18 Computer Control Concepts

26 Typical forms of computer control system

DDC, Direct Digital Control
SCC, Supervisory Computer Control
   SCC + Analog regulator
   SCC + DDC
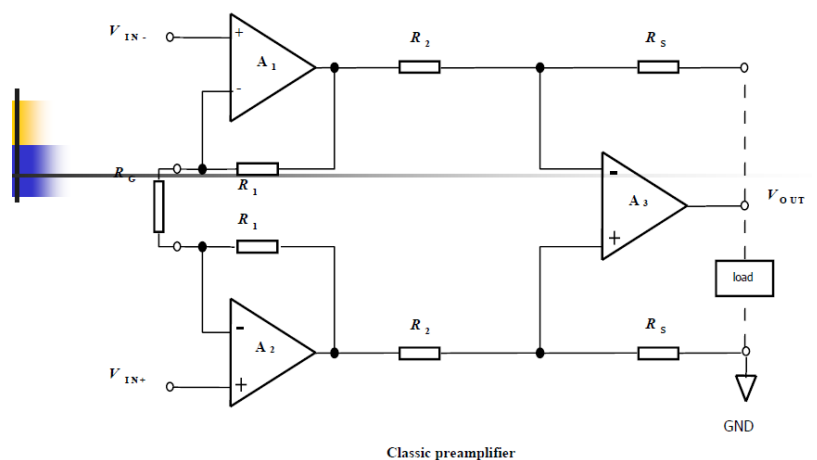DCS, Distributed Control System
FCS, Fieldbus Control System

63 Operational amplifiers

The current signal between V + and V- points is identically zero, that is, the input impedance is infinite.

The output voltage of the amplifier is constant for a certain value, that is, the output impedance is zero.

Infinite open-loop gain

Classic preamplifier

RG external resistor, dedicated to adjust the gain of the amplifier. Therefore, the gain G of the amplifier has a close relationship with the external resistor RG. Gain formula

$$G = \frac{V_{\text{OUT}}}{V_{\text{IN+}} - V_{\text{IN-}}} = \frac{R_S}{R_2}(1 + \frac{2R_1}{R_G})$$
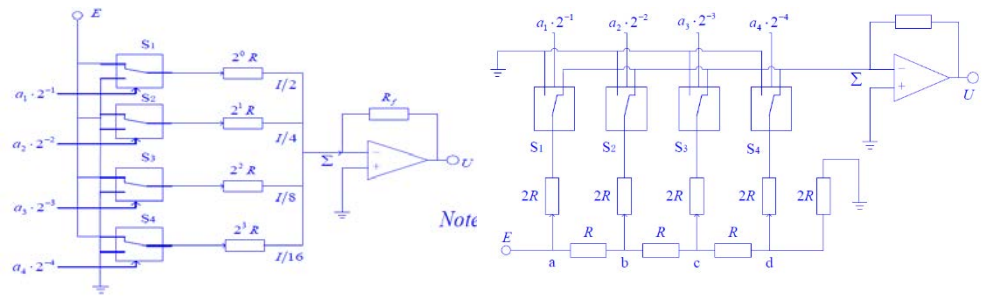
DAC

49 Conversion process

Sampling, Quantity, Coding

85 decoding networks

Binary weighted resistor network
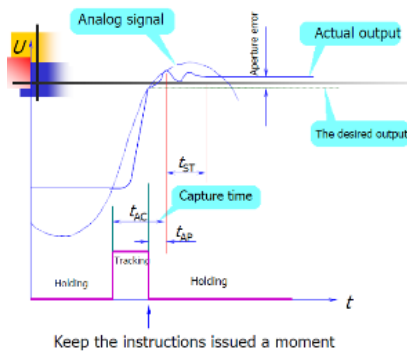
T-type resistor network

111 Technical Indicators

Resolution: $\frac{1}{2^n} * 100\%$

Settling time

Output level

ADC

68 ADC Sample and Hold

Holder

   general structure

   tandem-type sample & hold

Switch

   The most common controllable switch is a relay

   CMOS analog switch is a controllable switch

Two of the most important components
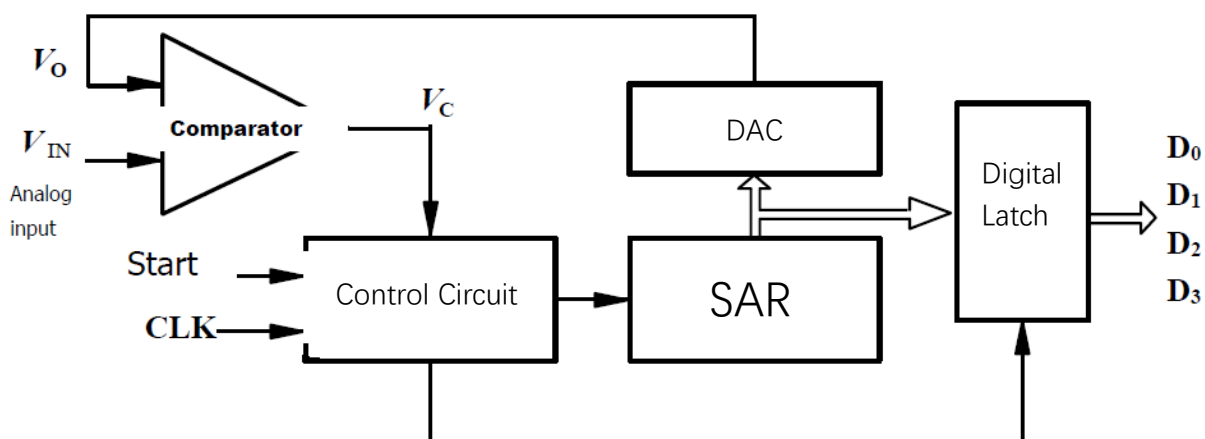
   Sample and holder

   Comparator

96 Type

Successive Approximation Register (SAR)

Dual-slope Integration
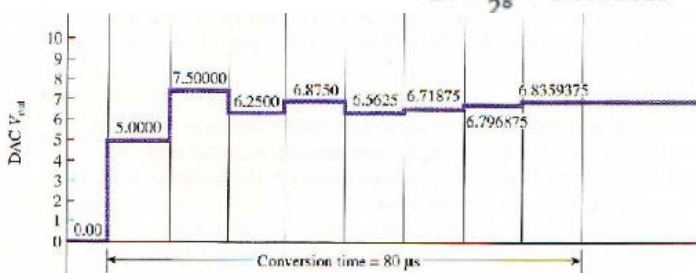
Voltage / Frequency Type

General structure

The structure of the tandem-type sample & hold

Example   Vin = 6.84V   8 bit $\Delta V = \frac{10}{2^8} = 0.0390625$

- so $(37.375)_{10} = (100101.011)_2$

Conversion time = 80 μs

2

108 Technical Indicators

Resolution: $\frac{1}{2^n} * 100\%$

Range: voltage

Accuracy: $\pm \frac{V}{2^{n+1}}$ (absolute) $\frac{1}{2^{n+1}} * 100\%$ (relative)

Conversion time
Output logic levels
Working temperature
Reference power supply requirements

Interfaces
155 Output Mode
Unipolar
Bipolar
160 DAC0832 & Interfaces with MCS-51
Direct mode
Single buffer mode
Double buffering mode

Process Channel

175 Opto-Coupler Isolation: most important anti-jamming technology with hardware in computer control system
Triode type
One-way thyristor type
TRIAC type
179 Digital input channels
182 Digital output channels
Transistor drive circuit
Thyristor drive circuit
Relay drive circuit —→
Solid-state relay drive circuit

Interference
196 Source
External
Internal
198 Way of transmission
Electrostatic coupling
Electromagnetic coupling
Common impedance coupling
204 How it works
Series mode interference: superimposed on the measured signal interference
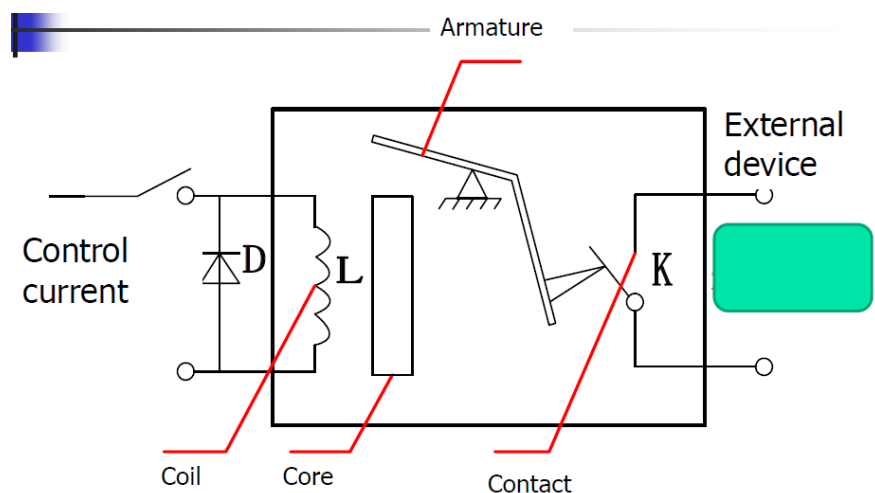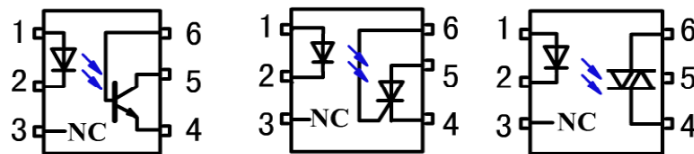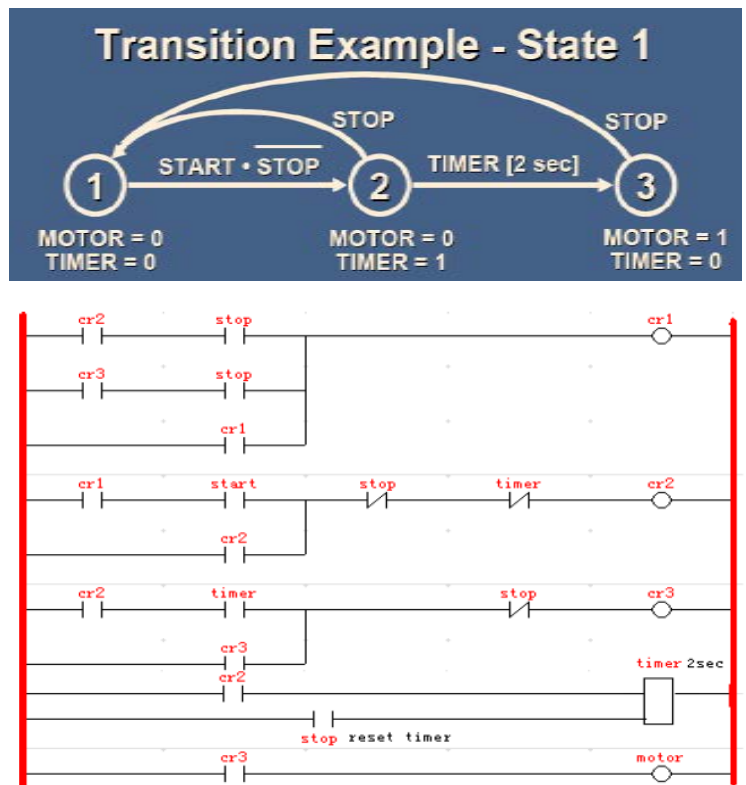Common mode interference: added in the gauge interference between any input and ground
Long-term transmission interference: wave reflection when high-frequency signals are transmitted in long wires
209 Anti methods
Hardware anti-interference
Software anti-interference

252 PLC

WD

8 Features of a real-time system
Hardware/software system
Data reception, data processing, data delivery to the process within a given time interval
External events
Processing priority

9 Requirements on Real-Time Systems
Timeliness: reaction right on time
Simultaneousness: simultaneous reaction to various events
Dependability: reliable, safe, available
Predictability: all reactions must be predictable and deterministic

17 Product Automation System Structure Simple →
25 Time Requirements on the different levels in ISA-95 Standard

Effects ranging between months-years

Effects ranging between days-weeks-months

Effects ranging between minutes-hours

microseconds-seconds
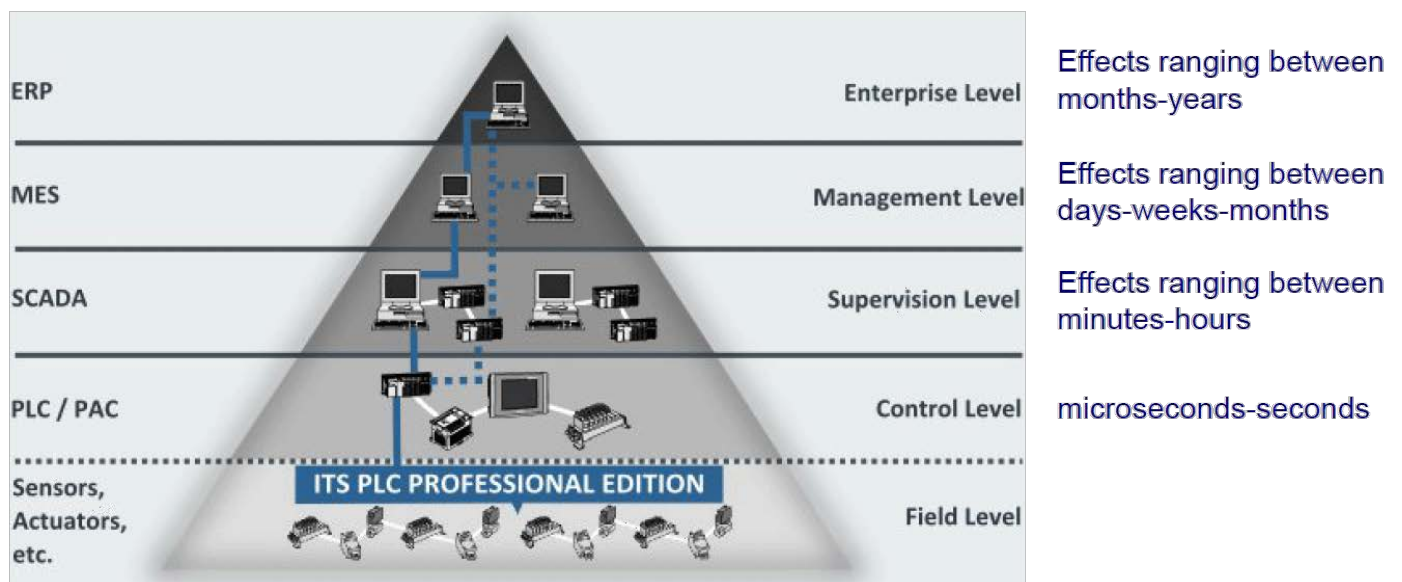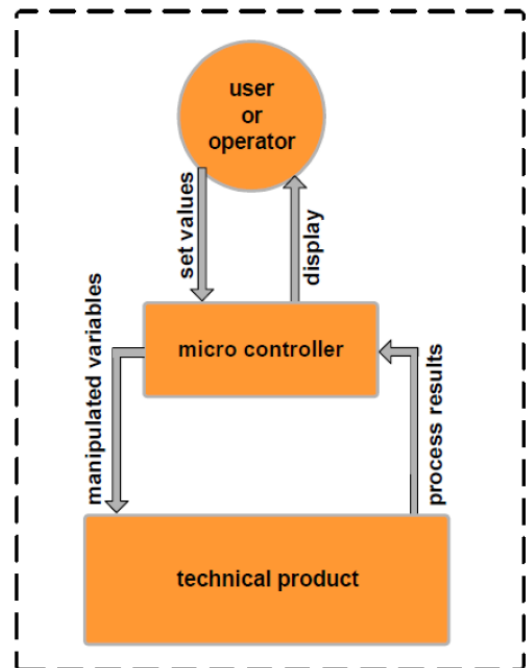
29 Cyclic Execution of PLCs
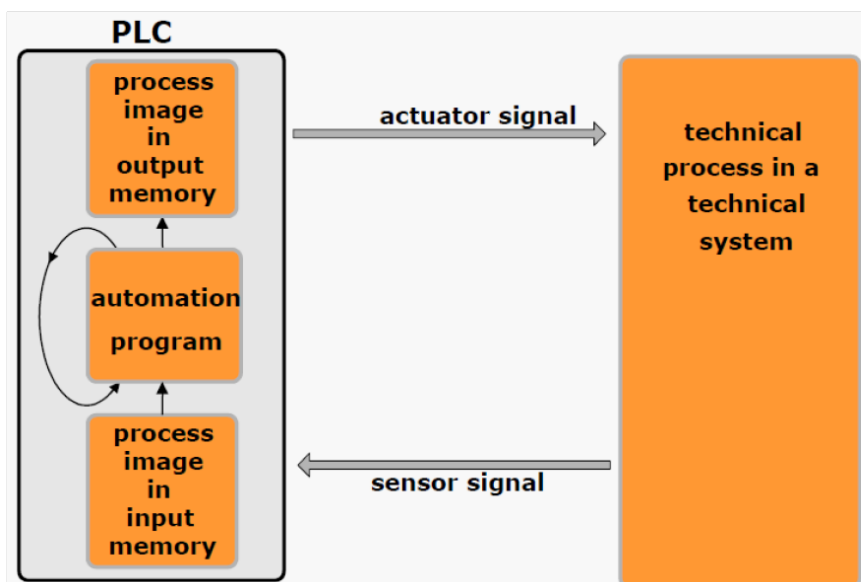
Advantage: simple programming through cyclic operational mode

Disadvantage: maximum reaction time on events in the technical process is equal to two program cycles

Program execution time:
– cycle time is not constant
– 1 ms per 1000 instructions

34 Microcontroller Components

CPU = Central Processing Unit micro processor

RAM = Random Access Memory data memory

EPROM/PROM/ROM = Erasable/ Programmable/ Read Only Memory fixed memory

I/O = parallel or serial input/output components process and data periphery

Counter/Timer = generation of clock pulses

Interrupt Controller = handling of hardware interrupts

38 System Communication

Operation level: large data, short reaction time

Process level: middle data, middle reaction time

Field level: small data, long reaction time

a) star topology b) ring topology c) net topology d) bus topology

CAU = centralized automation unit
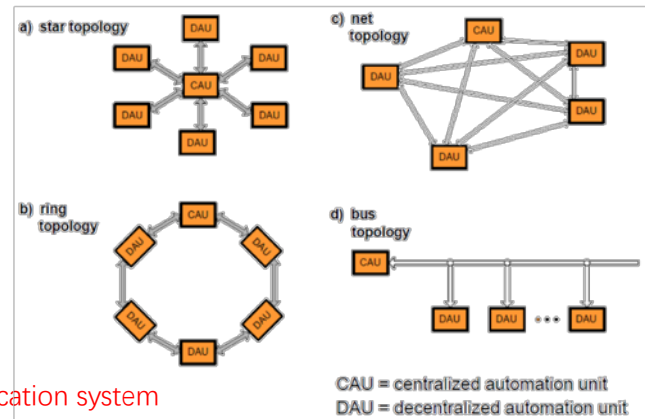DAU = decentralized automation unit

44 Communication in Distributed Automation Systems

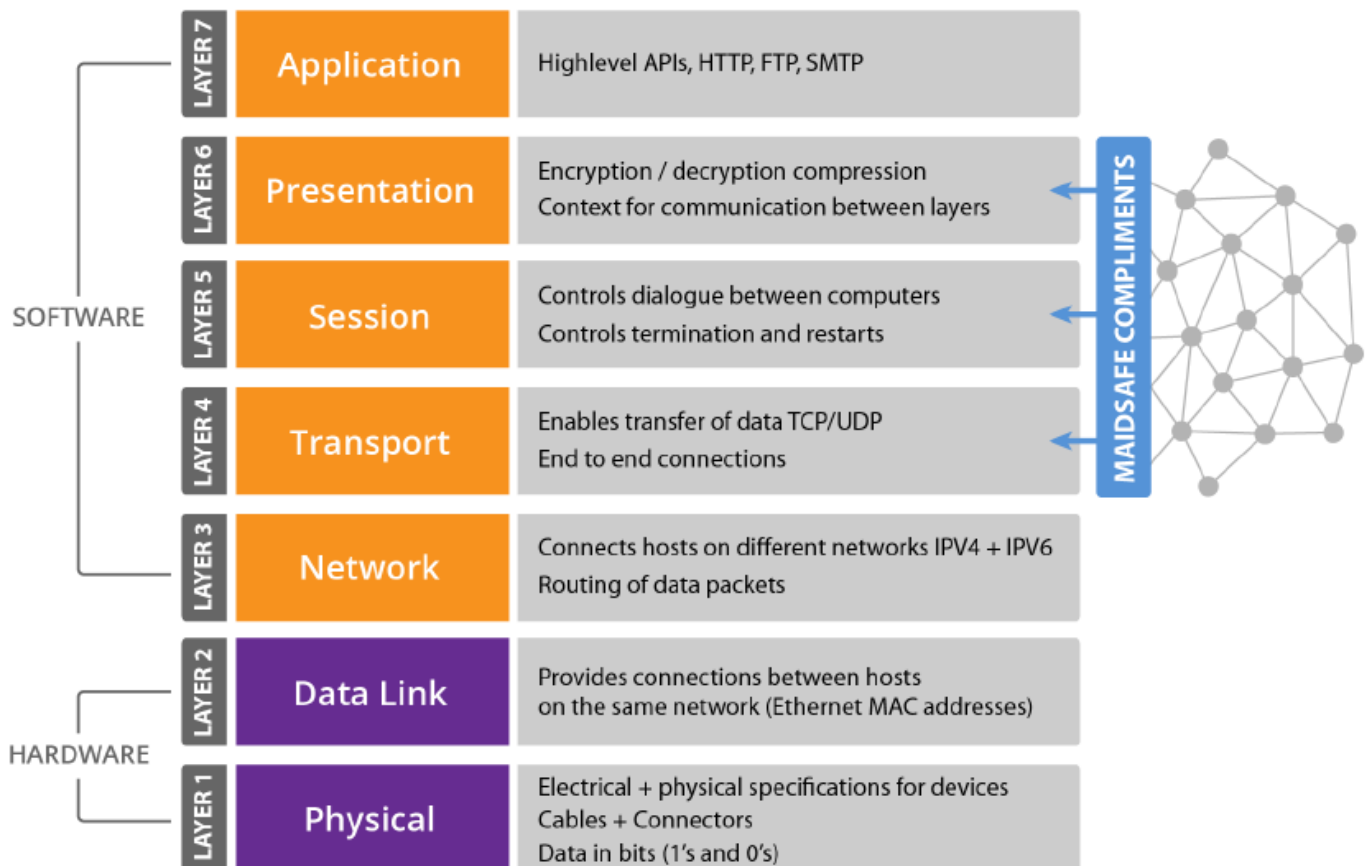Star: failure of the central unit causes failure of the communication system

Ring: each unit can only transmit messages to its direct neighbors

Net: parallel information transmission, short reaction time, many interfaces, high cabling costs

Bus: only one participant at a time is able to send, simultaneous information reception from all participants

45 Open System Interconnection (OSI)

	Layer		Description
SOFTWARE	LAYER 7	Application	Highlevel APIs, HTTP, FTP, SMTP
	LAYER 6	Presentation	Encryption / decryption compression — Context for communication between layers
	LAYER 5	Session	Controls dialogue between computers — Controls termination and restarts
	LAYER 4	Transport	Enables transfer of data TCP/UDP — End to end connections
	LAYER 3	Network	Connects hosts on different networks IPV4 + IPV6 — Routing of data packets
HARDWARE	LAYER 2	Data Link	Provides connections between hosts on the same network (Ethernet MAC addresses)
	LAYER 1	Physical	Electrical + physical specifications for devices — Cables + Connectors — Data in bits (1's and 0's)

MAIDSAFE COMPLIMENTS

46 Redundancy

Hardware redundancy
  Goal: Detection of hardware failures
  Operation principle: m-of-n-redundancy majority ruling
  Effect: no faults, until multiple defects occur
Software redundancy
  Objective: detection of errors in software
  Starting point: software has errors
  Redundant software alternatives are executed one by one and are compared with a voter (× real-time systems)
Measured value redundancy
  redundant measured value
  dependent measured value
Time redundancy
  multiple inquiry of the same measured value in certain intervals
Levels of fault-tolerance
  complete fault-tolerance
  reduced operational capacity
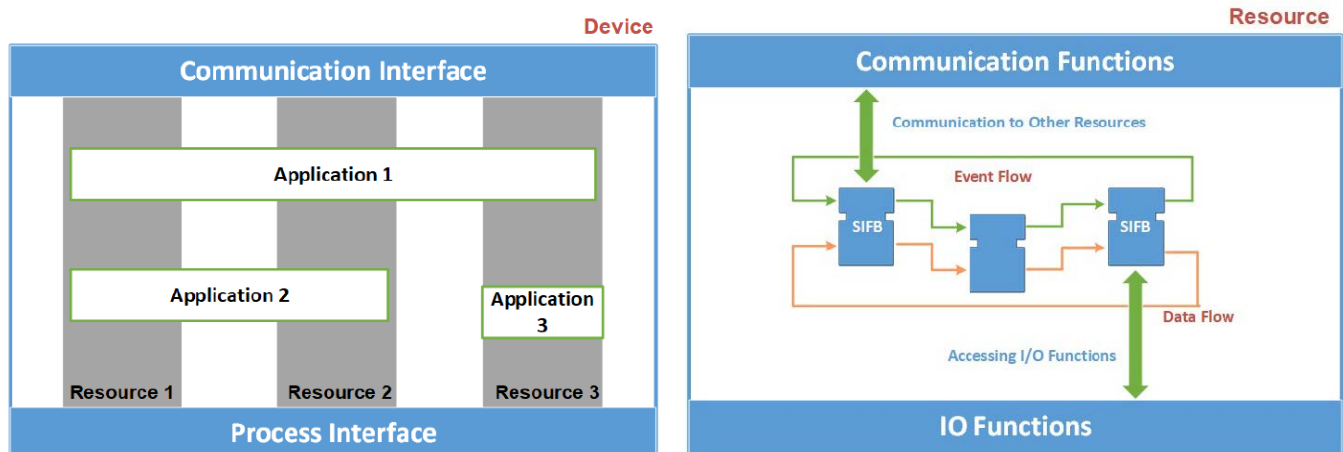 transition to a safe state

57 The IEC 61499 Standard Goals

Reusability
Portability
Reconfigurability
Interoperability

61 IEC 61499 Device / Resource Models

65 IEC 61499 Function Block Models

Atomic function block model
  Basic function block (BFB)
  Service interface function block (SIFB)
Composite function block model
 Composite function block
  Sub-application

69 IEC 61499 Execution Control Chart

In an ECC, there will be one <u>initial state (START)</u> with some <u>EC states</u>. Switching between EC states is called <u>EC transition</u>. An EC transition consists of two parts: <u>event and guard condition (must at least have one)</u>. When the EC transition condition is true, the ECC will jump to the next state, execute the algorithm and emit event output. If more than one EC transition condition is satisfied, the ECC will jump to the <u>highest priority EC state</u> labelled with {priority}.

The execution semantics of the ECC are defined as:

- The ECC will be executed once when one input event is activated.
- Both the event and the guard condition must be satisfied before an EC state can jump to another EC state.
- Once the ECC enters the new state, it will execute the algorithm, emit the output event.
- Then the ECC will check all EC transitions from the new EC state. If any EC transition condition is met, the ECC will jump to that state. As there is no new incoming event during this time, EC transitions with guard condition only will be considered.
- If the event condition is not set in the EC transition condition, this means any event will satisfy the requirement.
- If the EC transition condition is set to 1, this means unconditionally true.
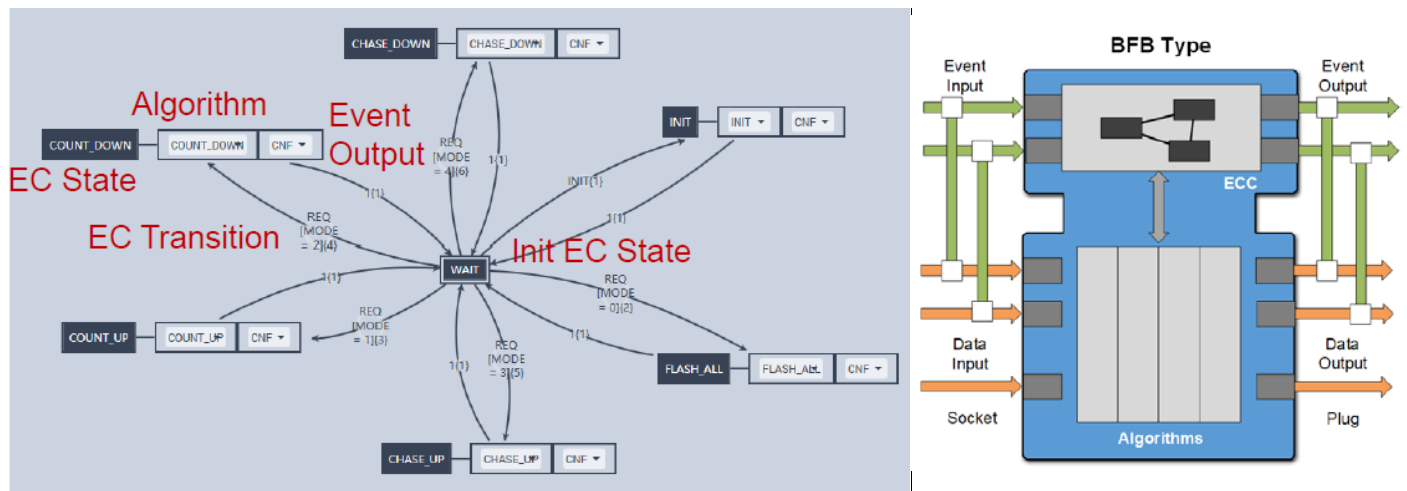
80 IEC 61499 Execution Models

Cyclic
Buffered Sequential
Non-preempted Multi-threading
Synchronous
Time-stamped Discrete-event
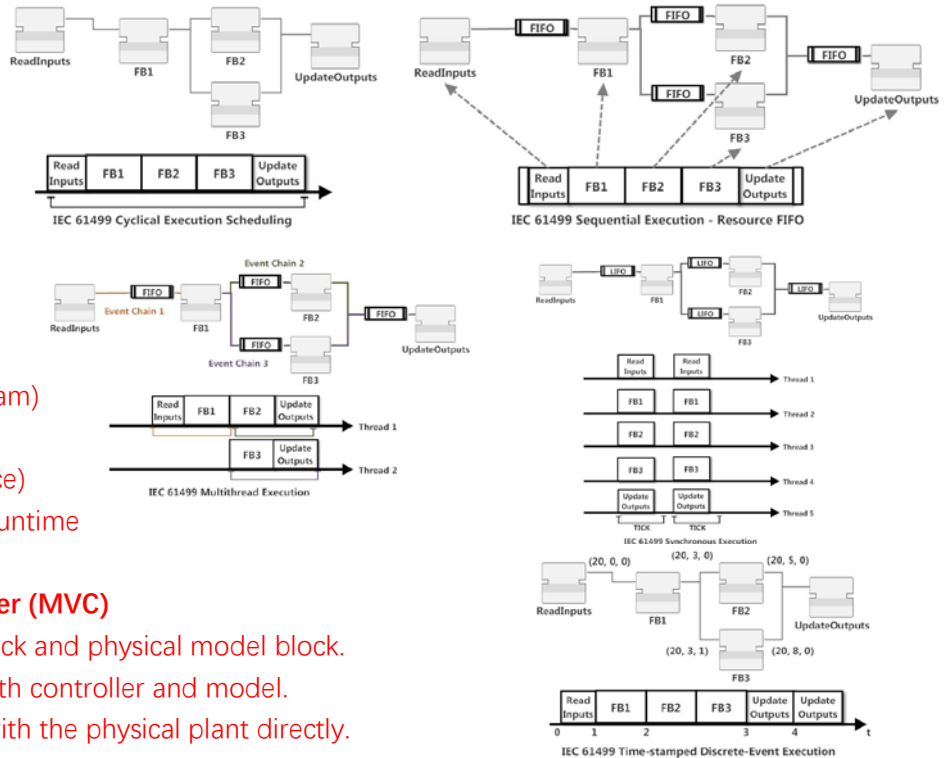
85 Design IEC 61499

Design the FB Interface
Create Algorithms (PLC Ladder Diagram)
Design ECC (Execution Control Chart)
Design HMI (Human Machine Interface)
Put All FBs Together and Deploy to Runtime

88 IEC 61499 Model-View-Controller (MVC)

A closed-loop between controller block and physical model block.
A view block could be attached to both controller and model.
The model block could be replaced with the physical plant directly.
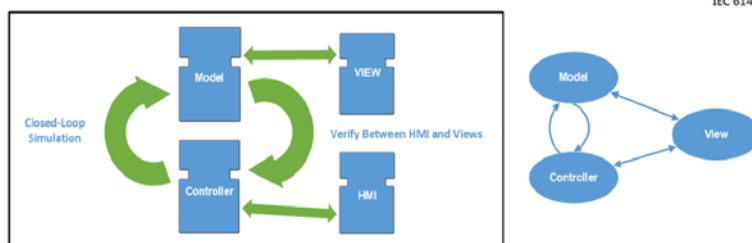
96 IEC 61131-3 language
Ladder Diagram (LD)
Structured Text (ST)
Instruction List (IL)
Function Block Diagram (FBD)
Sequential Function Chart (SFC)

99 Software Model in IEC 61131-3 →
100 IEC 61131-3 Tasks
Class
  Continuous Tasks
  Periodic Tasks
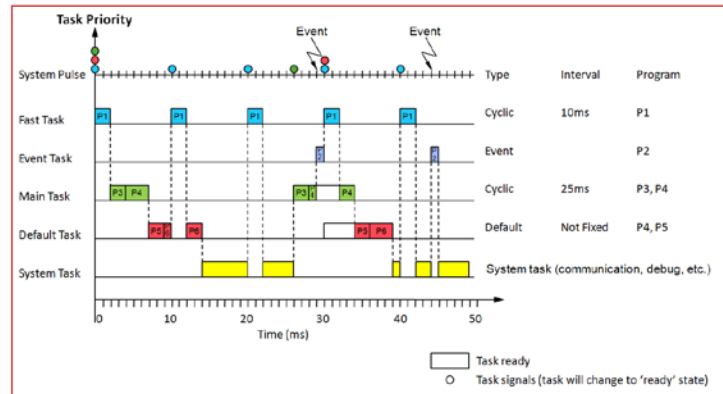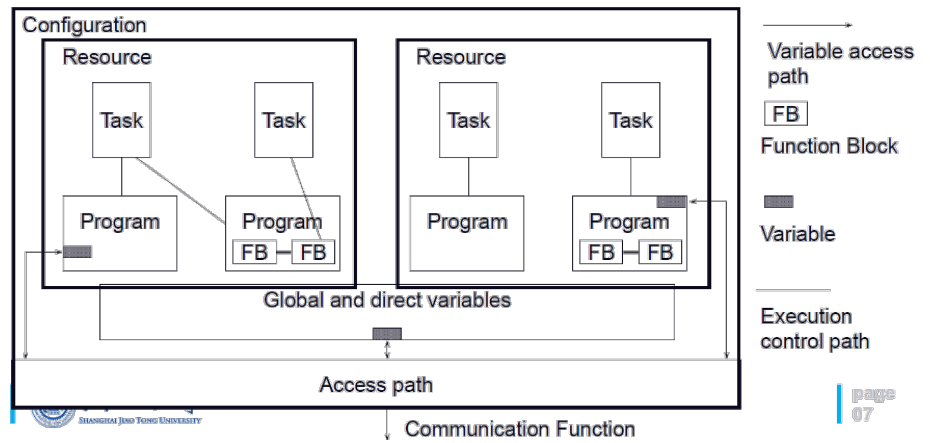 Interrupt Tasks
Priority
  System Pulse
  Fast Task
  Event Task
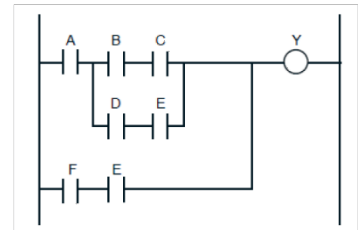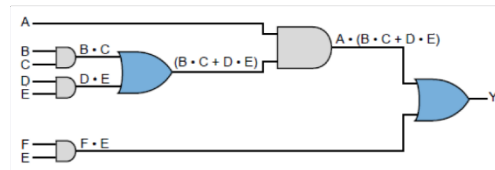 Main Task
  Default Task
  System Task

101 IEC 61131-3 Programming Organization Units
Function
Function Block
Program

which can be minimized, using Boolean algebra's distributed rule, to
$$Y = A \cdot (B \cdot C + D \cdot E) + (F \cdot E)$$

111 Ladder Diagram Logic →
112 Ladder Relay Instructions

Ladder Relay Instructions
(Purpose: To provide hardwired relay capabilities in a PLC)

Instruction	Symbol	Function
Examine-ON/Normally Open	⊣ ⊢	Tests for an ON condition in a reference address
Examine-OFF/Normally Closed	⊣/⊢	Tests for an OFF condition in a reference address
Output Coil	─( )─	Turns real or internal outputs ON when logic is 1
NOT Output Coil	─(∅)─	Turns real or internal outputs OFF when logic is 1
Latch Output Coil	─(L)─	Keeps an output ON once it is energized
Unlatch Output Coil	─(U)─	Resets a latched output
One-Shot Output	─(OS)─	Energizes an output for one scan or less
Transitional Contact	⊣↑⊢	Closes for one scan when its trigger contact makes a positive transition

Timer Instructions
(Purpose: To provide hardware timer capabilities in a PLC)

Instruction	Symbol	Function
ON-Delay Energize Timer	TON ─( )─	Energizes an output after a set time period when logic 1 exists
ON-Delay De-energize Timer	TON ─( )─	De-energizes an output after a set time period when logic 1 exists
OFF-Delay Energize Timer	TOF ─( )─	Energizes an ouput after a set time period when logic 0 exists
OFF-Delay De-energize Timer	TOF ─( )─	De-energizes an output after a set time period when logic 0 exists
Retentive ON-Delay Timer	RTO ─( )─	Energizes an output after a set time period when logic 1 exists and then retains the accumulated value
Retentive Timer Reset	RTR ─( )─	Resets the accumulated value of a retentive timer
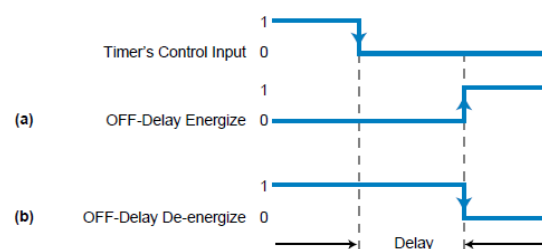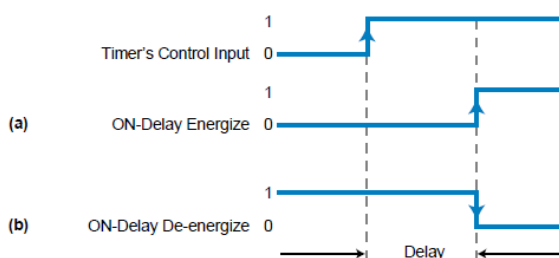
159 SCADA Key Features

Input / Output task: interface between the control and monitoring system and the plant floor
Alarm task: detecting digital alarm points and compare the values of analog alarm points to alarm thresholds
Trends task: collects data to be monitored over time
Reports task: periodic, event triggered or activated by the operator and acquired from plant data
Display task: manages all data to be monitored by the operator and all control actions requested by the operator

161 SCADA Real-Time Constraints

Display: Display of value from RTU (1 to 2 seconds maximum)
Display: Display of entire new display (1 second)
Trend: Retrieval of historical trend and display (2 seconds)
Alarm: Acknowledge of alarm on operator screen (1 second)
RTU: Control request from operator to RTU (1 second critical; 3 seconds noncritical)
RTU: Sequence of events logging at RTU of critical events (1 millisecond)

165 Bus Access Methods

Deterministic bus access
    determined transmission rights
    predictable response time
    permanent monitoring of the bus
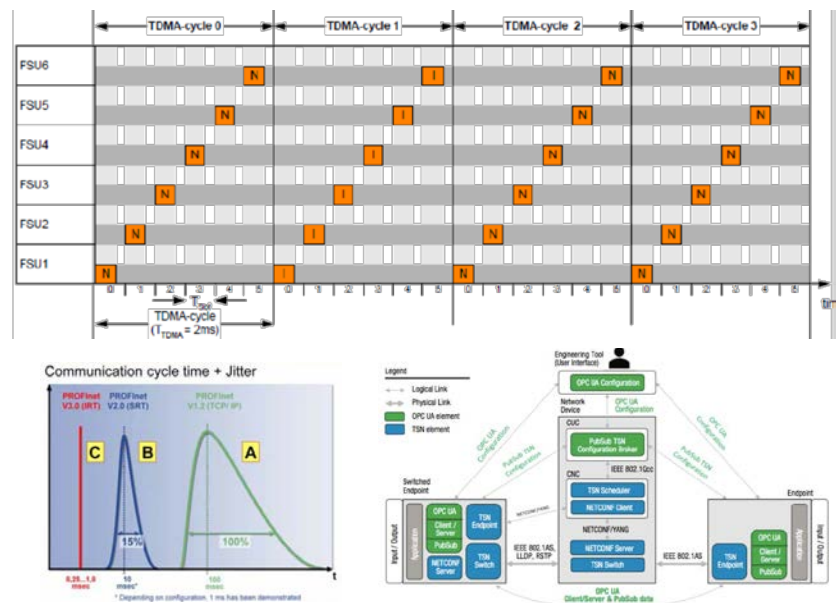   event driven communication
    low/middle bus load
    E.g. master/slave, Token-Passing, TDMA
Random bus access
    random transmission rights
    non-predictable response time
    E.g. CSMA (CSMA/CD, CSMA/CA)

168 TDMA →

Procedure: Within a period (TDMA-cycle) each participant is assigned one or several time slots of determined length.
Principle: distributed shift registers
Advantages: short and constant cycle time, low protocol overhead
Disadvantages: √ synchronization of participants, × autonomous participants, low flexibility, no dynamic adaptation

169 CSMA

Procedure: Each participant checks if bus is free/occupied (carrier sense). If bus is free, it starts transmission. In case of collisions, try transmission once again.
Principle: each participant has bus access without explicit broadcasting right dispatching. (Multiple Access)
CSMA-CD (Collision Detection, Recognition of collisions through data comparison, Transmission repetition after participant specific waiting period)
CSMA-CA (Collision Avoidance, Avoidance of collisions through priority rules)

172 CAN (Controller Area Network) Bus

Message oriented addressing   Multi-master bus access method
In case of access conflict: bus arbitration according to priority like CSMA/CA
Short length of messages (0...8 byte)   Fast Transmission (1Mbit/s)
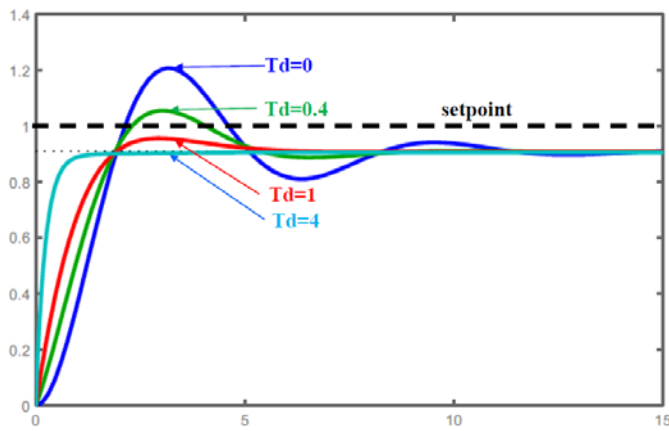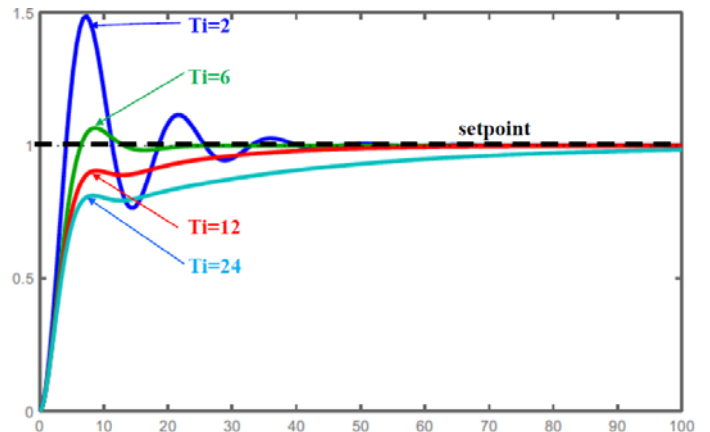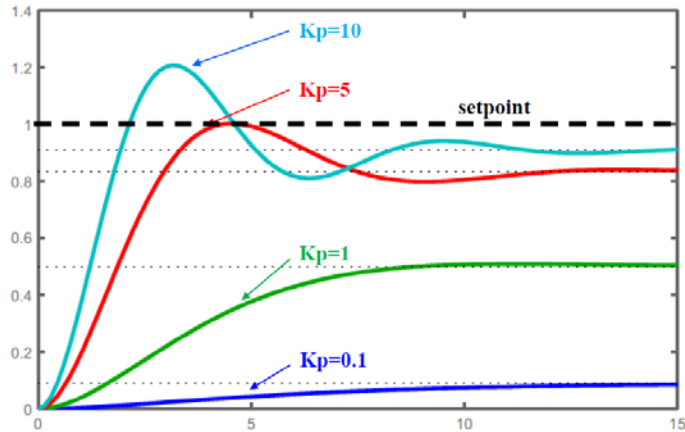Various error recognition mechanisms   Self-test through error counter
ProfiNet↑, Ethernet/IP, Ethernet Powerlink, Modbus TCP, OPC UA, OPC UA TSN↑

L6 PID Control
Analog PID
23 Parameters

EFFECTS OF INDEPENDENT P, I, AND D TUNING

Closed-Loop Response	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

Digital PID
34 Forms

$$u_k = K_{\mathrm{p}} \left(e_k + \frac{T}{T_{\mathrm{i}}} \sum_{j=1}^{k} e_j + \frac{T_{\mathrm{d}}}{T}(e_k - e_{k-1}) \right)$$

$$u_k = P_{\mathrm{p}}(k) + P_{\mathrm{i}}(k) + P_{\mathrm{d}}(k)$$

$$\Delta u_k = K_{\mathrm{p}} \left(e_k - e_{k-1} + \frac{T}{T_{\mathrm{i}}}e_k + \frac{T_{\mathrm{d}}}{T}(e_k - 2e_{k-1} + e_{k-2}) \right)$$
$$= d_0 e_k + d_1 e_{k-1} + d_2 e_{k-2}$$

$$d_0 = K_{\mathrm{p}} \left(1 + \frac{T}{T_{\mathrm{i}}} + \frac{T_{\mathrm{d}}}{T} \right), \ d_1 = -K_{\mathrm{p}} \left(1 + 2\frac{T_{\mathrm{d}}}{T} \right), \ d_2 = K_{\mathrm{p}} \frac{T_{\mathrm{d}}}{T}$$

$$P_{\mathrm{p}}(k) = K_{\mathrm{p}} e_k$$
$$P_{\mathrm{i}}(k) = K_{\mathrm{p}} \frac{T}{T_{\mathrm{i}}} \sum_{j=1}^{k} e_j = P_{\mathrm{i}}(k-1) + K_{\mathrm{p}} \frac{T}{T_{\mathrm{i}}} e_k$$
$$P_{\mathrm{d}}(k) = K_{\mathrm{p}} \frac{T_{\mathrm{d}}}{T}(e_k - e_{k-1})$$

Practical Issues Related to Implementation
    37 Selecting the sampling period

        Sample rule modification: $T \leq \frac{1}{Mf_{max}}$ $M = 5 \sim 10$

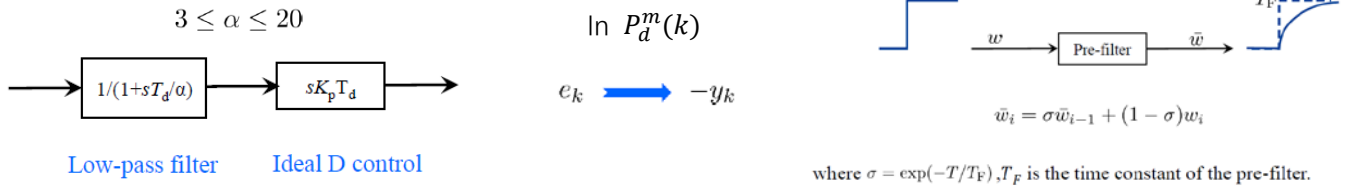   In control algorithm: $\frac{t_s}{15} \leq T \leq \frac{t_s}{6}$ t_s is settling time

Setpoint weighting: $w_k \rightarrow \beta w_k \quad 0 \le \beta \le 1$

Filtering the D action

Suppressing large changes resulting from setpoint changes

Setpoint weighting

$$3 \le \alpha \le 20 \qquad\qquad \text{In } P_d^m(k)$$

Low-pass filter   Ideal D control

$$e_k \implies -y_k$$

$\bar{w}_i = \sigma \bar{w}_{i-1} + (1-\sigma)w_i$

where $\sigma = \exp(-T/T_F)$, T_F is the time constant of the pre-filter.

- **Modified digital PID controller**

$$u_k = P_p^m(k) + P_i(k) + P_d^m(k)$$

$P_d^m(k) = K_p(\beta(\sigma\bar{w}_{k-1} + (1-\sigma)w_k) - y_k)$	**(SW)** for P control to reduce the maximum overshooting of the process output and D control
$P_i(k) = K_p\dfrac{T}{T_i}\displaystyle\sum_{j=1}^{k} e_j = P_i(k-1) + K_p\dfrac{T}{T_i}e_k$	No change
$P_d^m(k) = \dfrac{T_d}{T_d + \alpha T}P_d^m(k-1) + \dfrac{K_p T_d \alpha}{T_d + \alpha T}[-y_k + y_{k-1}]$	**(F)** to limit the amplification of high-frequency noise and **(S)** to damp the response to the setpoint change

Switch off the I action when the control error is large

$$u_k = P_p(k) + \chi_{e_k}P_i(k) + P_d(k)$$

$$u_k = P_p(k) + P_i(k) + P_d(k) \qquad P_i(k) = P_i(k-1) + K_p\frac{T}{T_i}e_k\chi_{e_k}$$

$$\chi_{e_k} = \begin{cases} 0, & \text{if } |e_k| > \epsilon \\ 1, & \text{if } |e_k| \le \epsilon \end{cases}$$

where

$$\chi_{e_k} = \begin{cases} 0, & \text{if } u_{k-1} > u_{max}\&e_k \ge 0 \ \text{ or } \ u_{k-1} < u_{min}\&e_k \le 0 \\ 1, & \text{otherwise} \end{cases}$$

Stop summation if control signal saturates, control variable sign = control error sign

Re-calculate the error

If the actuator saturates at k, i.e.,

$$u_k > u_{max} \ \text{ or } \ u_k < u_{min}$$

then calculate the effective error

$$e_k \leftarrow \frac{\frac{1}{K_p}u^* - \frac{T}{T_i}\sum_{j=1}^{k-1}e_j + \frac{T_d}{T}e_{k-1}}{1 + \frac{T}{T_i} + \frac{T_d}{T}}$$

where $u *= u_{max}$ or u_{min}.

Exponential filter

$$\bar{e}_k = \frac{\tau_F}{T + \tau_F}\bar{e}_{k-1} + \frac{T}{T + \tau_F}e_k$$

Moving average filter ($\lambda = 1$, or Introducing the forgetting factor $\lambda < 1$)

$$\bar{e}_k = \bar{e}_{k-1} + \frac{1}{N}[e_k - e_{k-N}] \quad \bigg| \quad \overline{e_k} = \lambda\overline{e_{k-1}} + \frac{1-\lambda}{1-\lambda^N}(e_k - \lambda^N e_{k-N})$$
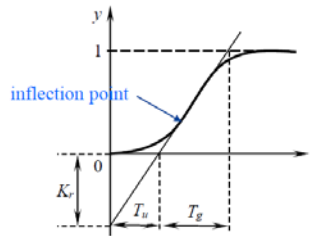
Noise-spike filter (rate-of-change filter)

$$\bar{y}_F(k) = \begin{cases} y_k, & \text{if } |y_k - \bar{y}_F(k-1)| \le \Delta \\ \bar{y}_F(k-1) - \Delta, & \text{if } y_k - \bar{y}_F(k-1) < -\Delta \\ \bar{y}_F(k-1) + \Delta, & \text{if } y_k - \bar{y}_F(k-1) > \Delta \end{cases}$$

Tuning

63 Methods
- Analytical methods
- Heuristic methods
- Frequency response methods
- Optimization methods
- Adaptive tuning methods

69 Ziegler-Nichols Step Response Method

Controller	K_p	T_i	T_d
P	$1/K_r$	–	–
PI	$0.8/K_r$	$3T_u$	–
PID	$1.2/K_r$	$2T_u$	$0.42T_u$

Controller parameters for Z-N step response method

Obtain the step response $y(t) = L^{-1}(\frac{H(s)}{s})$

Find the point with maximum gradient $\dot{y}_{max}$ in (t_0, y_0) $(\ddot{y} = 0)$

Calculate the auxiliary line $y - y_0 = \dot{y}_{max}(t - t_0)$

$let\ t = 0, we\ get\ -K_r$

$let\ y = 0, we\ get\ T_u$

s^3	1	23	
s^2	9	$15 + Kp$	auxiliary equation
s^1	$\dfrac{192 - K_p}{9}$	0	critical equation
s^0	$15 + Kp$		

72 Ziegler-Nichols Frequency Method

Obtain the critical gain K_r by Routh-Hurwitz condition in system's $den + K_p num = 0$

Solve the auxiliary formula (right above the critical formula) to get T_r

Controller	K_p	T_i	T_d
P	$0.5K_r$	–	–
PI	$0.45K_r$	$0.85T_r$	–
PID	$0.6K_r$	$0.5T_r$	$0.12T_r$

L7 Digital Controller Design
88 Z transformation

Time function $y(t)$	Time sequence $y(kT)$	z transform $Y(z)$	Laplace transform $Y(s)$
$\delta(t)$	$\begin{cases}1, & k=0 \\ 0, & k>0\end{cases}$	1	1
$1_{t\geq 0}$	$1_{k\geq 0}$	$\dfrac{1}{1-z^{-1}}$	$\dfrac{1}{s}$
t	kT	$\dfrac{Tz^{-1}}{(1-z^{-1})^2}$	$\dfrac{1}{s^2}$
$\dfrac{1}{2}t^2$	$\dfrac{1}{2}(kT)^2$	$\dfrac{T^2 z^{-1}(1+z^{-1})}{2(1-z^{-1})^3}$	$\dfrac{1}{s^3}$
e^{-at}	e^{-akT}	$\dfrac{1}{1-e^{-aT}z^{-1}}$	$\dfrac{1}{s+a}$

- Linearity: $\mathcal{Z}[af(k) + bg(k)] = aF(z) + bG(z)$
- Multiplication by a^k: $\mathcal{Z}[a^k y(k)] = Y(z/a)$
- Real translation theorem: $\mathcal{Z}[y(k-i)] = z^{-i}Y(z)$,
- Initial-value theorem:
$$y(0) = \lim_{z\to\infty} Y(z)$$
- Final-value theorem:
$$\lim_{k\to\infty} y(k) = \lim_{z\to 1}(z-1)Y(z)$$
Approximation of $s = \ln z/T$ is necessary!

$z = e^{sT} \approx 1 + sT$		$s \approx \dfrac{z-1}{T}$
$z^{-1} = e^{-sT} \approx 1 - sT$	Taylor expansion →	$s \approx \dfrac{z-1}{Tz}$
$z = e^{sT} = \dfrac{e^{sT/2}}{e^{-sT/2}} \approx \dfrac{1+sT/2}{1-sT/2}$		$s \approx \dfrac{2}{T}\dfrac{z-1}{z+1}$

Indirect approach (Design in continuous domain, discretize controllers to obtain discrete-time destiny)

97 Emulation (Laplace domain) ↑

$$S = \frac{\ln z}{T}$$

99 Discretization (Time domain)

Proportional sampling $e_k \triangleq e(kT)$
$e(t)$ e_k

integral backward/forward rectangle, trapezoidal
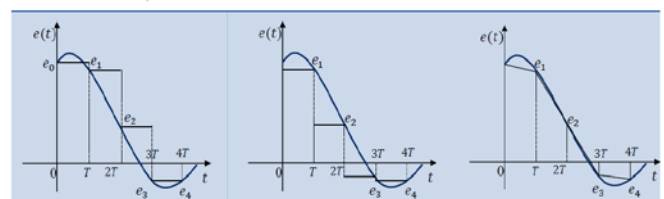
$\int_0^t e(\tau)d\tau$ →

derivative backward/forward difference

$\dfrac{de(t)}{dt}$ $\dfrac{e_k - e_{k-1}}{T}, \dfrac{e_{k+1} - e_k}{T}$

Forward backward rectangular method, trapezoidal method for integral approximation

$$\int_0^t e(\tau)d\tau \approx T\sum_{j=1}^{k} e_{j-1} \qquad \int_0^t e(\tau)d\tau \approx T\sum_{j=1}^{k} e_j \qquad \int_0^t e(\tau)d\tau \approx T\sum_{j=1}^{k} \frac{e_j + e_{j-1}}{2}$$
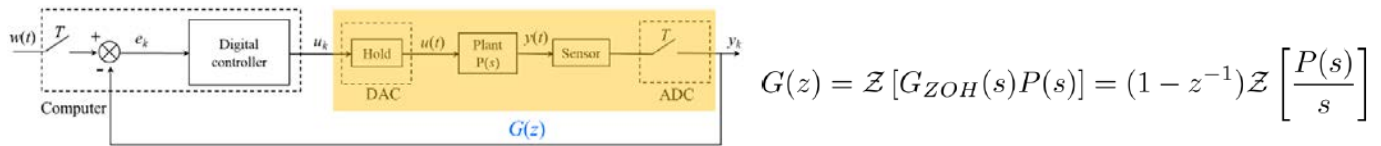
$$s \approx \frac{z-1}{T} \qquad\qquad s \approx \frac{z-1}{Tz} \qquad\qquad s \approx \frac{2}{T}\frac{z-1}{z+1}$$

The trapezoidal method is also called the Bilinear / Tustin transformation.

Direct approach (Design in discrete domain, <u>ZOH</u> discretize plant model to obtain design foundation)

$$G(z) = \mathcal{Z}[G_{ZOH}(s)P(s)] = (1 - z^{-1})\mathcal{Z}\left[\frac{P(s)}{s}\right]$$

$$D(z) = \frac{U(z)}{E(z)} = \frac{a_0 + a_1 z^{-1} + \cdots + a_m z^{-m}}{b_0 + b_1 z^{-1} + \cdots + b_l z^{-l}}$$ The digital controller $D(z)$ is physically realizable **if and only if maxPowerU(z) ≤ maxPowerE(z)**

$$b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_l u_{k-l} = a_0 e_k + a_1 e_{k-1} + \cdots + a_m e_{k-m}$$

➡ $$u_k = \frac{1}{b_0}\left[\sum_{i=0}^{m} a_i e_{k-i} - \sum_{j=1}^{l} b_j u_{k-j}\right]$$

Needs much memory to store all the previous data $O(m + l)$.

Define the intermediate variable $C(z) = \frac{E(z)}{\sum_{j=0}^{l} b_j z^{-j}}$, then $U(z) = C(z)\sum_{i=0}^{m} a_i z^{-i}$

$$\begin{cases} c_k = \frac{1}{b_0}\left[e_k - \sum_{i=1}^{l} b_i c_{k-i}\right] \\ u_k = \sum_{i=0}^{m} a_i c_{k-i} \end{cases}$$

Memory size is reduced to $O(\max\{m, l\})$.

110 Time Domain

Consider M time steps <u>into the future</u> and collect the control errors e_k and control actions into u_k

$$\{e_0, e_1, \ldots, e_M\}, \quad \{u_0, u_1, \ldots, u_M\} \qquad \min \quad J = \sum_{k=0}^{M}\left[e_k^2 + \alpha(u_k - u_\infty)^2\right]$$

Define the objective function J to measure the derivation of e_k from zero and u_k from the final value u_∞

Solve the minimization by exhaustive search, gradient descent, evolutionary algorithm …, to obtain {a_i, b_j} offline

If we are only interested in the steady-state error subject to unit-step, we get $\sum_{j=0}^{l} b_j = 0$ ↓

Recalling that $b_0 \neq 0$ (physical realizability), the simplest form is $b_0 = 1$, $b_1 = -1$

We can recover the ideal PI or PID by choosing appropriate a_0, a_1, a_2.

Final-value theorem to find the steady-state error

$$e(\infty) = \lim_{z \to 1}(z - 1)E(z) = \lim_{z \to 1}\frac{1}{1 + D(z)G(z)}$$

In order to have $e(\infty) = 0$, it is necessary that $D(1) = \infty$, i.e., $D(z)$ has a pole at $z = 1$. This means that

Minimal Prototype Control (synthesis in z plane)

117 Raw

We mainly focus on the cases when the reference input is a step, a ramp, or an acceleration function. z transforms of such time-domain polynomial inputs can be written as

$$W(z) = \frac{A(z)}{(1 - z^{-1})^q}$$

→ q is the den power of input

$$G_w(z) = 1 - (1 - z^{-1})^q(1 + f_1 z^{-1} + \cdots + f_p z^{-p})$$

$F(z) = 1$, we obtain the **minimal prototype controller**.

$$D(z) = \frac{1}{G(z)}\frac{G_w(z)}{1 - G_w(z)}$$

$$E(z) = [1 - G_w(z)]W(z) = \sum_{k=0}^{\infty} e(kT)z^{-k}$$

$e(0) \quad e(T) \quad e(2T)$ → how many steps to follow

125 Optimization

Stability

Not only e(t), but also the control output u(t) should be stable.

Include all zeros of $G(z)$ that lie on or outside the unit circle in $G_w(z)$ as zeros

Plant model estimation deviation

Any error in the pole-zero cancellation will diverge as time elapse

Include all poles of $G(z)$ that lie on or outside the unit circle in 1-$G_w(z)$ as zeros.

Intersample ripple

The control is unchanged between two consecutive sampling points.

Include all non-zero zeros of $G(z)$ in the unit circle in $G_w(z)$ as zeros.

134 Conclusion

Include non-zero zeros and z^{-1} of $G(z)$ in $G_w(z)$ and poles on/outside of unit circle of $G(z)$ in 1-$G_w(z)$

Balance coefficient with $G_w'(z)$ and $F(z)$

13

- Consider a discrete-time $G(z)$ with l pure delays, assume it has w non-zero zeros $b_1, b_2, \ldots, b_w$, and v poles $a_1, a_2, \ldots, a_v$ lying on or outside the unit circle,

$$G(z) = z^{-l}\frac{(1 - b_1 z^{-1})\ldots(1 - b_w z^{-1})}{(1 - a_1 z^{-1})\ldots(1 - a_v z^{-1})}G'(z)$$

- The minimal prototype controller is determined by
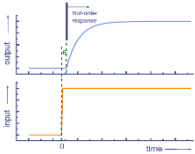
$$D(z) = \frac{1}{G(z)}\frac{G_w(z)}{1 - G_w(z)}$$

q is the power of input denominator $G'_w(z)$ and $F(z)$ are the unfixed parts with c_i and f_i carrying the other's power to balance

where

$$G_w(z) = z^{-l}(1 - b_1 z^{-1})\ldots(1 - b_w z^{-1})G'_w(z)$$
$$1 - G_w(z) = (1 - a_1 z^{-1})\ldots(1 - a_v z^{-1})(1 - z^{-1})^q F(z)$$
$$G'_w(z) = c_1 z^{-1} + c_2 z^{-2} + \cdots + c_s z^{-s}$$
$$F(z) = 1 + f_1 z^{-1} + \cdots + f_p z^{-p}$$

and $s = v + q, p = w + l$. If there are k poles at $z = 1$, it suffices to have $(1 - z^{-1})^{\max\{q,k\}}$ in $1 - G_w(z)$.

135 Dahlin's Method (synthesis in z plane)

$$G_w(z) = \frac{(1 - \sigma)z^{-l-1}}{1 - \sigma z^{-1}}$$

$$\sigma = e^{-T/\tau_c}, \quad l = \theta/T$$

$$D(z) = \frac{1}{G(z)}\frac{(1 - \sigma)z^{-l-1}}{1 - \sigma z^{-1} - (1 - \sigma)z^{-l-1}}$$

A negative <u>pole near (-1, 0)</u> on the unit circle has pronounced effect on the response, causing the controller to oscillate or ring. Dahlin suggested that ringing can be eliminated by setting $z^{-1} = 1$ in the ringing term.

$$D(z) = \frac{2.6356(1 - 0.7413z^{-1})}{(1 + 0.733z^{-1})(1 - z^{-1})(1 + 0.3935z^{-1})}$$

138 Internal Model Control

The plant model is factored as $\tilde{G}(z) = \tilde{G}_+(z)\tilde{G}_-(z)$

$\tilde{G}_+(z)$ is the bad part, containing
time-delay term z^{-l-1}
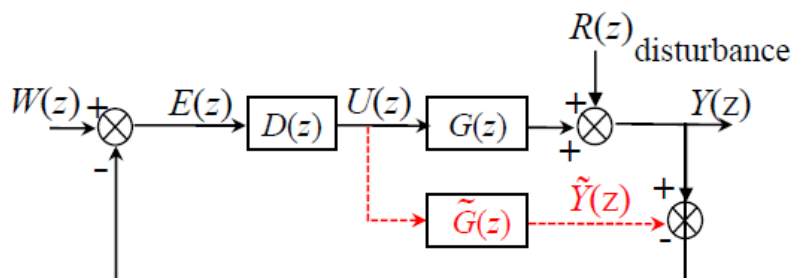zeros that outside the unit circle
zeros that lie inside the circle near (-1, 0)
steady state gain of unity <u>as denominator</u> (found by setting $z = 1$)

The controller is obtained by inverting good part of $G(z)$ and multiplying by a first-order filter $F(z)$ $\quad D(z) = \dfrac{F(z)}{\tilde{G}_-(z)}$

$F(z) = \frac{1-\alpha}{1-\alpha z^{-1}}, \frac{(1-\alpha)z^{-1}}{1-\alpha z^{-1}}$

L8 Model Predictive Control

163 Parameters

N model rank   T sample period
P prediction horizon   M control horizon

S is the $P \times M$ **dynamic matrix**

$$S = \begin{bmatrix} S_1 & 0 & \cdots & 0 \\ S_2 & S_1 & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ S_M & S_{M-1} & \cdots & S_1 \\ \vdots & \vdots & \ddots & \vdots \\ S_P & S_{P-1} & \cdots & S_{P-M+1} \end{bmatrix}$$

√ 164 154 Finite Step Response Model

$$\mathbf{y(t)} = \boldsymbol{L}^{-1}\left(\frac{H(s)}{s}\right) \quad \boldsymbol{S_i} = \boldsymbol{y(iT)}$$

$$y(k) = \sum_{i=0}^{N-1} S_i \Delta u(k-i) + S_N u(k-N)$$

$$\hat{y}^o(k+j) \triangleq \sum_{i=j+1}^{N-1} S_i \Delta u(k+j-i) + S_N u(k+j-N)$$

156 Model prediction

$$\hat{\mathbf{y}}(k+1) = \boldsymbol{S}\Delta\mathbf{u}(k) + \hat{\mathbf{y}}^o(k+1)$$

$y(k+j)$ 实际输出
$y^0(k+j)$ 非受迫输出，k 时刻起不进行任何控制
$\hat{y}(k+j)$ 预测输出
$\tilde{y}(k+j)$ k 时刻实际误差作为恒定误差的修正预测输出

158 Model prediction with current error

$$\delta(k) = y(k) - \hat{y}(k) \quad \tilde{\mathbf{y}}(k+1) = \hat{\mathbf{y}}(k+1) + \delta(k)\mathbf{1}$$

√ 160 Destiny function optimization

$$\min_{\Delta\mathbf{u}(k)} \phi = \tilde{\boldsymbol{e}}^T(k+1)\tilde{\boldsymbol{e}}(k+1)$$

$$\tilde{\boldsymbol{e}}(k+1) = \boldsymbol{w}(k+1) - \tilde{\mathbf{y}}(k+1) \;= \boldsymbol{w}(k+1) - \hat{\boldsymbol{y}}^0(k+1) - \delta(t)\mathbf{1} - \boldsymbol{S}\Delta\boldsymbol{u}(k) = \tilde{\boldsymbol{e}}^0(k+1) - \boldsymbol{S}\Delta\boldsymbol{u}(k)$$

$$\frac{\partial\phi}{\partial\Delta\mathbf{u}(k)} = 0 \quad\Longrightarrow\quad \Delta\mathbf{u}(k) = (\boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{S}^T\tilde{e}^o(k+1) \quad \frac{\partial(\tilde{e}(k+1)^T\tilde{e}(k+1))}{\partial\Delta u(t)} = -2S^T\tilde{\boldsymbol{e}}^0(k+1) + 2S^TS\Delta u(t)$$

$$\min_{\Delta\mathbf{u}(k)} \phi = \tilde{e}(k+1)^T\tilde{e}(k+1) + \alpha\Delta u(k)^T\Delta u(k) \quad \Delta\mathbf{u}(k) = (\boldsymbol{S}^T\boldsymbol{S} + \alpha I)^{-1}\boldsymbol{S}^T\tilde{\boldsymbol{e}}^o(k+1)$$

√ Matrix derivation

$$\frac{\partial\mathbf{A}^T x}{\partial x} = A = \left(\frac{\partial Ax}{\partial x}\right)^T \quad \frac{\partial x^T Ax}{\partial x} = (A + A^T)x \quad (AB)^T = B^T A^T \quad \frac{\partial f(x)}{\partial x} = \frac{\partial f^T(x)}{\partial x} \quad \frac{\partial f(x)}{\partial x} = \frac{\partial g(x)}{\partial x} \cdot \frac{\partial f(x)}{\partial g(x)}$$

162 Implementation

$$\Delta u(k) = \boldsymbol{K}_{c1}\tilde{\boldsymbol{e}}^o(k+1) \quad \boldsymbol{K}_{c1} \text{ is the firs row of } (\boldsymbol{S}^T\boldsymbol{S} + \alpha I)^{-1}\boldsymbol{S}^T \quad = \boldsymbol{K}_c \quad (M*P)$$

167 Extensions

Step response model for integrating process: replace $\hat{y}(k)$ with $\Delta\hat{y}(k)$
Step response model with known disturbances: a known disturbance variable can be included in step-response model
Reference trajectory specification: specify the reference trajectory to be the filtered setpoint
Prediction for multi-input multi-output systems: r inputs and m outputs

$$\min_{\Delta\mathbf{u}(k)} \phi = \tilde{e}(k+1)^T\boldsymbol{Q}\tilde{e}(k+1) + \Delta u(k)^T\boldsymbol{R}\Delta u(k) \quad \Delta U(k) = \left(\boldsymbol{S}^T\boldsymbol{Q}\boldsymbol{S} + \boldsymbol{R}\right)^{-1}\boldsymbol{S}^T\boldsymbol{Q}\tilde{\boldsymbol{e}}^o(k+1)$$

MPC with constraints
   A hard constraint cannot be violated at any time.
   A soft constraint can be violated, but the amount of violation is penalized by the objective function.