital computer technology, all these dynamic compensation functions are easy to implement and readily available in a control system. Some single-loop controllers even have these capabilities programmed within, ready to use when needed.

A dead time function block is most easily implemented using the concept of a *first-in, first-out shift register*, sometimes called a *FIFO*. With this concept, successive values of the input variable are stored in a series of registers (memory), their progression to the output delayed by a certain amount of time. An example is shown in Figure 26.47.

Figure 26.47: A first-in, first-out (FIFO) shift register used to create dead time

Lead and lag functions are also implemented digitally in modern controllers and control systems, but they are actually easier to comprehend in their analog (RC circuit) forms. The most common way lead and lag functions are found in modern control systems is in combination as the so-called *lead/lag function*, merging both lead and lag characteristics in a single function block (or relay). An example is shown in Figure 26.48.

Figure 26.48: An RC circuit that exemplifies a first-order lead/lag function

Each parallel RC subcircuit represents a time constant ($\tau$), one for lead and one for lag. The overall behavior of the network is determined by the relative magnitudes of these two time constants. The network with the larger time constant determines the overall characteristic of the network.

If the two time constant values are equal to each other ($\tau_{lead} = \tau_{lag}$), then the circuit performs no dynamic compensation at all, simply passing the input signal to the output with no change except for some attenuation. A timing diagram that illustrates this is shown in Figure 26.49.

Figure 26.49: A first-order lead/lag function with equal time constants

A square wave signal entering this network will exit the network as a square wave. If the input signal is sinusoidal, the output will also be sinusoidal and in-phase with the input.

If the lag time constant exceeds the lead time constant ($\tau_{lag} > \tau_{lead}$), then the overall behavior of the circuit will be to introduce a first-order lag to the voltage signal. A timing diagram that illustrates this is shown in Figure 26.50.

Figure 26.50: A first-order lead/lag function with dominant lag

A square wave signal entering the network will exit the network as a sawtooth-shaped wave. A sinusoidal input will emerge sinusoidal, but with a lagging phase shift. This, in fact, is where the *lag* function gets its name: from the negative phase shift it imparts to a sinusoidal input.

Conversely, if the lead time constant exceeds the lag time constant ($\tau_{lead} > \tau_{lag}$), then the overall behavior of the circuit will be to introduce a first-order lead to the voltage signal (a step-change voltage input will cause the output to "spike" and then settle to a steady-state value). A timing diagram that illustrates this is shown in Figure 26.51.

Figure 26.51: A first-order lead/lag function with dominant lead

A square wave signal entering the network will exit the network with sharp transients on each leading edge. A sinusoidal input will emerge sinusoidal, but with a leading phase shift. Not surprisingly, this is where the *lead* function gets its name: from the positive phase shift it imparts to a sinusoidal input.

One of the design challenges for analog lead/lag networks is how to built them in such a way that they would not attenuate in the steady-state condition, since any reduction in signal strength interferes with the proper proportioning of the feedforward signal, effectively altering the gain of the feedforward action and introducing an offset to the signal if the signal had a live-zero base such as 3-15 psig or 1-5 volts. This is not a problem in digital lead/lag algorithms, where the lead/lag function is implemented using equations evaluated by a microprocessor.

For example, a typical time-domain equation describing a digital lead/lag function block's output

response ($y$) to an input step-change from zero (0) to magnitude $x$ over time ($t$) is as follows:

$$y = x\left(1 + \frac{\tau_{lead} - \tau_{lag}}{\tau_{lag}\, e^{\frac{t}{\tau_{lag}}}}\right)$$

As can be seen, if the two time constants are set equal to each other ($\tau_{lead} = \tau_{lag}$), the second term inside the parentheses will have a value of zero at all times, simplifying the equation and making $y$ equal to $x$ at all times. If the lead time constant exceeds the lag time constant ($\tau_{lead} > \tau_{lag}$), then the fraction will begin with a positive value and decay to zero over time, giving the "spike" response expected from a lead function. Conversely, if the lag time constant exceeds the lead ($\tau_{lag} > \tau_{lead}$), the fraction begin with a negative value at time = 0 (the beginning of the step-change) and decay to zero over time, giving the "sawtooth" response expected from a lag function.

From both an examination of the analog lead/lag networks and from this equation, it is easy to tell that the proper configuration of a lead/lag function requires *two* time constant values be set. The rate of decay for the lead/lag function (i.e., how quickly it settles to a steady-state condition after a step-change input) is primarily determined by the lag time constant ($\tau_{lag}$), while the gain of the function (i.e., how severely the output will either "spike" or "retard" following a step-change input) is determined by the difference between the two time constants ($\frac{\tau_{lead}-\tau_{lag}}{\tau_{lag}}$). To summarize with several examples:

- If $\tau_{lead} = \tau_{lag}$, it will simply pass the input signal through to the output (no lead or lag action at all)
- If $\tau_{lead} = 0$, it will provide a lag function with a gain of unity and a time constant of $\tau_{lag}$
- If $\tau_{lead} = 2(\tau_{lag})$, it will provide a lead function with a gain of unity and a time constant of $\tau_{lag}$

Figure 26.52: A first-order lead/lag function response with varying dominant terms

## 26.5 Ratio control

Most people reading this book have likely had the experience of adjusting water temperature as they took a shower with dual valves: one valve controlling hot water and one valve controlling cold water. In order to adjust water temperature, the *proportion* of one valve opening to the other must be changed. Increasing or decreasing total water flow rate without upsetting the outlet temperature is a matter of adjusting both valves in the same direction, maintaining that same proportion of hot to cold water flow.

This is a control strategy known as *ratio control*. In ratio control, the ratio of one flow rate to another is controlled for some desired outcome. Many industrial processes also require the precise mixing of two or more ingredients to produce a desired product. Not only do these ingredients need to be mixed in proper proportion, but it is usually desirable to have the total flow rate subject

to arbitrary increases and decreases so production rate as a whole may be altered at will. Most examples of ratio control are actually special types of feedforward control, however, it is usually discussed as its own type of control scheme.

A simple example of ratio control is in the production of paint, where a base liquid must be mixed with one or more pigments to achieve a desired consistency and color. A manually controlled paint mixing process, similar to the hot and cold water valve "process" in some home showers, is shown in Figure 26.53. Two flowmeters, a ratio calculating relay, and a display provide the human operator with a live measurement of pigment-to-base ratio.

Figure 26.53: An operator-controlled mixing process

One alteration that could be made to this mixing system is to link the two manual control valve handles together in such a way that the ratio of base to pigment was *mechanically* established. All the human operator needs to do now is move the one link to increase or decrease mixed paint production, as seen in Figure 26.54.

Figure 26.54: An operator-controlled mixing process with a linked control

Adjusting the pigment-to-base ratio is now a matter of adjusting the linkage ratio, a task most likely performed by a mechanic or someone else knowledgeable in the operation of mechanical linkages. The convenience of total flow adjustment gained by the link comes at the price of inconvenient ratio adjustment.

Mechanical link ratio-control systems are commonly used to manage simple burners, proportioning the flow rates of fuel and air for clean, efficient combustion. A photograph of such a system appears in Figure 26.55, showing how the fuel gas valve and air damper motions are coordinated by a single rotary actuator.

Figure 26.55: An operator-controlled mixing process with a linked control

As you can see in Figure 26.55, the fuel gas valve is actuated by means of a cam, allowing precise "tuning" of the valve characteristics for consistent fuel/air ratio across a wide range of firing rates. Making ratio adjustments in such a linkage system is obviously a task for a skilled mechanic or technician.

A more automated approach to the general problem of ratio control involves the installation of a flow control loop on one of the lines, while keeping just a flow transmitter on the other line. The

signal coming from the uncontrolled flow transmitter becomes the setpoint for the flow control loop as shown in Figure 26.56.

Figure 26.56: An automated mixing process with simple ratio control

Here, the flow transmitter on the uncontrolled line measures the flow rate of base, sending a flow rate signal to the pigment flow controller which acts to match flow rates. If the calibrations of each flow transmitter are precisely equal to one another, the ratio of pigment to base will be 1:1 (equal). The flow of base liquid into the mixing system is called a *wild flow* or *wild variable*, since this flow rate is not controlled by the ratio control system. The only purpose served by the ratio control system is to match the pigment flow rate to the wild (base) flow rate, so the same ratio of pigment to base will always be maintained regardless of total flow rate. Thus, the flow rate of pigment will be held *captive* to match the "wild" base flow rate, which is why the controlled variable in a ratio system is sometimes called the *captive variable* (in this case, a captive *flow*)[21].

As with the mechanically-linked manual ratio mixing system, this ratio control system provides convenient total flow control, but inconvenient control over mixing ratio. In order to alter the ratio of pigment to base, someone would have to re-calibrate one (or both) flow transmitters. To achieve a 2:1 ratio of base to pigment, for example, the base flow transmitter's range would have to be double that of the pigment flow transmitter. This way, an equal percentage of flow registered by both flow transmitters (as the ratio controller strives to maintain equal percentage values of flow between pigment and base) would actually result in twice the amount of base flow than pigment

[21] This is an example of how a ratio controller is really a type of feedforward controller. Disturbances on the wild flow are fed forward via the ratio controller which automatically adjusts the captive flow

flow.

A convenient ratio adjustment may be incorporated into this system by adding another component (or function block) to the control scheme: a device called a *signal multiplying relay* (or alternatively, a *ratio station*). This device (or computer function) takes the flow signal from the base (wild) flow transmitter and multiplies it by some constant value known as the *ratio* ($k$) before sending the signal to the pigment (captive) flow controller as a setpoint. This is shown in Figure 26.57.

Figure 26.57: An automated mixing process with a ratio station

With identical flow range calibrations in both flow transmitters, this multiplying constant $k$ directly determines the base-to-pigment ratio (i.e., the ratio will be 1:1 when $k = 1$; the ratio will be 2:1 when $k = 2$, etc.). If the $k$ value is easily adjusted by a human operator, the mixing ratio becomes a very simple parameter to change at will, just as the total production rate is easy to adjust by moving the base flow control valve.

Another example of ratio control at work is in a process whereby hydrocarbon gases (usually methane) are converted into hydrogen gas and carbon dioxide gas. This is known as the *steam-hydrocarbon reforming process* or *cracking*, and it is one of the more popular ways of generating hydrogen gas for industrial use, such as for fuel cells. The overall reaction for this process with methane gas ($CH_4$) and steam ($H_2O$) as the reactants is as follows[22]:

[22]The conversion from hydrocarbon and steam to hydrogen and carbon dioxide is typically a two-stage process: the first (*reforming*) stage produces hydrogen gas and carbon monoxide, while a second (*water-gas-shift*) stage adds more steam to convert the carbon monoxide into carbon dioxide with more hydrogen liberated. Both reactions are endothermic, with the reforming reaction being more endothermic than the water-gas-shift reaction.

$$CH_4 + 2H_2O \rightarrow 4H_2 + CO_2$$

This is an endothermic reaction, which means a net input of energy is required to make it happen. Typically, the hydrocarbon gas and steam are mixed together in a heated environment in the presence of a catalyst (to reduce the activation energy requirements of the reaction). This usually takes the form of catalyst-packed metal tubes inside a gas-fired furnace. It is important to control the proportion of gas to steam flow into this process. Too much hydrocarbon gas, and the result will be "coking" (solid hydrocarbon deposits) inside the heated tubes and on the surface of the catalyst beads, decreasing the efficiency of the process over time. Too much steam and the result is wasted energy as unreacted steam simply passes through the heater tubes, absorbing heat and carrying it away from the catalyst where it would otherwise do useful work.

One way to achieve the proper ratio of hydrocarbon gas to steam flow is to install a normal flow control loop on one of these two reactant feed lines, then use that process variable (flow) signal as a setpoint to a flow controller installed on the other reactant feed line. This way, the second controller will maintain a proper balance of flow to proportionately match the flow rate of the other reactant. An example P&ID is shown in Figure 26.58, where the methane gas flow rate establishes the setpoint for steam flow control.

Figure 26.58: An automated ratio control system for a reformer

Note how the methane gas flow transmitter signal goes both to the methane flow controller and to a *multiplying relay* that multiplies this signal by a constant value ($k$) before passing it on to the steam flow controller as a setpoint. This $k$ value sets the *ratio* of steam flow to methane flow. Although this might appear to be a cascade control system at first glance, it is actually quite different. In a cascade system, the *output* of one controller becomes the setpoint for another. Here in a ratio control system, the *process variable* of one controller becomes the setpoint for another, such that two process variables remain in constant proportion (ratio) to one another.

If the two flow transmitters are compensated to measure mass flow, the ideal value of $k$ should be set such that two molecules of steam vapor ($H_2O$) enter the reforming furnace for every one molecule of methane ($CH_4$). With a 2-to-1 molecular ratio of steam to methane (2 moles of steam per one mole of methane), this equates to a 9-to-4 mass flow ratio once the formula weights of steam and methane are calculated[23]. Thus, if the methane and gas flowmeters are calibrated for equal mass

[23]Steam has a formula weight of 18 amu per molecule, with two hydrogen atoms (1 amu each) and one oxygen atom (16 amu). Methane has a formula weight of 16 amu per molecule, with one carbon atom (12 amu) and four hydrogen atoms (1 amu each). If a molecular ratio of 2:1 steam-to-methane is desired, this makes a formula weight ratio of

flow ranges, the ideal value for $k$ should be $\frac{9}{4}$, or 2.25. Alternatively, the flow transmitter calibrations could be set in such a way that the ideal ratio is intrinsic to those transmitters' ranges (i.e., the methane flow transmitter has 2.25 times the mass flow range of the steam flow transmitter), with $k$ set to an ideal value of 1. This way a 9:4 ratio of methane mass flow to steam mass flow will result in equal percentage output values from both flow transmitters. In practice, the value for $k$ is set a bit higher than ideal, in order to ensure just a little excess steam to guard against coking inside the reaction heater tubes[24].

Another layer of sophistication to this ratio control system could be added by installing a gas analyzer at the outlet of the reaction furnace designed to measure the composition of the product stream. This analyzer's signal could be used to adjust the value of $k$ so the ratio of steam to methane would automatically vary to ensure optimum production quality even if the feedstock composition (i.e., percentage concentration of methane in the hydrocarbon gas input) changes. This is shown in Figure 26.59.

36:16, or 9:4.

[24]It is quite common for industrial control systems to operate at ratios a little bit "skewed" from what is stoichiometrically ideal due to imperfect reaction efficiencies. Given the fact that no chemical reaction ever goes to 100% completion – simply because 100% mixing is virtually impossible – a decision must be made as to which form of incompleteness is worse. In a steam-hydrocarbon reforming system, which is worse: excess (unreacted) steam at the outlet, or excess (unreacted) hydrocarbon at the outlet? Excess hydrocarbon content will "coke" the catalyst and heater tubes, which is very bad for the process over time. Excess steam merely results in a bit more operating energy loss, with no degradation to equipment life. The choice, then, is clear: it is better to operate this process "lean" (more steam than ideal) than "rich" (less steam than ideal).

Figure 26.59: A control system for a reformer using analytical feedback for automatic ratio adjustment

## 26.6 Limit, Selector, and Override controls

Another category of control strategies involves the use of signal relays or function blocks with the ability to switch between different signal values, or redirect signals to new pathways. Such functions are useful when a control system must choose between multiple signals of differing value in order to make the best control decisions.

The "building blocks" of such control strategies are special relays (or function blocks in a digital control system) shown in Figure 26.60.

Figure 26.60: Limit and selector control functions on a P&ID

*High-select* functions output whichever input signal has the *greatest* value. *Low-select* functions do just the opposite: output whichever input signal has the *least* value. "Greater-than" and "Less than" symbols mark these two selector functions, respectively, and each type may be equipped to receive more than two input signals.

Sometimes you will see these relays represented in P&IDs simply by an inequality sign in the middle of the large bubble, rather than off to the side in a square. Bear in mind that the location of the input lines has no relationship at all to the direction of the inequality symbol – e.g., it is not as though a high-select relay looks for the input on the left side to be greater than the input on the right. Note the examples shown in Figure 26.61, complete with sample signal values.

Figure 26.61: Selector control functions with sample signals

*High-limit* and *low-limit* functions are similar to high- and low-select functions, but they only receive one input each, and the limit value is a parameter programmed into the function rather than received from another source. The purpose of these functions is to place a set limit on how high or how low a signal value is allowed to go before being passed on to another portion of the control system. If the signal value lies within the limit imposed by the function, the input signal value is simply passed on to the output with no modification.

Like the select functions, limit functions may appear in diagrams with nothing more than the limit symbol inside the bubble, rather than being drawn in a box off to the side, as shown in Figure 26.62.

Figure 26.62: Limit control functions with sample signals

*Rate limit* functions place a maximum rate-of-change limit on the input signal, such that the output signal will follow the input signal precisely until and unless the input signal's rate-of-change over time ($\frac{dx}{dt}$) exceeds the pre-configured limit value. In that case, the relay still produces a ramping output value, but the rate of that ramp remains fixed at the limit $\frac{dx}{dt}$ value no matter how fast the input keeps changing. After the output value "catches up" with the input value, the function once again will output a value precisely matching the input unless the input begins to rise or fall at too fast a rate again.

### 26.6.1 Limit controls

A common application for select and limit functions is in *cascade* control strategies, where the output of one controller becomes the setpoint for another. It is entirely possible for the primary (master) controller to call for a setpoint that is unreasonable or unsafe for the secondary (slave) to attain. If this possibility exists, it is wise to place a limit function between the two controllers to limit the cascaded setpoint signal.

In the example in Figure 26.63, a cascade control system regulates the temperature of molten metal in a furnace, the output of the master (metal temperature) controller becoming the setpoint of the slave (air temperature) controller. A high limit function limits the maximum value this cascaded setpoint can attain, thereby protecting the refractory brick of the furnace from being exposed to excessive air temperatures.

Figure 26.63: A limit control function used in a cascade control system

It should be noted that although the different functions are drawn as separate bubbles in the P&ID, it is possible for multiple functions to exist within one physical control device. In this example, it is possible to find a controller able to perform the functions of both PID control blocks (master and slave) and the high limit function as well. It is also possible to use a distributed technology such as FOUNDATION Fieldbus to place all control functions inside field instruments, so only three field instruments exist in the loop: the air temperature transmitter, the metal temperature transmitter, and the control valve (with a Fieldbus positioner).

This same control strategy could have been implemented using a low select function block rather than a high limit, as shown in Figure 26.64.

Figure 26.64: A select control function used in a cascade control system

Here, the low-select function selects whichever signal value is lesser: the setpoint value sent by the master temperature controller, or the maximum air temperature limit value sent by the hand indicating controller (HIC – sometimes referred to as a *manual loading station*).

An advantage of this latter approach over the former might be ease of limit value changes. With a pre-configured limit value residing in a high-limit function, it might be that only qualified maintenance people have access to changing that value. If the decision of the operations department is to have the air temperature limit value easily adjusted by anyone, the latter control strategy's use of a manual loading station would be better suited[25].

Another detail to note in this system is the possibility of *integral windup* in the master controller in the event that the high setpoint limit takes effect. Once the high-limit (or low-select) function secures the slave controller's remote setpoint at a fixed value, the master controller's output is no longer controlling anything: it has become decoupled from the process. If the metal temperature is still below setpoint when this occurs, the master controller's integral action will "wind up" the

[25]It is reasonable to keep such limit values inaccessible to low-level operations personnel. This is especially true in cases such as this where the presence of a high temperature setpoint limit is intended for the longevity of the equipment. There is a strong tendency in manufacturing environments for equipment operators to "push the limits" of production beyond values considered safe or expedient by the engineers who designed the equipment. Limits are there for a reason, and should not be altered except by people with full understanding of and full responsibility for the consequences!

output value over time with absolutely no effect, since the slave controller is no longer following its output signal. If and when the metal temperature reaches setpoint, the master controller's output will likely be saturated at 100% due to the time it spent winding up. This will cause the metal temperature to overshoot setpoint, as a positive error will be required for the master controller's integral action to wind back down from saturation.

A relatively easy solution to this problem is to configure the master controller to stop integral action when the high limit relay engages. This is easiest to do if the master PID and high limit functions both reside in the same physical controller. Many digital limit function blocks generate a bit representing the state of that block (whether it is passing the input signal to the output or limiting the signal at the pre-configured value), and some PID function blocks have a boolean input used to disable integral action. If this is the case with the function blocks comprising the high-limit control strategy, it may be implemented as shown in Figure 26.65.

Figure 26.65: A method of integral windup prevention using a limit function

Another method used to prevent integral windup is to make use of the *feedback* input available on some PID function blocks. This is an input used to calculate the integral term of the PID equation. In the days of pneumatic PID controllers, this option used to be called *external reset*. Normally connected to the output of the PID block, if connected to the output of the high-limit function it will let the controller know whether or not any attempt to wind up the output is having an effect. If the output has been de-selected by the high-limit block, integral windup will cease, as shown in Figure 26.66.

Figure 26.66: A second method of integral windup prevention using a limit function

Limit control strategies implemented in FOUNDATION Fieldbus instruments use the same principle, except that the concept of a "feedback" signal sending information backwards up the function block chain is an aggressively-applied design philosophy throughout the FOUNDATION Fieldbus standard. Nearly every function block in the Fieldbus suite provides a "back calculation" output, and nearly every function block accepts a "back calculation" input from a downstream block. The "Control Selector" (CS) function block specified in the FOUNDATION Fieldbus standard provides the limiting function needed between the master and slave controllers. The BKCAL_OUT signal of this selector block connects to the master controller's BKCAL_IN input, making the master controller aware of its selection status. If ever the Control Selector function block de-selects the master controller's output, the controller will immediately know to halt integral action. This is shown in Figure 26.67.

Figure 26.67: Antireset windup using a limit function in a FOUNDATION Fieldbus system

### 26.6.2 Selector controls

In the broadest sense, a "selector" control strategy is one where one signal gets selected from multiple signals in a system to perform a measurement control function. In the context of this book and this chapter, I will use the term "selector" to categorize the automatic selection of a measurement or setpoint signal. Selection of a controller output signal will be explored in the next subsection.

Perhaps one of the simplest examples of a selector control strategy is where a process variable signal must be selected from multiple transmitters. For example, consider the chemical reactor shown in Figure 26.68, where the control system must throttle the flow of coolant to keep the *hottest* measured temperature at setpoint, since the reaction happens to be exothermic (heat-releasing)[26].

[26]Only the coolant flow control instruments and piping are shown in this diagram, for simplicity. In a real P&ID, there would be many more pipes, valves, and other apparatus shown surrounding this process vessel.

Figure 26.68: Use of a select function to control the hottest temperature in a reactor

The high-select relay (TY-24) sends only the highest temperature signal from the three transmitters to the controller. The other two temperature transmitter signals are simply ignored.

Another use of selector relays (or function blocks) is for the determination of a *median* process measurement. This sort of strategy is often used on triple-redundant measurement systems, where three transmitters are installed to measure the exact same process variable, providing a valid measurement even in the event of transmitter failure.

The median select function may be implemented one of two ways using high- and low-select function blocks as shown in Figure 26.69.

Figure 26.69: Use of a select function to find the median value of a set of signals

The left-hand selector strategy selects the highest value from each pair of signals (A and B, B and C, A and C), then selects the lowest value of those three primary selections. The right-hand strategy is exactly opposite – first selecting the lowest value from each input pair, then selecting the highest of those values – but it still accomplishes the same function. Either strategy outputs the *middle* value of the three input signals[27].

Although either of these methods of obtaining a median measurement requires four signal selector functions, it is quite common to find function blocks available in control systems ready to perform the median select function all in a single block. The median-select function is so common to redundant sensor control systems that many control system manufacturers provide it as a standard function unto itself.

This is certainly true in the FOUNDATION Fieldbus standard, where two standardized function blocks are capable of this function. The CS (Control Selector) and the ISEL (Input Selector) blocks are shown in Figure 26.70.

[27] In order to understand how this works, try a "thought experiment" for each function block network by arbitrarily assigning three different numerical values for A, B, and C, then determine which of those three values becomes the output value.

Figure 26.70: CS and ISEL function blocks in FOUNDATION Fieldbus programming

Of these two Fieldbus function blocks, the latter (ISEL) is expressly designed for selecting transmitter signals, whereas the former (CS) is best suited for selecting controller outputs with its "back calculation" facilities designed to modify the response of all de-selected controllers. Using the terminology of this book section, the ISEL function block is best suited for *selector* strategies, while the CS function block is ideal for *override* strategies.

If receiving three "good" inputs, the ISEL function block will output the middle (median) value of the three. If one of the inputs carries a "bad" status[28], the ISEL block outputs the averaged value

[28]In FOUNDATION Fieldbus, each and every signal path not only carries the signal value, but also a "status" flag declaring it to be "Good," "Bad," or "Uncertain." This status value gets propagated down the entire chain of

of the remaining two (good) inputs. Note how this function block also possesses individual "disable" inputs, giving external boolean (on/off) signals the ability to disable any one of the transmitter inputs to this block. Thus, the ISEL function block may be configured to de-select a particular transmitter input based on some programmed condition other than internal diagnostics.

If receiving four "good" inputs, the ISEL function block normally outputs the average value of the two middle (median) signal values. If one of the four inputs becomes "bad" is disabled, the block behaves as a normal three-input median select.

A general design principle for redundant transmitters is that exactly two transmitters should *never* be installed to measure the same process variable. Instead, install three (minimum). The problem is that two transmitters don't provide enough information for "voting" to select a signal if the two transmitters disagree (and a deviation tolerance must be used to signal an alarm when disagreement occurs). In a three-transmitter system, the function blocks may select the median signal value, or average the "best 2 out of 3." If there are just two transmitters installed, and they do not substantially agree with one another, it is impossible to determine which one should be trusted[29].

### 26.6.3 Override controls

An "override" control strategy involves a selection between two or more controller *output* signals, where only one controller at a time gets the opportunity to exert control over a process by passing its signal to the output channel. All other controllers are thus *overridden* by the selected controller.

Consider the water pumping system shown in Figure 26.71, where a water pump is driven by a variable-speed electric motor to draw water from a well and provide constant water pressure to a customer.

connected function blocks, to alert dependent blocks of a possible signal integrity problem if one were to occur.

[29]This principle holds true even for systems with no function blocks "voting" between the redundant transmitters. Perhaps the installation consists of two transmitters with remote indications for a human operator to view. If the two displays substantially disagree, which one should the operator trust? A set of *three* indicators would be much better, providing the operator with enough information to make an intelligent decision on which display(s) to trust.

Figure 26.71: A water pumping system

Incidentally, this is an excellent application for a variable-speed motor as the final control element rather than a control valve. Reducing pump speed in low-flow conditions will save a lot of energy over time compared to the energy that would be wasted by a constant-speed pump and control valve.

A potential problem with this system is the pump running "dry" if the water level in the well gets too low, as might happen during summer months when rainfall is low and customer demand is high. If the pump runs for too long with no water passing through it, the seals will become damaged. This will necessitate a complete shut-down and costly rebuild of the pump, right at the time customers need it the most.

One solution to this problem would be to install a level switch in the well, sensing water level and shutting off the electric motor driving the pump if the water level ever gets too low. This control scheme is shown in Figure 26.72.

Figure 26.72: A hardwired level interlock protects against running the pump dry

This may be considered a kind of "override" strategy, because the low-level switch overrides the pressure controller's command for the pump to turn. It is also a crude solution to the problem, for while it protects the pump from damage, it does so at the cost of completely shutting off water to customers. One way to describe this control strategy would be to call it a *hard override* system, suggesting the uncompromising action it will take to protect the pump. [30].

A better solution to the dilemma would be to have the pump merely slow down as the well water level approaches a low-level condition. This way at least the pump could be kept running (and some amount of pressure maintained), decreasing demand on the well while maintaining curtailed service to customers and still protecting the pump from dry-running. This would be termed a *soft override* system.

Such a control strategy may be created by replacing the well water level switch with a level *transmitter*, connecting the level transmitter to a level controller, and using a low-select relay or function block to select the lowest-valued output between the pressure and level controllers. The level controller's setpoint will be set at some low level above the acceptable limit for continuous pump operation. This control variation is shown in Figure 26.73.

[30]Hard overrides are often also called hardware interlocks, and are usually implemented by direct action in the circuitry as opposed to actions in software inside a digital controller. Note that the level switch interacts with the motor control circuitry, not the controller program.

Figure 26.73: An override control strategy protects against running the pump dry

If ever the well's water level goes below the setpoint, the level controller will command the pump to slow down, even if the pressure controller is calling for a higher speed. The level controller will have *overridden* the pressure controller, prioritizing pump longevity over customer demand.

Bear in mind that the concept of a low-level switch completely shutting off the pump is not an entirely bad idea. In fact, it might be prudent to integrate such a hardwired interlock in the motor control system, just in case something goes wrong with the level controller (e.g., an improperly adjusted setpoint or poor tuning) or the low-select function.

With two layers of safety control for the pump, this system provides both a "soft constraint" providing moderated action and a "hard constraint" providing aggressive action to protect the pump from dry operation as shown in Figure 26.74.

Figure 26.74: Both an override control strategy and a hardwired interlock protect against running the pump dry

In order that these two levels of pump protection work in the proper order, the level controller's (LC) setpoint needs to be set to a higher value than the low level alarm's (LAL) trip point. Note that while the level switch can still shut off the pump by interacting with the motor control circuitry, it also lets the controller know when this has occurred. This is typically accomplished by using a level switch with two sets of contacts (which can sometimes be difficult to find), or by using the level switch to control a relay in the motor control circuit that has two sets of contacts, one of which shuts down the motor power and the other which is wired to the control system discrete inputs.

A very important consideration for any override control strategy is how to manage integral windup. Any time a controller with any integral (reset) action at all is de-selected by the selector function, the integral term of the controller will have the tendency to wind up (or wind down) over time. With the output of that controller decoupled from the final control element, it can have no effect on the process variable. Thus, integral control action – the purpose of which being to constantly drive the output signal in the direction necessary to achieve zero error between process variable and setpoint – will work in vain to eliminate an error it cannot influence. If and when control is handed back to that controller, the integral action will have to spend time "winding" the other way to un-do what it did while it was deselected.

Thus, override controls demand some form of integral windup limits that engage when a controller is de-selected. Methods of accomplishing this function are discussed in an earlier section on limit controls (section 26.6.1, beginning on page 1136).

# References


[1] T. E. Marlin, Process Control, 2nd ed. New York, NY: McGraw-Hill, 2000.

[2] B. A. Ogunnaike and W. H. Ray, Process Dynamics, Modeling, and Control. New York, NY: Oxford University Press, 1994.

[3] C. A. Smith and A. Corripio, Principles and Practice of Automatic Process Control, 3rd ed. New York, NY: John Wiley and Sons, 2006.

[4] G. T. Austin, Shreve's Chemical Process Industries, 5th ed. New York, NY: McGraw-Hill, 1984.

[5] B. Lipták, Ed., Instrument Engineers Handbook: Process Control and Optimization, 4th ed. Boca Raton, FL: CRC Press, 2006, vol. 2.

[6] F. G. Shinskey, Process Control Systems: Application, Design, and Tuning, 4th ed. Boston, MA: McGraw-Hill, 1996.

# Part IV

# Appendices

# Appendix A

# Instrumentation Connections

All instruments connect to their respective processes and to each other by means of pipe, tube, and/or wires. Improper installation of these connective lines can make the difference between success or failure in an installation. Safety is also impacted by improper connections between instruments and the process, and from instrument to instrument.

## A.1 Pipe and pipe fittings

Pipe is a hollow structure designed to provide an enclosed pathway for fluids to flow, usually manufactured from cast metal (although plastic is a common pipe material for many industrial applications). This section discusses some of the more common methods for joining pipes together (and joining pipe ends to equipment such as pressure instruments).

### A.1.1 Flanged pipe fittings

In the United States of America, most large industrial pipes are joined together by *flanges*. A pipe "flange" is a ring of metal, usually welded to the end of a pipe, with holes drilled in it parallel to the pipe centerline to accept several bolts, as shown in Figure A.1.

Figure A.1: Flanged pipe connections

Flange joints are made pressure-tight by inserting a donut-shaped gasket between the flange pairs prior to tightening the bolts. Gaskets are manufactured from materials softer than the hard steel of the flanges. When sandwiched between a pair of flanges, the gasket will be "crushed" between them to seal all potential leak paths.

A photograph showing a Rosemount magnetic flowmeter installed with 4-bolt flange fittings appears in Figure A.2.

Figure A.2: A flowmeter installed using flange fittings

Examination of the flanged connections closely shows the gap between the flange faces created by the thickness of the gasket material "sandwiched" between the flange pairs.

A common method of installing such a flange gasket is to first install only half of the bolts (in the holes lower than the centerline of the pipe), drop the gasket between the flanges, insert the remaining bolts, the proceed to tighten all bolts to the proper torques as shown in Figure A.3.

Figure A.3: Method of making flanged connections

Flanges differ with regard to their sealing design and required gasket type. In the United States, one of the most common flange "face" designs is the *raised-face* (RF) flange, designed to seal against a gasket by means of a set of concentric grooves machined on the face of the flange. These grooves form a sealing surface with far greater leakage path length than if the faces were smooth, thus discouraging leakage of process fluid under pressure.

Another flange face design is called *ring-type joint* (RTJ). In this design, a special metal ring sits inside a groove machined into the faces of both mating flanges, crushing and filling that groove when the flanges are properly tightened together. RTJ flanges are typically found on high-pressure applications where leakage control is more challenging. The grooves in RTJ flanges must be completely free of foreign material, and well-formed (not distorted) in order to achieve proper sealing.

In the United States, flanges are often rated according to a system of "pressure classes" defined in the ANSI (American National Standards Institute) standard 16.5. These pressure classes are designated by numerical values followed by "pound", "lb", or "#". Common ANSI ratings include the 150#, 300#, 400#, 600#, 900#, 1500#, and 2500# pressure classes. It should be noted that these class numbers do *not* refer directly to pressure ratings in units of PSI, but that they do scale with pressure (i.e., a 600# flange will have a greater pressure rating than a 300# flange, all other factors being equal). Pressure ratings not only vary with the "class" of the flange, but also with operating temperature, as metals tend to weaken at elevated temperature.

Originally, the ANSI class designations were based on the ratings of these flanges in steam line service. A 250# flange, for instance, was rated such because it was designed to be used in piping service where the fluid was steam at 250 PSI (and 400 degrees Fahrenheit). As metallurgy advanced, these flanges became capable of handling higher pressures at higher temperatures, but the original "pound" rating remained[1]. This state of affairs is not unlike the "tonnage" rating of American light trucks: a "one-ton" truck is actually capable of hauling far more than 2000 pounds of cargo. The "one-ton" designation refers to a particular design which used to be rated for approximately 2000 pounds, but through advances in metallurgy and manufacturing is now able to carry well over that rating.

Piping flanges and components must have matching flange ratings and sizes in order to properly function. For example, a control valve with a flanged body rated as a 4-inch ANSI class 300# can only be properly joined to another 4-inch ANSI class 300# pipe flange. The physical integrity of the piping system will be jeopardized if mis-matched pressure-class flanges are connected together. Proper gasket types must also be selected to coordinate with the pressure class of the mating flanges. Thus, each and every flanged joint must be considered a complete *system*, with integrity ensured only if all components comprising that system are designed to work together.

A very important procedure to observe when tightening the bolts holding two flanges together is to evenly distribute the bolt pressure, so that no single region of the flange receives significantly more bolt pressure than any other region. In an ideal world, all bolts would be tightened to the same torque limit *simultaneously*. However, since this is impossible with just a single wrench, the best alternative is to tighten the bolts in alternating sequence, in stages of increasing torque. An illustrative torque sequence is shown in Figure A.4 (the numbers indicate the order in which the bolts should be tightened).

[1] EBAA Iron Sales, Inc published a two-page report in 1994 ("Connections" FL-01 2-94) summarizing the history of flange "pound" ratings, from the ASME/ANSI B16 standards.

Figure A.4: Method of tightening bolts on flanged connections

With one wrench, tighten each bolt to a preliminary torque in the sequence shown. Then, repeat the tightening sequence with additional torque for a couple more cycles until all bolts had been tightened to the recommended torque value. Note how the torque sequence alternates between four quadrants of the flange, ensuring the flanges are evenly compressed together as all bolts are gradually tightened. This technique of alternating quadrants around the circle is often referred to as *cross-torquing*.

Special wrenches called *torque wrenches* exist for the purpose of measuring applied torque during the tightening process. In critical, high-pressure applications, the actual *stretch* of each flange bolt is measured as a direct indication of bolting force. A special bolt sold under the brand name of *Rotabolt* contains it own built-in strain indicator, letting the mechanic know when the bolt has been sufficiently tightened regardless of the tool used to tighten it.

Another important procedure to observe when working with flanged pipe connections is to loosen the bolts on the *far* side of the flange before loosening the bolts on the near side of the flange. This is strictly a precautionary measure against the spraying of process fluid toward the face or body in the event of stored pressure inside of a flanged pipe. By reaching over the pipe to first loosen flange bolts on the far side, if any pressure happens to be inside the pipe, it should leak there first, venting the pressure in a safer direction.

### A.1.2 Tapered thread pipe fittings

For smaller pipe sizes, *threaded fittings* are more commonly used to create connections between pipes and between pipes and equipment (including some instruments). A very common design of threaded pipe fitting is the *tapered* pipe thread design. The intent of a tapered thread is to allow the pipe and fitting to "wedge" together when engaged, creating a joint that is both mechanically rugged and leak-free.

When male and female tapered pie threads are first engaged, they form a loose junction as shown in Figure A.5.

Figure A.5: Initial tapered pipe thread connection

After tightening, however, the tapered profile of the threads acts to wedge both male and female pieces tightly together as shown in Figure A.6.

Figure A.6: Completed tapered pipe thread connection

Several different standards exist for tapered-thread pipe fittings. For each standard, the angle of the thread is fixed, as is the angle of taper. Thread *pitch* (the number of threads per unit length) varies with the diameter of the pipe fitting[2].

[2]For example, 1/8 inch NPT pipe fittings have a thread pitch of 27 threads per inch. 1/4 inch and 3/8 inch NPT fittings are 18 threads per inch, 1/2 inch and 3/4 inch NPT fittings are 14 threads per inch, and 1 inch through 2 inch NPT fittings are 11.5 threads per inch.

In the United States, the most common tapered thread standard for general-purpose piping is the *NPT*, or *National Pipe Taper* design. NPT threads have an angle of $60^o$ and a taper of $1^o$ 47' ($1.7833^o$) as shown in Figure A.7.

Figure A.7: National Pipe Tapered (NPT) thread

NPT pipe threads must have some form of *sealant* applied prior to assembly to ensure pressure-tight sealing between the threads. Teflon tape and various liquid pipe "dope" compounds work well for this purpose. Sealants are necessary with NPT threads for two reasons: to lubricate the male and female pieces (to guard against galling the metal surfaces), and also to fill the spiral gap formed between the root of the female thread and the crest of the male thread (and visa-versa).

NPTF (National Pipe Thread) pipe threads are engineered with the same thread angle and pitch as NPT threads, but carefully machined to avoid the spiral leak path inherent to NPT threads. This design – at least in theory – avoids the need to use sealant with NPTF threads to achieve a pressure-tight seal between male and female pieces, which is why NPTF threads are commonly referred to as *dryseal*. However, in practice it is still recommended that some form of sealant be used (or at the very least some form of thread *lubricant*) in order to achieve reliable sealing.

ANPT (Aeronautical National Pipe Tapered) is identical to NPT, except with a greater level of precision and quality for its intended use in aerospace and military applications.

Another tapered-thread standard is the *BSPT*, or *British Standard Pipe Tapered*. BSPT threads have a narrower thread angle than NPT threads ($55^o$ instead of $60^o$) but the same taper of $1^o$ 47' ($1.7833^o$) as shown in Figure A.8.

Figure A.8: British Standard Pipe Tapered (BSPT) thread

### A.1.3 Parallel thread pipe fittings

An alternative to tapered threads in pipe joints is the use of parallel threads, similar to the threads of machine screws and bolts. Since parallel threads are incapable of forming a pressure-tight seal on their own, the sealing action of a parallel thread pipe fitting must be achieved some other way. This function is usually met with an O-ring or gasket.

In the United States, a common design of parallel-thread pipe fitting is the *SAE straight thread*, named after the *Society of Automotive Engineers*, and shown in Figure A.9.

Figure A.9: SAE straight thread

Sealing is accomplished as the O-ring is compressed against the shoulder of the female fitting.

The threads serve only to provide force (not fluid sealing), much like the threads of a fastener.

Another parallel-thread pipe standard is the *BSPP*, or *British Standard Pipe Parallel*. Like the BSPT (tapered) standard, the thread angle of BSPP is $55^o$. Like the SAE parallel-thread standard, sealing is accomplished by means of an O-ring which compresses against the shoulder of the matching female fitting:

Figure A.10: British Standard Pipe Parallel (BSPP) thread

### A.1.4 Sanitary pipe fittings

Food processing, pharmaceuticals manufacturing, and biological research processes are naturally sensitive to the presence of micro-organisms such as bacteria, fungi, and algae. It is important in these processes to ensure the absence of harmful microorganisms, for reasons of both human health and quality control. For this reason, the process piping and vessels in these industries is designed first and foremost to be thoroughly cleaned without the need for disassembly. Regular cleaning and sterilization cycles are planned and executed between production schedules (batches) to ensure no colonies of harmful microorganisms can grow.

A common *Clean-In-Place* (CIP) protocol consists of flushing all process piping and vessels with alternating acid and caustic solutions, then washing with purified water. For increased sanitization, a *Steam-In-Place* (SIP) cycle may be incorporated as well, flushing all process pipes and vessels with hot steam to ensure the destruction of any microorganisms.

An important design feature of any sanitary process is the elimination of any "dead ends" (often called *dead legs* in the industry), crevices, or voids where fluid may collect and stagnate. This includes any instruments contacting the process fluids. It would be unsafe, for example, to connect something as simple as a bourdon-tube pressure gauge to a pipe carrying biologically sensitive fluid(s), since the interior volume of the bourdon tube will act as a stagnant refuge for colonies of microorganisms to grow. This is illustrated in Figure A.11.

Figure A.11: Diagram showing a dead leg where microorganisms can grow and flourish

Instead, any pressure gauge must use an isolating diaphragm, where the process fluid pressure is transferred to the gauge mechanism through a sterile "fill fluid" that never contacts the process fluid. An example of this is shown in Figure A.12.

Figure A.12: Diagram showing an isolating diaprhagm and fill fluid

With the isolating diaphragm in place, there are no stagnant places for process fluid to collect and avoid flushing by CIP or SIP cycles.

Standard pipe fittings are problematic in sanitary systems, as tiny voids between the mating threads of male and female pipe fittings may provide refuge for micro-organisms. To avoid this problem, special *sanitary fittings* are used instead. These fittings consist of a matched pair of flanges, held together by an external clamp. An array of sanitary fittings on an instrument test bench appear in the photograph in Figure A.13.

Figure A.13: Diagram showing an isolating diaprhagm and fill fluid

The photograph in Figure A.14 shows the installation of a pressure transmitter on an ultra-pure water line using one of these sanitary fittings. The external clamp holding the two flanges together is clearly visible.

Figure A.14: Diagram showing an isolating diaprhagm and fill fluid

Sanitary pipe fittings are not limited to instrument connections, either. Figure A.15 contains

two photographs of process equipment (a ball valve in Figure A.15a and a pump in Figure A.15b) connected to process pipes using sanitary fittings.

(a) A ball valve

(b) A pump

Figure A.15: Other sanitary fitting connections

## A.2 Tube and tube fittings

Tube, like pipe, is a hollow structure designed to provide an enclosed pathway for fluids to flow. In the case of tubing, it is usually manufactured from rolled or extruded metal (although plastic is a common tube material for many industrial applications). This section discusses some of the more common methods for joining tubes together (and joining tube ends to equipment such as pressure instruments).

One of the fundamental differences between tube and pipe is that tube is *never* threaded at the end to form a connection. Instead, a device called a *tube fitting* must be used to couple a section of tube to another tube, or to a section of pipe, or to a piece of equipment (such as an instrument). Unlike pipes which are thick-walled by nature, tubes are thin-walled structures. The wall thickness of a typical tube is simply too thin to support threads.

Tubes are generally favored over pipe for small-diameter applications. The ability for skilled workers to readily cut and bend tube with simple hand tools makes it the preferred choice for connecting instruments to process piping. When used as the connecting units between an instrument

and a process pipe or vessel, the tube is commonly referred to as an *impulse tube* or *impulse line*[3].

### A.2.1 Compression tube fittings

By far the most common type of tube fitting for instrument impulse lines is the *compression-style* fitting, which uses a compressible *ferrule* to perform the task of sealing fluid pressure. The essential components of a compression tube fitting are the *body*, the *ferrule*, and the *nut*. The ferrule and body parts have matching conical profiles designed to tightly fit together, forming a pressure-tight metal-to-metal seal. Some compression fitting designs use a two-piece ferrule assembly, such as this tube fitting shown in Figure A.16[4] (prior to full assembly).

Figure A.16: A two-piece ferrule assembly

Just prior to assembly, the nut will cover the ferrule components and push them into the conical entrance of the fitting body, as shown in Figure A.17.

Figure A.17: A two-piece ferrule assembly prior to tightening the nut

After properly tightening the nut, the ferrule(s) will *compress* onto the outside circumference of the tube, slightly crimping the tube in the process and thereby locking the ferrules in place, as shown in Figure A.18.

[3]Impulse lines are alternatively called *gauge lines* or *sensing lines*.
[4]This happens to be a Swagelok brass instrument tube fitting being installed on a 3/8 inch copper tube.

Figure A.18: A two-piece ferrule assembly with the nut tightened

When assembling compression-style tube fittings, always precisely follow the manufacturer's instructions to ensure correct compression. For Swagelok-brand instrument tube fittings 1 inch in size and smaller, the general procedure is to tighten the nut 1-1/4 turns past finger-tight. Insufficient turning of the nut will fail to properly compress the ferrule around the tube, and excessive turning will over-compress the ferrule, resulting in leakage.

Swagelok provides special gauges which may be used to measure proper ferrule compression during the assembly process. The design of the gauge is such that its thickness will fit between the nut and fitting shoulder if the nut is insufficiently tightened, but will not fit if it is sufficiently tightened. Thus the gauge has the ability to reveal an under-tightened fitting, but not an over-tightened fitting. These gauges fit easily in the palm of one's hand, as shown in Figure A.19.

Figure A.19: This Swagelok gauge is used to measure proper ferrule compression

Such gauges are referred to in the industry as *no-go gap gauges*, because their inability to fit between the nut and body shoulder of a tube fitting indicates a properly-tightened fitting. In other words, the gauge fit will be "no-go" if the tube fitting has been properly assembled.

Photographs showing one of these gauges testing a properly-tightened fitting (Figure A.20a) versus an under-tightened fitting (Figure A.20b) appear in Figure A.20.

(a) A properly-tightened Swagelok fitting (b) An under-tightened Swagelok fitting

Figure A.20: Use of a Swagelok compression gauge

Parker is another major manufacturer[5] of instrument tube fittings, and their product line uses a single-piece ferrule instead of the two-piece design preferred by Swagelok. Like Swagelok fittings, Parker instrument fitting sized 1/4 inch to 1 inch require 1-1/4 turns past hand tight to properly compress the ferrule around the circumference of the tube. Parker also sells gauges which may be used to precisely determine when the proper amount of ferrule compression is achieved.

Regardless of the brand, compression-style instrument tube fittings are incredibly strong and versatile. Unlike pipe fittings, tube fittings may be disconnected and reconnected with ease. No special procedures are required to "re-make" a disassembled instrument fitting connection: merely tighten the nut "snug" to maintain adequate force holding the ferrule to the fitting body, but not so tight that the ferrule compresses further around the tube than it did during initial assembly.

A very graphic illustration of the strength of a typical instrument tube fitting is shown in the Figure A.21, where a short section of 3/8 inch stainless steel instrument tube was exposed to high liquid pressure until it ruptured. Neither compression fitting on either side of the tube leaked during the test, despite the liquid pressure reaching a peak of 23,000 psig before rupturing the tube[6]:

[5]So is Gyrolok, Hoke, and a host of others. It is not the intent to advertise for different manufacturers in this textbook, but merely to point out some of the more common brands an industrial I&C engineer or technician might encounter on the job.

[6]It should be noted that the fitting nuts became seized onto the tube due to the tube's swelling. The tube fittings may not have leaked during the test, but their constituent components should never be placed into service again!

Figure A.21: The tube failed before the compression fittings did during a pressure test

### A.2.2 Common tube fitting types and names

Tube fittings designed to connect a tube to pipe threads are called *connectors*. Tube fittings designed to connect one tube to another are called *unions*. Examples of each are shown in Figure A.22.

Figure A.22: Tube connector and union fittings

If a tube union joins together different tube sizes rather than tubes of the same size, it is called a *reducing union*.

A variation on the theme of tube connectors and unions is the *bulkhead* fitting. Bulkhead fittings are designed to fit through holes drilled in panels or enclosures to provide a way for a fluid line to

pass through the wall of the panel or enclosure. In essence, the only difference between a bulkhead fitting and a normal fitting is the additional length of the fitting "barrel" and a special nut used to lock the fitting into place in the hole. Figure A.23 shows three types of bulkhead fittings.

Figure A.23: Tube bulkhead fittings

Tubing *elbows* are tube connectors with a bend. These are useful for making turns in tube runs without having to bend the tubing itself. Like standard connectors, they may terminate in male pipe thread, female pipe threads, or in another tube end. Elbow fittings are illustrated in Figure A.24.

Figure A.24: Tube elbow fittings

These elbows shown in the above illustration are all $90^o$, but this is not the only angle available. $45^o$ elbows are also common.

*Tee* fittings join three fluid lines together. Tees may have one pipe end and two tube ends (*branch* tees and *run* tees), or three tube ends (*union* tees). The only difference between a branch tee and a run tee is the orientation of the pipe end with regard to the two tube ends. Example tee fittings are illustrated in Figure A.25.

Figure A.25: Tube tee fittings

Of course, branch and run tee fittings also come in female pipe thread versions as well. A variation of the theme of union tees is the *cross*, joining four tubes together, as shown in Figure A.26.

Figure A.26: Tube cross fitting

Special tube fittings are made to terminate tube connections, so they are sealed up instead of open. A piece designed to seal off the open end of a tube fitting is called a *plug*, while a piece designed to seal off the end of an open tube is called a *cap*. Figure A.27 shows examples of cap and plug fittings.

Figure A.27: Tube cap and plug fittings

### A.2.3 Bending instrument tubing

Tube bending is something of an art, especially when done with stainless steel tubing. It is truly magnificent to see a professionally-crafted array of stainless steel instrument tubes, all bends perfectly made, all terminations square, all tubes parallel when laid side by side and perfectly perpendicular when crossing.

If possible, a goal in tube bending is to eliminate as many connections as possible. Connections invite leaks, and leaks are problematic. Long runs of instrument tubing made from standard 20 foot tube sections, however, require junctions be made somewhere, usually in the form of tube *unions*. When multiple tube unions must be placed in parallel tube runs, it is advisable to offset the unions so it is easier to get a wrench around the tube nuts to turn them. The philosophy here, *as always*, is to build the tubing system with future work in mind. Figure A.28 is a photograph of several tube junctions that shows one way to do this.

Figure A.28: Offset unions in tube fitting allow maintenance activity if needed

If an instrument tube must connect between a stationary object and a vibrating object, a straight (square) run of tube is actually not desirable, since it will not have much flexibility to absorb the vibration. Instead, a *vibration loop* should be made in the tube, giving it the necessary elasticity to handle the vibrational stresses. An example of a vibration loop placed in the air supply tube going to a control valve appears in Figure A.29.

Figure A.29: Tube containing a vibration loop

When bending such a loop, it is helpful to use the circumference of a large pipe as a mandrel to form the tube rather than attempt to form a loop purely by hand.

## A.3 Electrical signal and control wiring

There is much to be said for neatness of assembly in electrical signal wiring. Even though the electrons don't "care" how neatly the wires are laid in place, human beings who must maintain the system certainly do. Not only are neat installations easier to navigate and troubleshoot, but they tend to inspire a similar standard of neatness when alterations are made[7].

The following photographs illustrate excellent wiring practice. Study them carefully, and strive to emulate the same level of professionalism in your own work!

Figure A.30 is 120 volt AC power distribution wiring. Note how the hoop-shaped "jumper" wires are all cut to (nearly) the same length, and how each of the wire labels is oriented such that the printing is easy to read.

[7] No one wants to become known as the person who "messed up" someone else's neat wiring job!

Figure A.30: Example of neat and workman-like wiring inside a control panel

Figure A.31 shows a great way to terminate multi-conductor signal cable to terminal blocks. Each of the pairs was twisted together using a hand drill set to very slow speed. Note how the end of the cable is wrapped in a short section of heat-shrink tubing for a neat appearance.

Figure A.31: Tube containing a vibration loop

Beyond aesthetic preferences for instrument signal wiring are several practices based on sound electrical theory. The following subsections describe and explain these wiring practices.

### A.3.1 Connections and wire terminations

Many different techniques exist for connecting electrical conductors together: twisting, soldering, crimping (using compression connectors), and clamping (either by the tension of a spring or under the compression of a screw) are popular examples. Most industrial field wiring connections utilize a combination of compression-style crimp "lugs" and screw terminals to attach wires to instruments and to other wires.

Figure A.32 shows a typical *terminal strip* or *terminal block* array whereby twisted-pair signal cables connect to other twisted-pair signal cables.

Figure A.32: A terminal strip consisting of terminal blocks

The bases of crimp-style compression lugs (also called ferrules) can be seen at the ends of the wires, just where they insert into the terminal block modules. These terminal blocks use screws to apply force which holds the wires in close electrical contact with a metal bar inside each block, but ferrules have been crimped on the end of each wire to provide a more rugged tip for the terminal block screw to grip. A close-up view in Figure A.33 shows what one of these straight compression lugs looks like on the end of a wire.

Figure A.33: A ferrule compression lug crimped on a wire

Also evident in Figure A.32 is the dual-level connection points on the left-hand side of each terminal block. Two pairs of twisted signal conductors connect on the left-hand side of each terminal block pair, where only one twisted pair of wires connects on the right-hand side. This also explains why each terminal block section has two screw holes on the left but only one screw hole on the right.

A close-up photograph of a single terminal block module in Figure A.34 shows how the screw-clamp system works. Into the right-hand side of this block a single wire (tipped with a ferrule) is clamped securely. No wire is inserted into the left-hand side.

Figure A.34: Side view of a screw-clamp terminal block

Some terminal blocks are *screwless*, using a spring clip to make firm mechanical and electrical contact with the wire's end. An example of a spring-clamp terminal block is shown in Figure A.35.

Figure A.35: Side view of a screwless spring-clamp terminal block

In order to extract or insert a wire end from or two a "screwless" terminal block, a narrow screwdriver is inserted into a hole in the block near the insertion point, then pivoted (like a lever) to exert force on the spring clip. Screwless terminal blocks are generally faster to terminate and un-terminate than screw type terminal blocks, and the pushing action of the release tool is gentler on the body[8] than the twisting action required to loosen and tighten screws.

Many different styles of modular terminal blocks are manufactured to suit different wiring needs. Some terminal block modules, for example, have multiple "levels" instead of just one. Figure A.36 shows a two-level terminal block with screwless wire clamps.

Figure A.36: Side view of a screwless spring-clamp terminal block

Figure A.37 shows a three-level terminal block with screw-type clamps. Note that one disadvantage is that the wire that is clamped in the middle level covers the screw of the lower level, and the wire that is clamped in the top level covers the screw of the middle level. If there isn't enough slack available to push the wires out of the way, the top wire may need to be removed in order to get a screwdriver into the hole for the screw on the middle level, and both the top and middle wires may need to be removed in order to get a screwdriver into the hole for the screw on the bottom level.

[8] An occupational hazard for technicians performing work on screw terminations is *carpal tunnel syndrome*, where repetitive wrist motion (such as the motions required to loosen and tighten screw terminals) damages portions of the wrist where tendons pass.

Figure A.37: Side view of a multilevel terminal block

Some multilevel terminal blocks provide the option of internal jumpers called plug jumpers to connect two or more levels together so they will be electrically common instead of electrically isolated. Note the small holes in the top and middle connection bars in the multilevel terminal block in Figure A.37. These holes allow a threaded or split-body plug to be inserted to jumper the top and middle layers together. Almost all terminal blocks provide the option of external jumpers to connect adjacent terminal blocks along the same level, usually called comb jumpers because they resemble the teeth of a comb. Using a comb jumper in the design would have reduced the amount of work required to make the jumper loops seen in Figure A.30.

Other modular terminal blocks include such features as LED indicator lamps, switches, fuses, diodes, resistors, and even resettable circuit breakers in their narrow width, allowing the placement of actual circuit components near connection points. Figure A.38 shows a swing-open fused terminal block module in the open position.

Figure A.38: Side view of a fused terminal block

Modular terminal blocks are useful for making connections with both solid-core and stranded metal wires. The clamping force applied to the wire's tip is direct, with no sliding or other motions involved. Some terminal blocks, however, are less sophisticated in design. Figure A.39 shows a pair of "isothermal" terminals designed to connect thermocouple wires together (refer to 11.4.5 for why an isothermal block is needed). The bare tip of the screw applies pressure to the wire inserted into the block.

Figure A.39: Primitive isothermal terminal block for connecting thermocouple wires

The rotary force applied to each wire's tip by these screws necessitates the use of solid wire. Stranded wire would become frayed by this combination of forces.

Many field instruments, however, do not possess "block" style connection points at all. Instead, they are equipped with pan-head machine screws designed to compress the wire tips directly between the heads of the screws and a metal plate below.

Solid wires may be adequately joined to such a screw-head connection point by wrapping the bare wire end around the screw's circumference a minimum of 3/4 of the way around and tightening the head on top of the wire, as is the case with the two short wire stubs terminated on this instrument in Figure A.40.

Figure A.40: Terminating with screw-head connections

The problem with directly compressing a wire tip beneath the head of a screw is that the tip is subjected to both compressive and shear forces. As a result, the wire's tip tends to become mangled with repeated connections. Furthermore, tension on the wire will tend to turn the screw, potentially loosening it over time.

This termination technique is wholly unsuitable for stranded wire[9], because the shearing forces caused by the screw head's rotation tends to "fray" the individual strands. The best way to attach a stranded wire tip directly to a screw-style connection point is to first crimp a flat compression-style *terminal* to the wire (not a ferrule). The flat metal "lug" portion of the terminal is then inserted underneath the screw head, where it can easily handle the shearing and compressive forces exerted by the head.

Figure A.41 shows five such stranded-copper wires connected to screw-style connection points on a field instrument using compression-style terminals:

[9] An exception is when the screw is equipped with a square washer underneath the head, designed to compress the end of a stranded wire with no shear forces. Many industrial instruments have termination points like this, for the express purpose of convenient termination to either solid or stranded wire ends.

Figure A.41: Terminating with screw-head connections using compression-style terminals

Compression-style terminals come in two basic varieties: *fork* and *ring*. An illustration of each type is shown in Figure A.42.

Figure A.42: Compression-style terminals

Fork terminals are easier to install and remove, since they merely require loosening of the connector screw rather than removal of the screw. Ring terminals are more secure, since they cannot fall off of the connection point if the screw ever accidently loosens.

Just as direct termination underneath a screw head is wholly unsuitable for stranded wires, compression-style terminals are wholly unsuitable for solid wire. Although the initial crimp may feel secure, compression terminals lose their tension rapidly on solid wire, especially when there is any motion or vibration stressing the connection. Compression terminals should only be crimped to stranded wire!

Properly installing a compression-type terminal on a wire end requires the use of a special *crimping* tool. Figure A.43 shows one of these tools in use.

Figure A.43: A commom electrician's multitool being used to crimp a compression terminal onto a wire

Note the different places on the crimping tool, labeled for different wire sizes (gauges). One location is used for 16 gauge to 10 gauge wire, while the location being used in the photograph is for wire gauges 22 through 18 (the wire inside of the crimped terminal happens to be 18 gauge).

This particular version of a crimping tool performs most of the compression on the underside of the terminal barrel, leaving the top portion undisturbed. The final crimped terminal (viewed from the top) is shown in Figure A.44.

Figure A.44: A commom electrician's multitool being used to crimp a compression terminal onto a wire

### A.3.2 DIN rail

An industry-standard structure for attaching terminal blocks and small electrical components to flat metal panels is something called a *DIN rail*. This is a narrow channel of metal – made of bent sheet steel or extruded aluminum – with edges designed for plastic components to "clip" on. Figure A.45 shows terminal blocks, relay sockets, fuses, and more terminal blocks mounted to a horizontal length of DIN rail in a control system enclosure.

Figure A.45: Components mounted on DIN rail inside an enclosure

Two photographs of a terminal block cluster clipped onto a length of DIN rail – one from above (Figure A.46a) and one from below (Figure A.46b) – show how specially-formed arms on each terminal block module fit the edges of the DIN rail for a secure attachment.

(a) Top view (b) Bottom view

Figure A.46: DIN rail with terminal blocks snapped onto it

The DIN rail itself mounts on to any flat surface by means of screws inserted through the slots in its base. In most cases, the flat surface in question is the metal subpanel of an electrical enclosure to which all electrical components in that enclosure are attached.

An obvious advantage of using DIN rail to secure electrical components versus individually attaching those components to a subpanel with their own sets of screws is convenience: much less labor is required to mount and unmount a DIN rail-attached component than a component attached with its own set of dedicated screws. This convenience significantly eases the task of altering a panel's configuration. With so many different devices manufactured for DIN rail mounting, it is easy to upgrade or alter a panel layout simply by unclipping components, sliding them to new locations on the rail, or replacing them with other types or styles of components.

Figure A.47 shows some of the diversity available in DIN rail mount components. From left to right are four relays, a power supply, and three HART protocol converters, all clipped to the same extruded aluminum DIN rail.

Figure A.47: Many different brands and types of components can be mounted on DIN rail

As previously mentioned, DIN rail is available in both stamped sheet-steel and extruded aluminum forms. A comparison of the two materials is shown in Figure A.48, with sheet steel on the left (Figure A.48a) and aluminum on the right (Figure A.48b).

(a) Sheet steel (b) Aluminum

Figure A.48: Types of DIN rail

The form of DIN rail shown in all photographs so far is known as "top hat" DIN rail. A variation in DIN rail design is the so-called "G-type" rail, with a markedly different shape:

Figure A.49: G-type DIN rail

Fortunately, many modular terminal blocks are formed with the ability to clip to either style of DIN rail, such as these two specialty blocks in Figure A.50, the left-hand example (Figure A.50a) being a terminal block with a built-in disconnect switch, and the right-hand example (Figure A.50b) being a "grounding" terminal block whose termination points are electrically common to the DIN rail itself.

(a) Disconnect-style (b) Grounding

Figure A.50: Specialty terminal blocks

The bottom structure of each block contains formations designed to clip either to the edges of a standard ("top hat") DIN rail or to a "G" shaped DIN rail.

A nice feature of many DIN rail type terminal blocks is the ability to attach preprinted terminal numbers. This makes documentation of wiring much easier, with each terminal connection having its own unique identification number, as shown in Figure A.51.

Figure A.51: Terminal blocks with preprinted labels installed

### A.3.3 Cable routing

In the interest of safety and longevity, one cannot simply run electrical power and signal cables randomly between different locations. Electrical cables must be properly supported to relieve mechanical stresses on the conductors, and protected from harsh conditions such as abrasion which might degrade the insulation.

A traditional and rugged technique for cable routing is *conduit*, either metal or plastic (PVC). Conduit resembles piping used to convey fluids, except that it is much thinner-walled than fluid pipe and is not rated to handle internal pressure as pipe is. In fact, threaded conduit uses the same thread pitch and diameter standards as NPT (National Pipe Taper) fluid pipe connections.

Metal conduit naturally forms a continuously-grounded enclosure for conductors which not only provides a measure of protection against electrical shock (all enclosures and devices attached to the conduit become safely grounded through the conduit) but also shields against electrostatic interference. This is especially important for power wiring to and from devices such as rectifiers and variable-frequency motor drive units, which have a tendency to broadcast large amounts of electromagnetic noise.

Plastic conduit, of course, provides no electrical grounding or shielding because plastic is a non-conductor of electricity. However, it is superior to metal conduit with regard to chemical corrosion resistance, which is why it is used to route wires in areas containing water, acids, caustics, and other wet chemicals.

*Thin-wall* conduit is made with metal so thin that threads cannot be cut into it. Instead, special connectors are used to join "sticks" of thin-wall conduit together, and to join thin-wall conduit to electrical enclosures. Figure A.52 shows several runs of thin-wall conduit. Two of those conduits have been severed following a wiring change, exposing the conductors inside.

Figure A.52: Parallel runs of thin-wall electrometallic tubing (EMT) conduit

Installing cable into an electrical conduit is a task referred to as *cable pulling*, and it is something of an art. Cable "pulls" may be especially challenging if the conduit run contains many bends, and/or is close to capacity in terms of the number and size of conductors it already holds. A good practice is to always leave a length of nylon *pull string* inside each length of conduit, ready to use for pulling a new wire or cable through. When performing a wire "pull," a new length of nylon pull string is pulled into the conduit along with the new wires, to replace the old pull string as it is pulled out of the conduit. Special lubricating "grease" formulated for electrical wiring may be applied to conductors pulled into a conduit, to reduce friction between those new conductors and the conductors already inside the conduit.

When connecting electrical conduit to end-point devices, it is common to use flexible *liquid-tight conduit* as a connector between the rigid metal (or plastic) conduit and the final device. This provides some stress relief to the conduit in the event the device moves or vibrates, and also allows more freedom in positioning the device with respect to the conduit. Figure A.53 shows a motor-operated control valve with two runs of liquid-tight conduit routing conductors to it.

Figure A.53: Flexible liquid-tight conduit used for final connections to a valve

Liquid-tight conduit comes in two general varieties: metallic and non-metallic. The metallic kind contains a spiraled metal sheath just underneath the plastic outer coating to provide a continuously-grounded shield much the same way that rigid metal conduit does. Non-metallic liquid-tight conduit is nothing more than plastic hose, providing physical protection against liquid exposure and abrasion, but no electrical grounding or shielding ability.

Another technique for cable routing is the use of *cable tray*. Trays may be made of solid steel wire for light-duty applications such as instrument signal cabling or computer network cabling,

or they may be made of steel or aluminum channel for heavy-duty applications such as electrical power wiring. Unlike conduit, cable trays are open, leaving the cables exposed to the environment. This often necessitates special cable insulation rated for exposure to ultraviolet light, moisture, and other environmental wear factors. A decided advantage of cable trays is ease of cable installation, especially when compared to electrical conduit.

While cable tray does provide a continuously-grounded surface for electrical safety the same as metal conduit, cable tray does *not* naturally provide shielding for the conductors because it does not completely enclose the conductors the way metal conduit does.

An example of light-duty cable tray appears in Figure A.54. The cable tray is made of solid steel wire, bent to form a "basket" to support dozens of yellow Ethernet cables.

Figure A.54: Light-duty cable tray

Heavy-duty cable tray appears throughout Figure A.55, supporting large-gauge power conductors for electric motors at a wastewater treatment facility. Here, the cable tray has the appearance of an aluminum ladder laid horizontally, with extruded rails and tubular rungs providing physical support for the cables.

Figure A.55: Heavy-duty cable tray

Even heavier cables trays supporting feeder cables from a stationary transformer and switchgear cabinets appear in Figure A.56.

Figure A.56: Another example of heavy-duty cable tray

A special form of wiring often seen in industrial facilities for power distribution is *busway*, also known as *bus duct*. These are rectangular sheet-metal tubes containing pre-fabricated copper busbars for the conduction of three-phase AC power. Special junction boxes, "tees," and tap boxes allow busways to extend and branch to other busways and/or standard conductor wiring.

Busways are used in indoor applications, often in motor control center (MCC) and power distribution center rooms to route electrical power to and from large disconnect switches, fuses, and circuit breakers. In Figure A.57, busway is used to distribute power along the ceiling of an MCC room, alongside regular rigid conduit.

Figure A.57: Busway

As useful and neat in appearance as busways are, they are definitely limited in purpose. Busways are only used for electrical power distribution; not for instrumentation, control, or signaling purposes.

Two materials useful for neatly routing power, signal, and instrumentation conductors inside an enclosure are *wire duct* and *wire loom*. Wire duct is a plastic channel with slotted sides, designed to be attached to the subpanel of an enclosure along with all electrical devices inside that enclosure. Wires pass from the devices to the duct through the slots (gaps) in the sides of the duct, and are enclosed by a removable plastic cover that snaps onto the top of the duct. A common brand name of wire duct in the industry is *Panduit*, and so people refer often to wire duct as "Panduit" whether or not that particular brand is the one being used[10]. Wire loom is a loose spiral tube made of plastic, used to hold a group of individual wires together into a neat bundle. Wire loom is frequently used when a group of conductors must periodically flex, as is the case of a wire bundle joining devices inside a panel to other devices mounted on the hinging door of that panel.

Figure A.58 shows both wire duct (grey plastic) and wire loom inside an instrumentation panel. The wire duct is grey in color, and the loom is a plastic spiral surrounding the bundle of wires near the door hinge.

[10]This is similar to people referring to adhesive bandages as "Band-Aids" or facial tissues as "Kleenex," because those particular brands have become popular enough to represent an entire class of device.

Figure A.58: Wire duct and wire loom

### A.3.4 Signal coupling and cable separation

If sets of wires lie too close to one another, electrical signals may "couple" from one wire (or set of wires) to the other(s). This can be especially detrimental to signal integrity when the coupling occurs between AC power conductors and low-level instrument signal wiring such as thermocouple or pH sensor cables.

Two mechanisms of electrical "coupling" exist: *capacitive* and *inductive*. Capacitance is a property intrinsic to any pair of conductors separated by a dielectric (an insulating substance), whereby energy is stored in the electric field formed by voltage between the wires. The natural capacitance existing between mutually insulated wires forms a "bridge" for AC signals to cross between those wires, the strength of that "bridge" inversely proportional to the capacitive reactance ($X_C = \frac{1}{2\pi f C}$). Inductance is a property intrinsic to any conductor, whereby energy is stored in the magnetic field formed by current through the wire. Mutual inductance existing between parallel wires forms another "bridge" whereby an AC current through one wire is able to induce an AC voltage along the length of another wire.

Capacitive coupling between an AC power conductor and a DC sensor signal conductor is shown in Figure A.59.

Figure A.59: Capacitive coupling

If the potentiometric (i.e., the measurement is based on voltage) sensor happens to be a thermocouple and the receiving instrument a temperature indicator, the result of this capacitive coupling will be a "noisy" temperature signal interpreted by the instrument. This noise will be proportional to both the *voltage* and the frequency of the AC power.

Inductive coupling between an AC power conductor and a DC sensor signal conductor is shown in Figure A.60.

Figure A.60: Inductive coupling

If the potentiometric (i.e., the measurement is based on voltage) sensor happens to be a thermocouple and the receiving instrument a temperature indicator, the result of this inductive coupling will be a "noisy" temperature signal interpreted by the instrument. This noise will be proportional to both the *current* and the frequency of the AC power.

A simple way to reduce signal coupling is to simply separate conductors carrying incompatible signals. This is why electrical power conductors and instrument signal cables are almost never found in the same conduit or in the same ductwork together. Separation decreases capacitance between the conductors (recall that $C = \frac{A\epsilon}{d}$ where $d$ is the distance between the conductive surfaces). Separation also decreases the coupling coefficient between inductors, which in turn decreases mutual inductance (recall that $M = k\sqrt{L_1 L_2}$ where $k$ is the coupling coefficient and $M$ is the mutual inductance between two inductances $L_1$ and $L_2$). In control panel wiring, it is customary to route AC power wires in such a way that they do not run parallel to low-level signal wires, so that both forms of coupling may be reduced.

If conductors carrying incompatible signals *must* intersect in a panel, it is advisable to orient them so the crossing is perpendicular rather than parallel, as shown in Figure A.61.

Figure A.61: Minimizing coupling via perpendicular crossing power and signal conductors

Parallel conductor orientation reduces both inter-conductor capacitance *and* mutual inductance by two mechanisms. Capacitance between conductors is reduced by means of minimizing overlapping area ($A$) resulting from the perpendicular crossing. Mutual inductance is reduced by decreasing the coupling coefficient ($k$) to nearly zero since the magnetic field generated perpendicular to the current-carrying wire will be *parallel* and not perpendicular to the "receiving" wire. Since the vector for induced voltage is perpendicular to the magnetic field (i.e., parallel with the current vector in the "primary" wire) there will be no voltage induced along the length of the "receiving" wire.

The problem of power-to-signal line coupling is most severe when the signal in question is *analog* rather than *digital*. In analog signaling, even the smallest amount of coupled "noise" corrupts the signal. A digital signal, by comparison, will become corrupted only if the coupled noise is so severe that it pushes the signal level above or below a detection threshold is should not cross. This disparity is best described through illustration.

Two signals are shown in Figure A.62, coupled with equal amounts of noise voltage.

Figure A.62: Corrupted analog and digital signals

The peak-to-peak amplitude of the noise on the analog signal is almost 20% of the entire signal range (the distance between the lower- and upper-range values), representing a substantial degradation of signal integrity. Analog signals have infinite resolution, which means *any* change in signal amplitude has meaning. Any noise whatsoever is degrading to an analog signal because that noise (when interpreted by a receiving circuit) will be interpreted as changes in the quantity that signal is supposed to represent.

That same amount of noise imposed on a digital signal, however, causes no degradation of the signal except for one point in time where the signal attempts to reach a "low" state but fails to cross the threshold due to the noise. Other than that one incident represented in the pulse waveform, the rest of the signal is completely unaffected by the noise, because digital signals only have meaning above the "high" state threshold and below the "low" state threshold. Changes in signal voltage level caused by induced noise will not affect the meaning of digital data unless and until the amplitude of that noise becomes severe enough to prevent the signal's crossing through a threshold (when it should cross), or causes the signal to cross a threshold (when it should not).

Digital signals are far more tolerant of induced noise than analog signals, all other factors being equal. If a signal wire must (as a last resort) be pulled through a conduit filled with AC power conductors, and an analog signal (e.g., mV, 4-20 mA, 0-10 V) or a digital signal (e.g., EIA/TIA-485,

Ethernet) may be chosen, the best option is to choose the digital signal.

### A.3.5 Electric field (capacitive) decoupling

The fundamental principle invoked in *shielding* signal conductor(s) from external electric fields is that an electric field cannot exist within a solid conductor. Electric fields exist due to imbalances of electric charge. If such an imbalance of charge ever were to exist within a conductor, charge carriers (typically electrons) in that conductor would quickly diffuse through the material to equalize the imbalance, thus eliminating the electric field. Thus, electric flux lines may be found only in the dielectric (insulating media) between conductors as shown in Figure A.63.

Figure A.63: Electric flux lines are only found in insulating material, not conductors

Not only does this mean that static electric fields cannot exist within a conductor, but it also means electric flux lines cannot exist within the confines of a hollow conductor. This is illustrated in Figure A.64.

Figure A.64: Electric flux lines are never found inside a hollow conductor

In order for an electric field to span the dielectric space within the hollow conductor, there would have to be an imbalance of electric charge from one side of the conductor to the other, which would be immediately equalized by charge motion within the hollow shell. The interior space of the hollow conductor, therefore, is free from external electric fields imposed by conductors outside the hollow conductor. To state this differently, the interior of the hollow conductor is *shielded* from external electrostatic interference.

It is possible for an external electric field to penetrate a hollow conductor from the outside, but only if the conductive shell is left "electrically floating" with respect to another conductor placed within the shell. This is shown in Figure A.65.

Figure A.65: Electric flux lines surround a conductor placed within an electrically floating conductive shell

However, if the hollow shell is made electrically common to the negative side of the high-voltage source, the flux lines inside the sphere vanish as shown in Figure A.66, since there is no potential difference between the internal conductor and the conductive shell.

Figure A.66: Electric flux lines do not exist around a conductor placed within a electrically-connected conductive shell

If the conductor within the hollow sphere is elevated to a potential different from that of the high-voltage source's negative terminal, electric flux lines will once again exist inside the sphere, but they will reflect this second potential and not the potential of the original high-voltage source. This is seen in Figure A.67. In other words, an electric field will exist inside the hollow sphere, but it will be completely isolated from the electric field outside the sphere. Once again, the conductor inside is *shielded* from external electrostatic interference.

Figure A.67: Electric flux lines created by the internal conductor exist, but external electric flux lines do not

If conductors located inside the hollow shell are thus shielded from external electric fields, it means there cannot exist any capacitance between external conductors and internal (shielded) conductors. If there is no capacitance between conductors, there will never be capacitive coupling of signals between those conductors. This is desired for industrial signal cables to protect those signals from external interference[11].

This is how *shielded cables* are manufactured: conductors within the cable are wrapped in a conductive metal foil or conductive metal braid, which may be connected to ground potential (the "common" point between external and internal voltage sources) to prevent capacitive coupling between those external voltage sources and the conductors within the cable. Figure A.68 illustrates a properly grounded shielded signal cable.

[11] Incidentally, cable shielding likewise guards against strong electric fields *within* the cable from capacitively coupling with conductors outside the cable. This means the engineer may elect to shield "noisy" power cables instead of (or in addition to) shielding low-level signal cables. Either way, good shielding will prevent capacitive coupling between conductors on either side of a shield.

Figure A.68: Shielded cable that is properly grounded can reject noise coupling

Figure A.69 shows a set of signal cables with braided shield conductors all connected to a common copper "ground bus." This particular application happens to be in the control panel of a 500 kV circuit breaker, located at a large electrical power substation where strong electric fields abound.

Figure A.69: Shields terminated on a ground bus in a 500 kV circuit breaker

It is very important to ground *only one end* of a cable's shield, or else the possibility for a *ground loop* exists: a path for current to flow through the cable's shield resulting from differences in Earth

potential at the cable ends. Not only can ground loops induce noise in a cable's conductor(s), but in severe cases it can even overheat the cable and thus present a fire hazard! A ground loop is shown in Figure A.70.

Figure A.70: Shields grounded at both ends can create undesired ground loops

An alternative to shielding electric fields is to use *differential signaling* to help nullify the effects of capacitive coupling. The schematic diagram in Figure A.71 illustrates how this works.

Figure A.71: Common-mode noise rejection using differential signaling

The lack of a ground connection in the DC signal circuit prevents capacitive coupling with the AC voltage from corrupting the measurement signal "seen" by the instrument. Noise voltage *will* still appear between either signal wire and ground, but not *between* the two signal wires, which is all the instrument is able to measure. In other words, the noise voltage will be *common-mode* on the two signal wires, but it will not manifest itself as a *potential difference* which is all the receiving instrument senses.

Some industrial data communications standards such as EIA/TIA-485 (RS-485) use this technique to minimize the corrupting effects of electrical noise. To see a practical example of how this works in a data communications circuit, refer to the illustration in section 4.6.2 beginning on page 163 of this book.

### A.3.6 Magnetic field (inductive) decoupling

Magnetic fields, unlike electric fields, are exceedingly difficult to completely shield. Magnetic flux lines do not terminate, but rather *loop*. Thus, one cannot block a magnetic field, only redirect its path. A common method for magnetically shielding a sensitive instrument is to encapsulate it in an enclosure made of some material having an extremely high magnetic permeability ($\mu$): a shell offering much easier passage of magnetic flux lines than air. A material often used for this application is *mu-metal*, or *$\mu$-metal*, so named for its excellent magnetic permeability. Shielding using $\mu$-metal shield is illustrated in Figure A.72.

Figure A.72: Shielding from magnetic field flux using mu-metal

This sort of shielding is impractical for protecting signal cables from inductive coupling, as mu-metal is rather expensive and must be layered relatively thick in order to provide a sufficiently low-reluctance path to re-direct a majority of any imposed magnetic flux lines.

The most practical method of granting magnetic field immunity to a signal cable follows the differential signaling method discussed in the electric field de-coupling section, with a twist (literally). By *twisting* a pair of wires rather than allow them to lie along parallel straight lines, the effects of electromagnetic induction are vastly reduced.

The reason this works is best illustrated by drawing a differential signal circuit with two thick

wires, drawn first with no twist at all. Suppose the magnetic field shown =in Figure A.73 (with three flux lines entering the wire loop) happens to be *increasing* in strength at the moment in time captured by the illustration.

Figure A.73: Magnetic flux coupling into a conductor loop results in noise

According to Lenz's Law, a current will be induced in the wire loop in such a polarity as to oppose the increase in external field strength. In other words, the induced current tries to "fight" the imposed field to maintain zero net change. According to the right-hand rule of electromagnetism (tracing current in conventional flow notation), the induced current must travel in a counter-clockwise direction as viewed from above the wire loop. This induced current works against the DC current produced by the sensor, detracting from the signal received at the instrument.

When the external magnetic field strength diminishes, then builds in the opposite direction, the induced current will reverse. Thus, as the AC magnetic field oscillates, the induced current will also oscillate in the circuit, causing AC "noise" voltage to appear at the measuring instrument. This is precisely the effect that must be mitigated.

If the wires are twisted so as to create a series of loops instead of one large loop, the inductive effects of the external magnetic field tend to cancel as shown in Figure A.74.

Figure A.74: Magnetic flux coupling into a twisted conductor results in noise cancellation

Not all the lines of flux go through the same loop. Each loop represents a reversal of direction for current in the instrument signal circuit, and so the direction of magnetically-induced current in one loop directly opposes the direction of magnetically-induced current in the next. So long as the loops are sufficient in number and spaced close together, the net effect will be complete and total opposition between all induced currents, with the result of no net induced current and therefore no AC "noise" voltage appearing at the instrument.

In order to enjoy the benefits of magnetic *and* electric field rejection, instrument cables are generally manufactured as *twisted, shielded pairs*. The twists guard against magnetic (inductive) interference, while the grounded shield guards against electric (capacitive) interference.

### A.3.7 High-frequency signal cables

Electronic signals used in traditional instrumentation circuits are either DC or low-frequency AC in nature. Measurement and control values are represented in *analog* form by these signals, usually by the magnitude of the electronic signal (how many volts, how many milliamps, etc.). Modern electronic instruments, however, often communicate process and control data in *digital* rather than analog form. This digital data takes the form of high-frequency voltage and/or current pulses along the instrument conductors. The most capable *fieldbus* instruments do away with analog signaling entirely, communicating all data in digital form at relatively high speeds.

If the time period of a voltage or current pulse is less than the time required for the signal to travel down the length of the cable (at nearly the speed of light!), very interesting effects may occur. When a pulse propagates down a two-wire cable and reaches the end of that cable, the energy contained by that pulse must be absorbed by the receiving circuit or else be *reflected* back down the cable. To be honest, this happens in all circuits no matter how long or brief the pulses may be, but the effects of a "reflected" pulse only become apparent when the pulse time is short compared to the signal propagation time. In such short-pulse applications, it is customary to refer to the cable as a *transmission line*, and to regard it as a circuit component with its own characteristics (namely, a continuous impedance as "seen" by the traveling pulse).

This problem has a familiar analogy: an "echo" in a room. When stepping into a large room with hard wall, floor, and ceiling surfaces, echoes immediately result from any sound made. Holding a conversation in such a room can be quite difficult, as the echoed sounds superimpose upon the most recently-spoken sounds, making it difficult to discern what is being said. The larger the room, the longer the echo delay, and the greater the conversational confusion.

Echoes happen in small rooms, too, but they are generally too short to be of any concern. If the reflected sound(s) return quickly enough after being spoken, the time delay between the spoken (incident) sound and the echo (reflected) sound will be too short to notice, and conversation will proceed unhindered.

The "echo" problem may be addressed in two entirely different ways. One way is to eliminate the echoes entirely by adding sound-deadening coverings (carpet, acoustic ceiling tiles) and/or objects (sofas, chairs, pillows) to the room. Another way to address the problem of echoes interrupting a conversation is to *slow down the rate of speech*. If the words are spoken slowly enough, the time delay of the echoes will be relatively short compared to the period of each spoken sound, and conversation may proceed without interference (albeit at a reduced speed).

Both the problem of and the solutions for reflected signals in electrical cables follow the same patterns as the problem of and solutions for sonic echoes in a hard-surfaced room. If an electronic circuit receiving pulses sent along a cable receives both the incident pulse and an echo (reflected pulse) with a significant time delay separating those two pulses, the digital "conversation" will be impeded in the same manner that a verbal conversation between two or more people is impeded by echoes in a room. This problem may be avoided either by eliminating the reflected pulses entirely (by ensuring all the pulse energy is absorbed when it reaches the cable's end) or by slowing down the data transfer rate (i.e., longer pulses, lower frequencies) so that the reflected and incident pulse signals virtually overlap one another at the receiver.

High-speed "fieldbus" instrument networks apply the former solution (eliminate reflections) while the legacy HART instrument signal standard apply the latter (slow data rate). Reflections are eliminated in high-speed data networks by ensuring the two furthest cable ends are both "terminated" by a resistance value of the proper size (matching the characteristic impedance of the cable). The designers of the HART analog-digital hybrid standard chose to use slow data rates instead, so their instruments would function adequately on legacy signal cables where the characteristic impedance is not standardized.

The potential for reflected pulses in high-speed fieldbus cabling is a cause for concern among I&C engineers, because it represents a new phenomenon capable of creating faults in an instrument system. No longer is it sufficient to have tight connections, clean wire ends, good insulation, and proper shielding for a signal cable to faithfully convey a 4-20 mA DC instrument signal from one device to another. Now the technician must ensure proper termination in the characteristic

impedance of the cable and the absence of any discontinuities[12] (sharp bends or crimps) along the cable's entire length, in addition to all the traditional criteria, in order to faithfully convey a digital fieldbus signal from one device to another.

Signal reflection problems may be investigated using a diagnostic instrument known as a *time-domain reflectometer*, or *TDR*. These devices are a combination of pulse generator and digital-storage oscilloscope, generating brief electrical pulses and analyzing the returned (echoed) signals at one end of a cable. If a TDR is used to record the pulse "signature" of a newly-installed cable, that data may be compared to future TDR measurements on the same cable to detect cable degradation or wiring changes.

[12]The characteristic, or "surge," impedance of a cable is a function of its conductor geometry (wire diameter and spacing) and dielectric value of the insulation between the conductors. Any time a signal reaches an abrupt change in impedance, some (or all) of its energy is reflected in the reverse direction. This is why reflections happen at the unterminated end of a cable: an "open" is an infinite impedance, which is a huge shift from the finite impedance "seen" by the signal as it travels along the cable. This also means any sudden change in cable geometry such as a crimp, nick, twist, or sharp bend is capable of reflecting part of the signal. Thus, high-speed digital data cables must be installed more carefully than low-frequency or DC analog signal cables.

# References


[1] T. Croft and W. Summers, Eds., American Electrician's Handbook, 11th ed. New York, NY: McGraw-Hill, 1987.

[2] Connections, EBAA Iron Sales, Inc., 1994.

[3] R. Morrison, Grounding and Shielding Techniques in Instrumentation. New York, NY: Wiley, 1967.

[4] G. T. Austin, Shreve's Chemical Process Industries, 5th ed. New York, NY: McGraw-Hill, 1984.

[5] W. V. Graves, The Pipe Fitters Blue Book. Webster, TX: W.V. Graves Publisher, 1973.

[6] CPI Tube Fittings, Catalog 4230, Parker Hannifin Corporation, Cleveland, OH, 2000.

[7] Industrial Pipe Fittings and Adapters, Catalog 4300, Parker Hannifin Corporation, Columbus, OH, 2000.

[8] Fitting Installation Manual, Hoke Incorporated, Spartanburg, SC, 1999.

[9] Gaugeable Tube Fittings and Adapter Fittings, MS-01-140, Rev. 7, Swagelok Company, Solon, OH, 2004.

[10] Pipe Fittings, MS-01-147, Rev. 3, Swagelok Company, Solon, OH, 2010.

[11] Thread and End Connection Identification Guide, MS-13-77, Rev. 3, Swagelok Company, Solon, OH, 2005.

[12] Piping Joints Handbook, D/UTG/054/00, BP Amoco Dyce Office, Aberdeen, UK, 2000.

[13] Instrumentation Electrical Installation Details, Process Industry Practices Std. PCIEL000, 1999.

# Appendix B

# Example Sequence of Operations

## B.1 System Name

HVAC

## B.2 System ID Number

001

## B.3 Equipment List

F-1 Fan 1
F-2 Fan 2

## B.4 System Description

### B.4.1 General Description

The fans are used to cool the spaces they serve. Manual start/stop control and automatic temperature control and monitoring is provided for each fan unit.

### B.4.2 Philosophy of Operation

Each fan unit (F-XX, where XX is the unit number) lowers the space temperature while running. Each fan unit has a temperature control loop that uses return air temperature as the process variable and manipulates chilled water flow to the unit with a fail-open control valve. If the temperature of the air in a particular unit drops too low, a temperature switch removes power from the fan and opens the control valve fully to prevent freezing the coil.

## B.5 Control Loop Descriptions

### B.5.1 Loop TIC-XX

#### Hard-wired Functions

None.

#### Control Details

When the temperature goes above the setpoint as sensed by the temperature transmitter (TT-XX), a reverse-acting PI controller (TIC-XX) lowers its output to the I/P converter (TY-XX) which causes the fail-open control valve (TV-XX) to open further, thus admitting more chilled water to the cooling coil and lowering the temperature of the supply air stream.

This loop is interlocked by fan operation. When the fan is not running and a low temperature condition is not sensed, the PLC program places the PI controller in manual mode and sets the PI controller output to zero to shut the valve fully.

When a low temperature (34 °F) is sensed by the freezestat temperature switch (TS-XX), the switch indicates the condition to the PLC. The PLC program removes power from the solenoid valve in the air supply line to the control valve, causing the solenoid to vent air away from the valve actuator and positioner which makes the fail-open control valve fully open. The PLC program also places the PI controller into manual mode and sets the PI controller output to zero. When the low temperature switch closes and the fan is restarted, the PLC program provides power to the solenoid valve and places the PI controller into automatic mode.

#### Conditions to start equipment

None.

#### Conditions to stop equipment

None.

### B.5.2 Loop TS-XX

#### Hard-wired Functions

When a low temperature (34 °F) is sensed by the freezestat temperature switch (TS-XX), the switch removes power from the fan motor (interlocks the MCC control wiring).

#### Control Details

The freezestat temperature switch (TS-XX) provides a discrete input into the controller. When the switch opens, the controller sets its output to zero, opening the temperature valve. The freezestat must be manually reset.

#### Conditions to start equipment

Temperature is above TS-XX setpoint and TS-XX is reset.

**Conditions to stop equipment**

TS-XX is open.

## B.6 Reference Documents

P&ID #: P-F-X
Schematic: S-F-X

## B.7 Troubleshooting

For a High or Low Temperature Alarm (TAH/L-XX), check:

- Fan operation
- Current switch
- MCC control switch
- PLC fuses
- Wiring connections
- I/P converter
- Air set
- Freezestat

Priority of Alarm Notification to System Owner:
Medium (If problem can not be resolved, inform system owner within 8 hours.)

# Appendix C

# Creative Commons Attribution License

## C.1 A simple explanation of your rights

This is an "open-source" textbook, meaning the digital files used to create the final form (PDF, printed paper, or other) are freely available for your perusal, reproduction, distribution, and even modification.

## C.2 Legal code

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

**1. Definitions**

1. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this

License.

2. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

3. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

4. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

5. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

6. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

7. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

8. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

9. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

**2. Fair Dealing Rights.** Nothing in this License is intended to reduce, limit, or restrict any

uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

**3. License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
2. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
3. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
4. to Distribute and Publicly Perform Adaptations.
5. For the avoidance of doubt:
   1. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
   2. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
   3. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

**4. Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

1. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms

of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

2. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

3. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

**5. Representations, Warranties and Disclaimer**

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

**6. Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

**7. Termination**

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

**8. Miscellaneous**

1. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

2. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

6. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted

under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

**Creative Commons Notice**

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at `http://creativecommons.org/`