What Is PID—Tutorial Overview

PID stands for Proportional, Integral, Derivative. Controllers are designed to eliminate the need for continuous operator attention. Cruise control in a car and a house thermostat are common examples of how controllers are used to automatically adjust some variable to hold the measurement (or process variable) at the set-point. The set-point is where you would like the measurement to be. Error is defined as the difference between set-point and measurement.

(error) = (set-point) - (measurement) The variable being adjusted is called the manipulated variable which usually is equal to the output of the controller. The output of PID controllers will change in response to a **change** in measurement or set-point. Manufacturers of PID controllers use different names to identify the three modes. These equations show the relationships:

```
P   Proportional Band = 100/gain
I            Integral = 1/reset          (units of time)
D          Derivative = rate = pre-act   (units of time)
```

Depending on the manufacturer, integral or reset action is set in either time/repeat or repeat/time. One is just the reciprocal of the other. Note that manufacturers are not consistent and often use reset in units of time/repeat or integral in units of repeats/time. Derivative and rate are the same.

Proportional Band

With proportional band, the controller output is proportional to the error or a change in measurement (depending on the controller).

(controller output) = (error)*100/(proportional band)

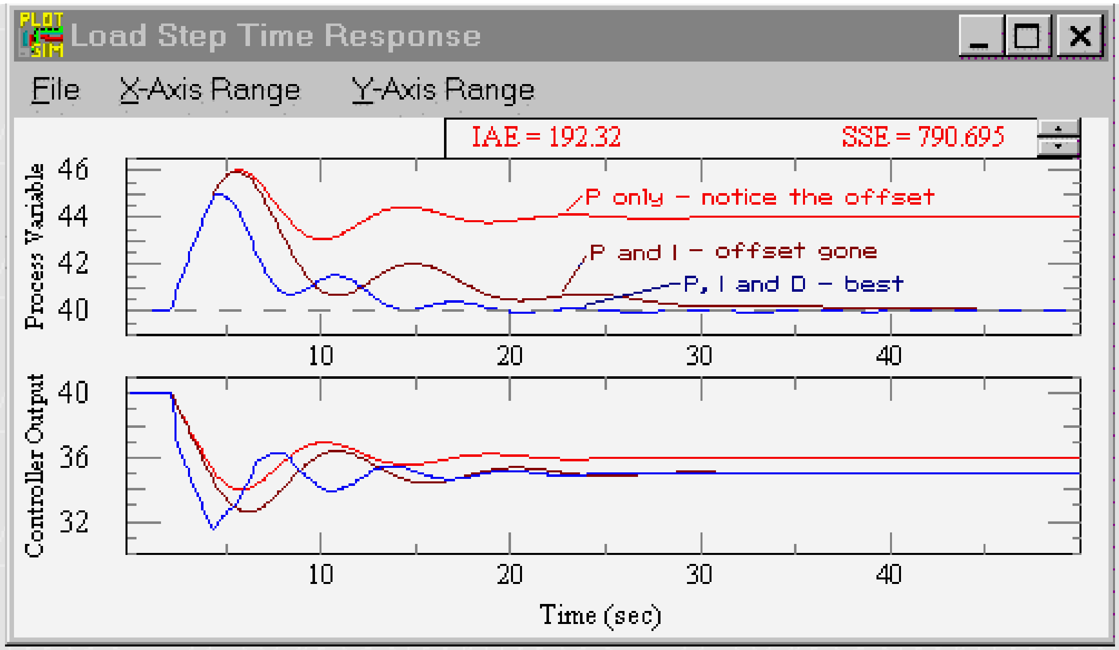
With a proportional controller offset (deviation from set-point) is present. Increasing the controller gain will make the loop go unstable. Integral action was included in controllers to eliminate this offset.

Integral

With integral action, the controller output is proportional to the amount of time the error is present. Integral action eliminates offset.

CONTROLLER OUTPUT = (1/INTEGRAL) (Integral of) e(t) d(t)

Notice that the offset (deviation from set-point) in the time response plots is now gone. Integral action has eliminated the offset. The response is somewhat oscillatory and can be stabilized some by adding derivative action. (Graphic courtesy of ExperTune Loop Simulator.)



Integral action gives the controller a large gain at low frequencies that results in eliminating offset and "beating down" load disturbances. The controller phase starts out at –90 degrees and increases to near 0 degrees at the break frequency. This additional phase lag is what you give up by adding integral action. Derivative action adds phase lead and is used to compensate for the lag introduced by integral action.

Derivative

With derivative action, the controller output is proportional to the rate of change of the measurement or error. The controller output is calculated by the rate of change of the measurement with time.

```
                                dm
CONTROLLER OUTPUT = DERIVATIVE ----
                                dt
```

Where *m* is the measurement at time *t*.

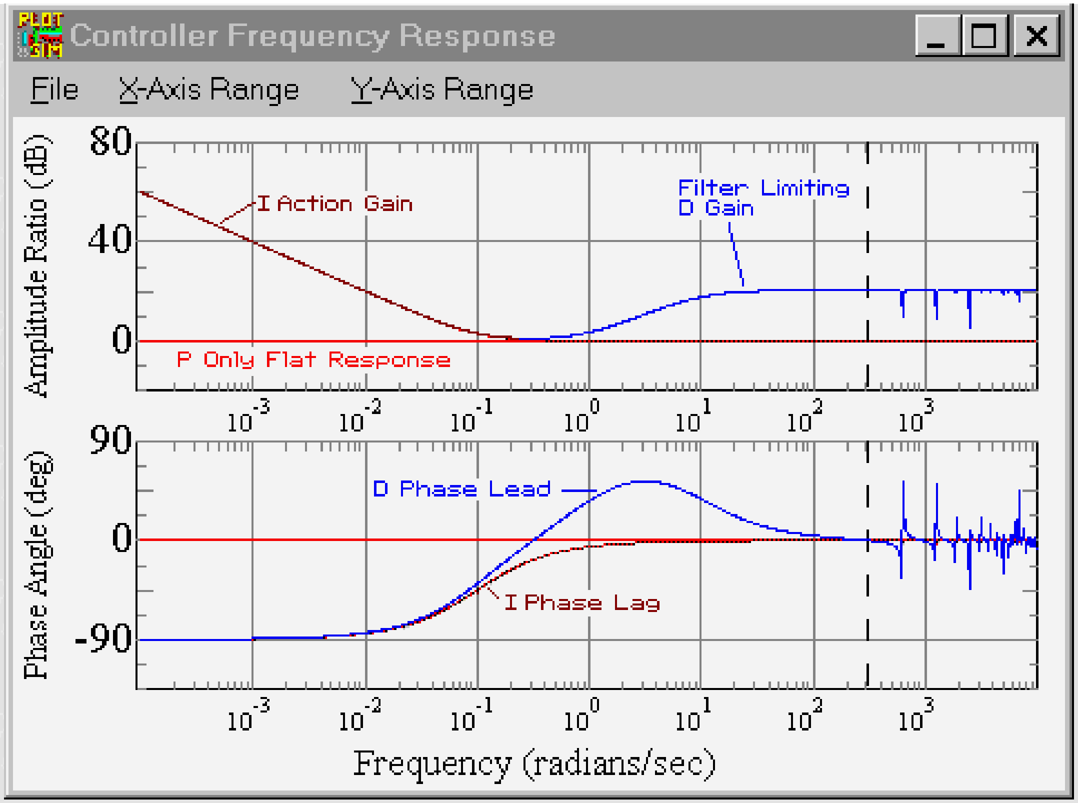
Some manufacturers use the term rate or pre-act instead of derivative. Derivative, rate, and pre-act are the same thing.

DERIVATIVE = RATE = PRE ACT

Derivative action can compensate for a changing measurement. Thus derivative takes action to inhibit more rapid changes of the measurement than proportional action. When a load or set-point change occurs, the derivative action causes the controller gain to move the "wrong" way when the measurement gets near the set-point. Derivative is often used to avoid overshoot.

Derivative action can stabilize loops since it adds phase lead. Generally, if you use derivative action, more controller gain and reset can be used.

With a PID controller the amplitude ratio now has a dip near the center of the frequency response. Integral action gives the controller high gain at low frequencies, and derivative action causes the gain to start rising after the "dip". At higher frequencies the filter on derivative action limits the derivative action. At very high frequencies (above 314 radians/time; the Nyquist frequency) the controller phase and amplitude ratio increase and decrease quite a bit because of discrete sampling. If the controller had no filter the controller amplitude ratio would steadily increase at high frequencies up to the Nyquist frequency (1/2 the sampling frequency). The controller phase now has a hump due to the derivative lead action and filtering. (Graphic courtesy of ExperTune Loop Simulator.)



The time response is less oscillatory than with the PI controller. Derivative action has helped stabilize the loop.

Control Loop Tuning

It is important to keep in mind that understanding the process is fundamental to getting a well designed control loop. Sensors must be in appropriate locations and valves must be sized correctly with appropriate trim.

In general, for the tightest loop control, the dynamic controller gain should be as high as possible without causing the loop to be unstable.

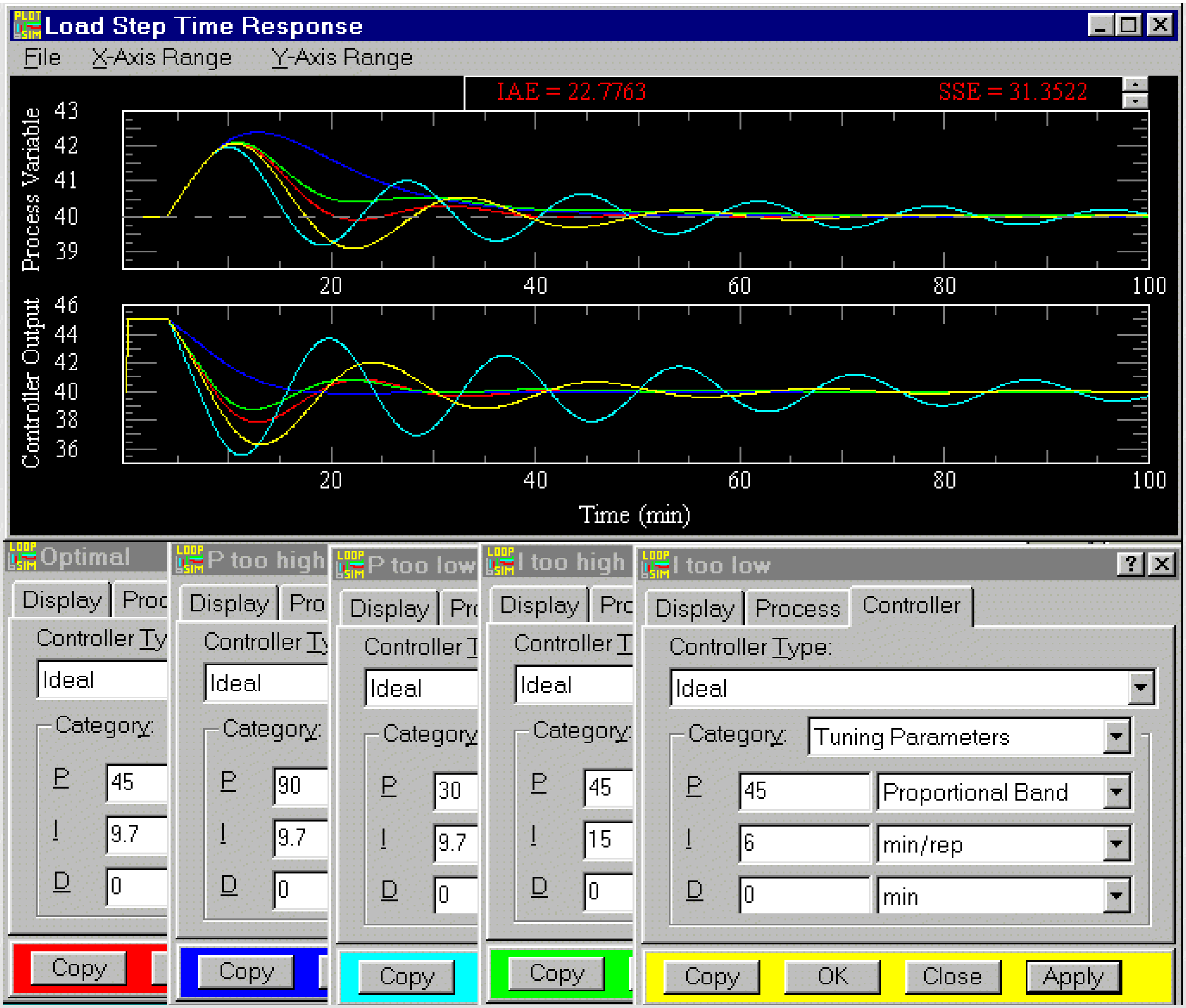
PID Optimization Articles

Fine Tuning "Rules"

This picture (from the Loop Simulator) shows the effects of a PI controller with too much or too little P or I action. The process is typical with a dead time of 4 and lag time of 10. Optimal is red.

You can use the picture to recognize the shape of an optimally tuned loop. Also see the response shape of loops with I or P too high or low. To get your process response to compare, put the controller in manual change the output 5 or 10%, then put the controller back in auto.

P is in units of proportional band. I is in units of time/repeat. So increasing P or I, *decreases* their action in the picture.



[View this as a full page](#) | **[PID Optimization Articles](#)**

Starting PID Settings For Common Control Loops

Initial Settings For Common Control Loops For Some Ideal and Series Controllers

Loop Type	PB %	Integral min/rep	Integral rep/min	Derivative min	Valve type
Flow	50 to 500	.005 to .05	20 to 200	none	Linear or Modified Percentage
Liquid Pressure	50 to 500	.005 to .05	20 to 200	none	Linear or Modified Percentage
Gas Pressure	1 to 50	.1 to 50	.02 to 10	.02 to .1	Linear
Liquid Level	1 to 50	1 to 100	.01 to 1	.01 to .05	Linear or Modified Percentage
Temperature	2 to100	.2 to 50	.02 to 5	.1 to 20	Equal Percentage
Chromatograph	100 to 2000	10 to 120	.008 to .1	.1 to 20	Linear

These settings are rough, assume proper control loop design, ideal or series algorithm and do not apply to all controllers.

Use ExperTune PID Tuner to find the proper PID settings for your process and controller.

Tuning process controllers starts in manual

Finding the lag and dead times and the process gain opens the door to PID control, efficiency, and higher profits.

PID controllers are designed to automatically control a process variable like flow, temperature, or pressure. A controller does this by changing process input so that a process output agrees with a desired result: the set point. An example would be changing the heat around a tank so that water coming out of that tank always measures 100° C.

Usually adjusting a valve controls the process variable. How the controller adjusts the valve to keep the process variable at the set point depends on process parameters entered into three mathematical functions: proportional (P), integral (I) and derivative (D). See *InTech's* January 1999 Tutorial for the details on the mathematics involved in P, I, and D control.

So, how does one set the parameters so that the controller does its job?

Some processes are unruly

First, know that there is more to tuning a PID loop than just setting the tuning parameters. The process has to be controllable. You won't be able to get good temperature in a hot shower if there is no hot water or if the adjusting valve is too small or too large.

Assuming the process can be conquered, then you can begin tuning it. The goal for good tuning is to have the fastest response possible without causing instability. One of the best tools for measuring response is integrated absolute error (IAE).

Honing in on the set point

A control scheme's goal is to minimize the time and magnitude that the process variable strays from the set point when an upset occurs. To calculate the IAE, simply add up the absolute value of the error during each digital controller sample.

Adding these values together yields a number. Adjusting the PID parameters to minimize this number is known as minimum integrated absolute error (MIAE) tuning. Graphically the IAE is the area in the graph between the set point and the process variable. In Figure 1, this area is colored blue.

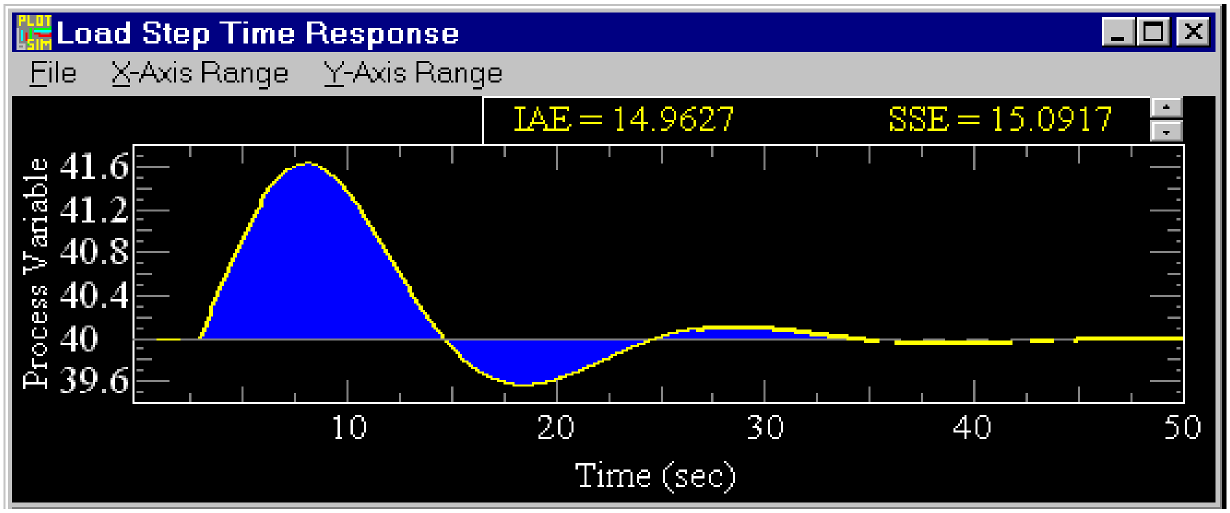


Figure 1. The error measurement is the area in blue. Minimizing this area maximizes the process's economic benefits.

A poorly tuned process results in sending a richer product than necessary out the door and with it, profits. Or, it causes off specification product, which requires rework and increased cost. With better tuning one can give away less while staying on spec.

For example, methyl tertiary butyl ethylene (MTBE) added to inexpensive gasoline increases the octane number. Because MTBE is expensive, you want to add just enough to reach the target octane level. Add too much MTBE and you give away unnescessarily strong gas. Add too little and the gasoline won't reach the regulated octane level. Ideally, you want to control the added MTBE to give the octane level close to the regulated level without going below it.

Bring in baseline parameters

To perform the tuning chore, certain fundamental measurements must be taken. Specifically the process's lag time, dead time, and gain must be determined. To do this, set the controller on manual. Set its output to somewhere between 10 and 90%. Then, wait for the process to reach steady state.

Next, change the controller output quickly in a stepwise fashion. The process variable will begin to change too, after a period of time. This period of time is called the process dead time.

The process lag time is how long it takes for the process variable (PV) to go 63% of the way to where it eventually ends up. This would mean that if the temperature increased from 100° to 200° , the lag time would be the time it took to go from 100° to 163° .

The process gain, or merely the gain, is found by dividing the total change in the PV divided by the change in the controller output.

Dead time dictates

A process that consists only of lag is easy to control. Simply use a P-only controller with lots of gain. It will be stable and fast. Unfortunately these processes are rare because of another dynamic element of most real processes: dead time.

Sometimes overlooked, dead time is the real limiting factor in process control. Dead time is the time it takes for the PV to just start to move after a change in the controller's output. During the dead time, nothing happens to the PV.

So, you wait. A control loop simply cannot respond faster than the dead time. Hopefully, the process is designed to make dead time as small as possible.

With dead time in the process, gain can be increased to get a faster response, but this will cause loop oscillation. If gain is increased even more, the process will become unstable.

Some like it simple

From the process gain, lag and dead times, we can build a simple tuning table for both PI and PID controllers. Table 1 comes from a controller design method called internal model control (IMC). Each cell yields a numerical setting that an operator plugs into a controller.

Controller Type	Controller Gain (no units)	Integral Time (seconds)	Derivative Time (seconds)
PI control	$\frac{\tau}{K(\lambda+\theta)}$	τ	not applicable
PID control	$\frac{\tau}{K(\lambda+\theta/2)}$	τ	$\theta/2$

θ = process dead time (seconds)

τ = process lag time (seconds)

K = process gain (dimensionless)

$\lambda = 2\theta$ used for aggressive but less robust tuning

$\lambda = 2(\tau + \theta)$ used for more robust tuning

Some controller mechanisms use proportional band instead of gain. Proportional band is equal to 100 divided by gain.

The values in the table are for an ideal type controller. The controller computes controller gain, integral time, and derivative time using the formulas shown. Other tables and computational methods, of which there are many, are needed for other systems.

Compare the methods for fun

Figure 2 compares the IMC tuning method outlined above to a more sophisticated method, the MIAE, which uses performance criteria developed using expert systems.

The process described was found to have a 30-second lag time, a 10-second dead time, and gain of 1. The more aggressive setting ($\lambda = 2\theta$) was used for the IMC method.

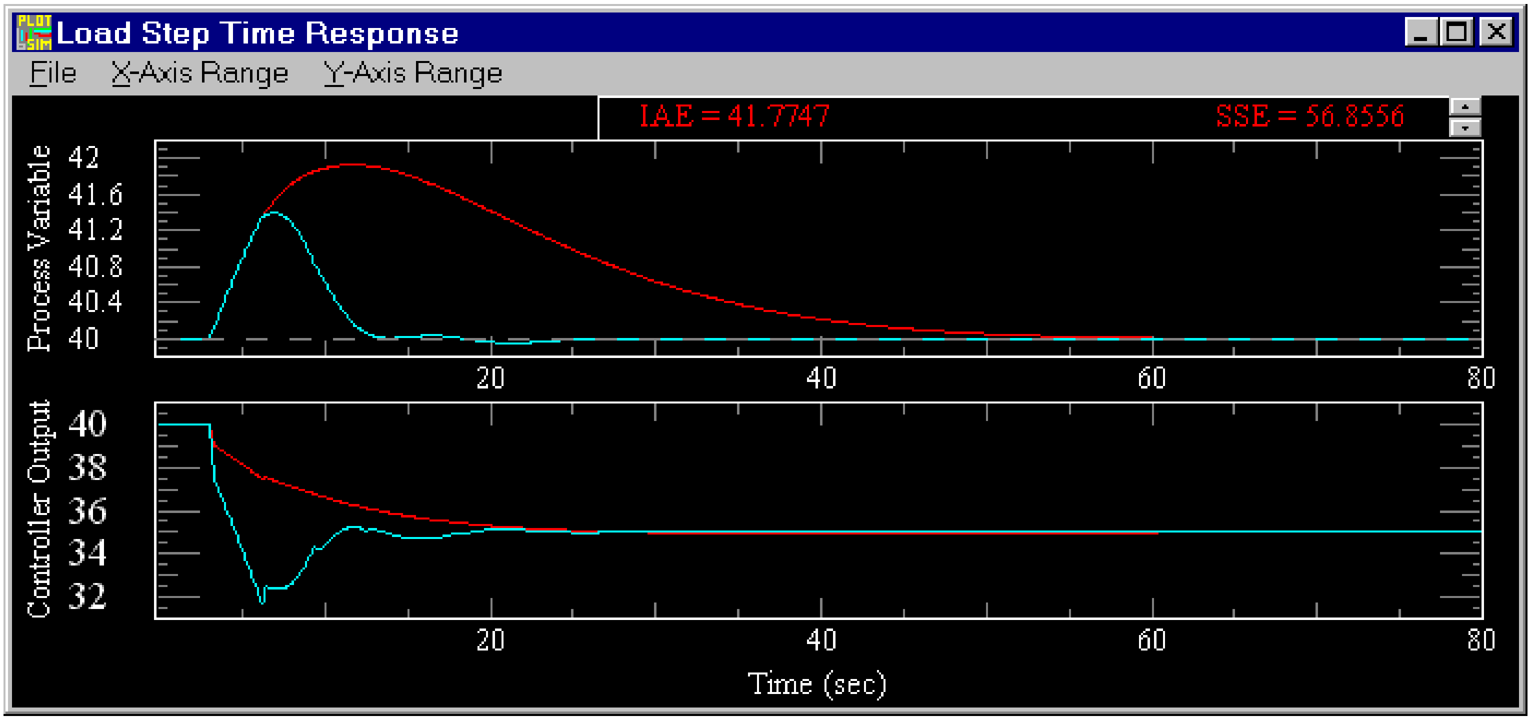


Figure 2. The red line is IMC tuning and yields an IAE of 42. The blue line shows an advanced tuning method which yields an IAE of 8.

The IMC does produce a nice smooth response and it provides a starting place for optimizing the control loop. However, tuning with a more advanced algorithm aimed at minimizing IAE gives an IAE that's better by a factor of 5. The advanced tuning method was much faster as well.

Further, the minimum IAE tuning ensures the minimum amount of excessively rich product production while staying close to and exceeding specifications. Thus, an improvement in IAE is directly proportional to the dollars saved. In this example, the IAE tuning saves the user 500% over the simpler IMC.

Start at ground zero

Assuming a given process and controller are of a specific type, a simple tuning method can get you started in setting PID parameters. Using more advanced optimization methods will enable increased process efficiency and higher profits.

Behind the byline

John Gerry holds a M.S. in chemical engineering from the University of Texas and he's a P.E. He has worked for Foxboro Company, Eastman Kodak, Eli Lilly, and S.C. Johnson. His article on power spectral density analysis appeared in *InTech's* August 1998 issue. He founded and is the president of ExperTune Inc., located in Hubertus, WI.

Credits

Simulations, figures, and IAE tuning provided by ExperTune Inc.

Control Fundamentals: PID settings sometimes fail to make the leap

Joe Engineer finally got that replacement for the single-loop controllers he'd been waiting for. The replacement was a shiny new programmable logic controller (PLC). "It should really do the job," he thought. "Well, now to tune the loops."

"Why not start with the settings from the old controllers?" The PLC had several proportional-integral-derivative (PID) equations to pick from, and Joe was careful to select one that matched his old single-loop controller equation.

Joe was also careful to convert the units between the two controllers. What a surprise when he put the PID settings in the new PLC! The controller outputs went crazy! He double-checked his conversions. They were correct. What now?

What happened to Joe? Why didn't those old settings work? As it turns out, some of the newer PLCs and distributed control systems (DCSs) give you a lot more control over the PID algorithm.

For example, in many PLCs you enter the update time of the PID loop in the PID block, and you trigger the PID block with a timer. The two times should be equal. If they're not, then the PID output will be unreliable. Sometimes, because of PLC scanning, it is difficult to tell whether the block triggers on time.

There are several ways to analyze your controller to see whether it is working properly. One way is to manually test the controller. Another method is more sophisticated and uses analysis software

Recall the PID terms

The "P" part of the controller is the proportional band, or gain setting. Some controllers use proportional band, and others use gain. Some manufacturers call the gain setting "proportional gain." They are related by the following:

Proportional band = 100/gain

With the "P" setting, the output of the controller is proportional to the change in the process variable.

The "I" part of the controller is the integral, or reset. Integral can be in any one of these units:

- minutes/repeat
- repeats/minute
- seconds/repeat
- repeats/second

With the "I" setting, the controller output continues to change based on the amount of time there is an error between set point and the process variable (PV).

For example, if the gain and integral settings of the controller equal 1 and the integral units are minutes/repeat, then the controller output will repeat the error every minute.

If there were a 10% error between set point and PV, then with our example the controller output would change 10% every minute.

The "D" part of the controller stands for derivative, or rate action. "D" action is in units of minutes or seconds. With derivative action, the output of the controller changes based on the rate of change in the process variable.

Experiment with numbers

Now to see if the controller is doing what Joe Engineer thinks it's doing.

The controller will need to be out of service. To test the "P" and "I" modes, enter a simulated or "fake" process variable value into the controller. Record the controller output (CO).

For each test, change the "fake" PV by 10%. To find out what a 10% change is in engineering units, multiply the process variable span by 0.1.

For example, if your controller PV span in engineering units is from 200° to 800°, then a 10% change would be (800-200)° x .01 = 60°.

Test the easiest first

The first and simplest mode to check is the proportional ("P").

1. First, eliminate "I" and "D" from the PID control algorithm.

    - If "I" is in minutes/repeat or seconds/repeat, then set it to a very large number, usually over 1,000 if it's in minutes/repeat or 100,000 if it's in seconds/repeat.
    - If "I" is in repeats/minute or repeats/second, then set it to 0.
    - Set "D" or derivative action to 0.
2. If the controller uses gain, set it to 1. If the controller uses proportional band, set it to 100.
3. Set the controller in auto with the output, set point, and PV at 50%:
    - Put the controller in manual, and set the output to 50%.
    - Set or force the PV to 50%.
    - Set the set point of the controller to 50% also.
    - Put the controller in auto and verify that the controller output is at 50%.
4. Change the PV to 60%.
5. Verify that the output of the controller changes by 10%.

If the controller output does not change by 10%, then the "P" part of the controller is not working properly. This may indicate a problem in scaling the controller inputs or outputs.

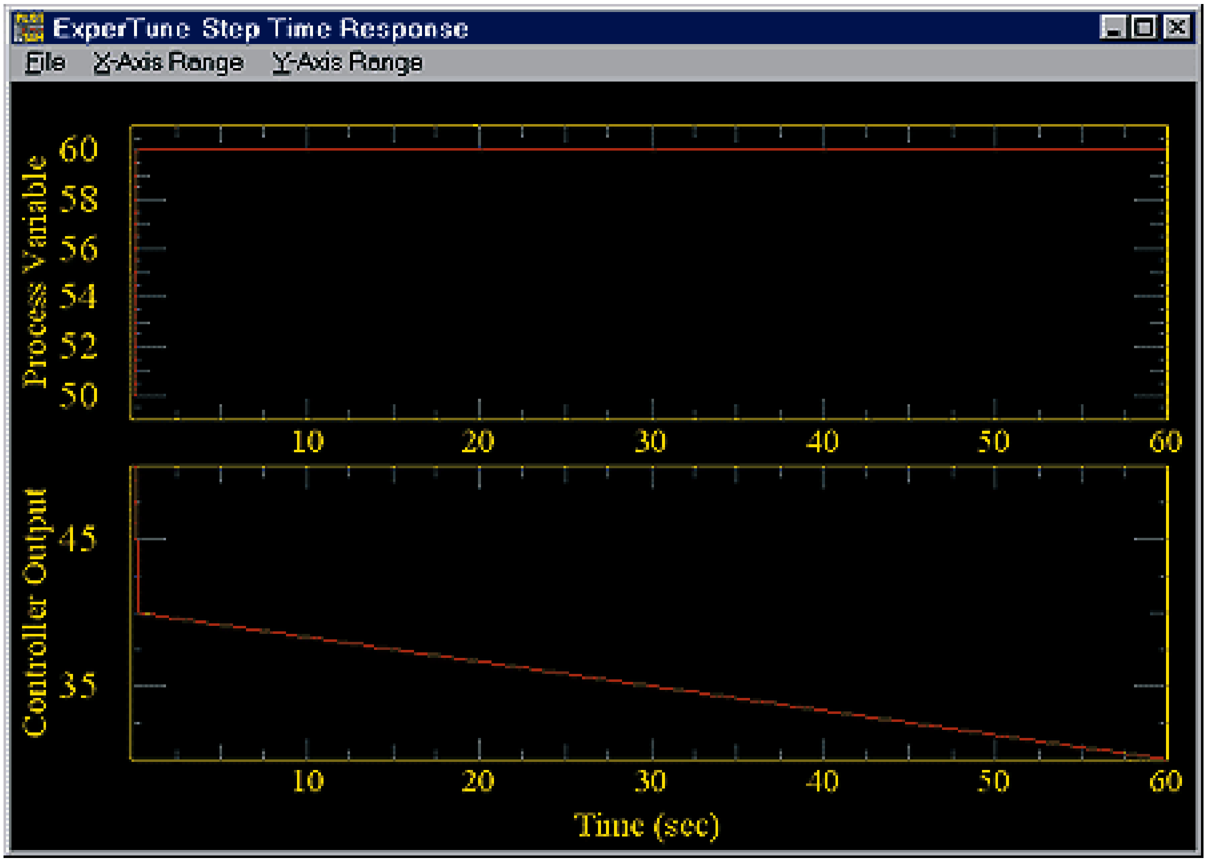
Integral action test

Now test the controller's "I" action.

1. Eliminate "D" from the PID algorithm by setting the derivative action to 0.
2. If your controller uses gain, set it to 1. If the controller uses proportional band, set it to 100.
3. If "I" is in:
    - minutes/repeat, then set it to 1.
    - seconds/repeat, then set it to 60.
    - repeats/minute, then set it to 1.
    - repeats/second, then set it to 1/60 or .0167.
4. Put the controller in manual, and set the output to 50%.
5. Set the PV to 50%.
6. Set the set point of the controller to 50% also.
7. Put the controller in auto and verify that the controller output is at 50%.
8. Now, change the PV to 60%.
9. The controller output should initially change 10% because of the "P" action in the controller. Then the output should change by an additional 10% every minute.

If the controller output did not change by an additional 10% every minute, then the "I" part of the controller is not working properly.

If this is the case, check that the timing is correct in the PID controller. Is the update time set correctly? Is the triggering for the controller set correctly?



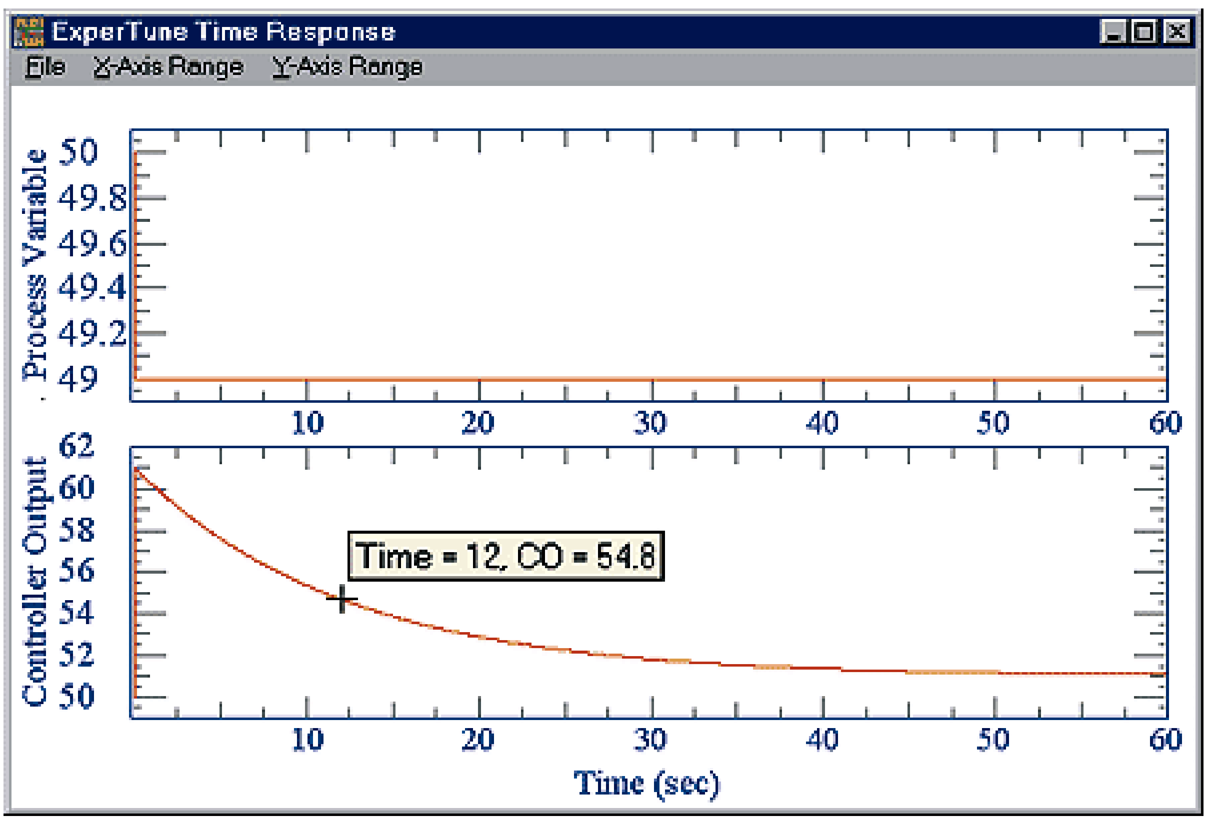
In the top graph, see that the PV was changed from 50% to 60%. The PI controller output shown in the bottom graph initially falls 10% because the gain in the controller is 1. This is a result of the proportional "P" compnent of the PI controller. Because the intergral time is set to 60 seconds, every 60 seconds the output changes by another 10%.

Ride the "D" train next

Testing derivative action is more complex because of the interplay with derivative gain limit or derivative filtering. This test requires having a recorder connected to the controller output. Because there are many different ways to compute the derivative component, this test is an approximation.

1. Eliminate "I" from the control algorithm. If the integral mode's units are in:
    - minutes/repeat or seconds/repeat, then set "I" to a very large number, usually over 1,000 if "I" is in minutes/repeat or over 100,000 if "I" is in seconds/repeat.
    - repeats/minute or repeats/second, then set it to 0.
2. If the controller uses gain, set it to 1. If the controller uses proportional band, set it to 100.

3. If "D" is in:
   - minutes, then set it to 1.
   - seconds, then set it to 60.
4. Put the controller in manual and set the output to 50%.
5. Set or force the PV to 50%.
6. Adjust the set point of the controller to 50% also.
7. Put the controller in auto and wait for the response to settle. Verify that the controller output is at 50%.
8. If the controller is direct acting, change the PV to 49%. If the controller is reverse acting, change it to 51%.
9. The controller output should exponentially decay as shown in the figure below.
10. Measure the peak of the controller output. This value minus 51% is the derivative gain limit, which has no units.
11. Measure the time it takes for the output to reach 54.7%. This time multiplied by the derivative gain limit yields the derivative time.



The "D" gain limit is 10. This is figured from the y axis from the CO graph (61-51=10). The CO was at 54.7% after 12 seconds. Thus, the derivative time is 120 seconds (12 x 10).

If the controller's derivative time is significantly different from the entered value, the problem is probably in the timing. See that the update time is set correctly. Make sure someone correctly triggers the controller. (Author's note: The example test shown in the figure yeilds a D time of 120 seconds. Either this controller is not working properly or the user dialed in an actual D time of 120 seconds and not 60.)

However, the derivative setting can be tricky to get right, and there may be other problems that only the manufacturer can repair.

Pinpoint the algorithm type

The above tests are applicable for all three classes of industrial controller algorithms: ideal (ISA or noninteracting), series (interacting), and parallel. However, because the test works for any of these types, it does not confirm or check the algorithm type you are using.

Each class of algorithm behaves differently and requires very different PID tuning. Software analysis tools exist that check not only the proper behavior of the controller actions but also the algorithm types.

Loop Optimization:
Before You Tune

Plant efficiency and consistent product quality depend on proper loop performance, but tuning the controller is only the last step. This is the first of a three-part series on loop optimization. In April, Part II will describe how to optimize loop characteristics. And finally, in May, Part III will cover PID tuning.

There is much to be gained by optimizing control loops. It has been estimated that 80% of process control loops are causing more variability running in automatic mode than in manual. The often-quoted EnTech study showed that some 30% of all loops oscillate due to nonlinearities such as hysteresis, stiction, deadband, and nonlinear process gain. Another 30% oscillate because of poor controller tuning.

With a poorly optimized loop, an upset in the direction towards inefficiency results in giving away product. Alternatively, a load may cause off-spec product. When a control loop is running optimally, variability is minimized. Better tuning keeps the process on spec and reduces giveaway of often-expensive ingredients.

But tuning objectives vary for different types of processes. For example, in a steam header, the pressure has to be maintained at the maximum allowable without large errors so the safety valves will not open. The PID controller must be tuned tightly to ensure the valve that controls the flow from the main header will move quickly to eliminate effects of disturbances.

On the other hand, the PID controller of a robot arm that manipulates nitroglycerin vessels has a different objective. The control loop must be optimized to change the setpoint without overshoot or cycling.

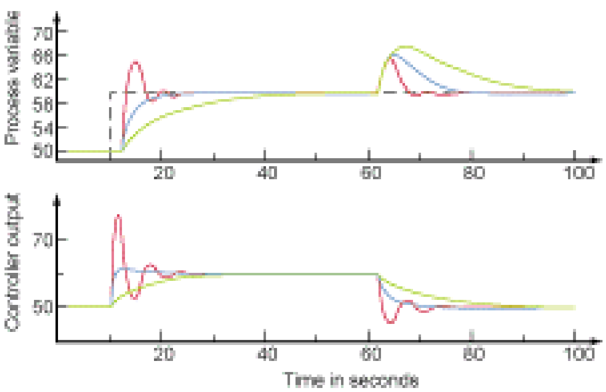
Performance Objectives

Most engineers and technicians tune process control loops using trial and error, observing the response to setpoint changes. To achieve good setpoint response takes a skilled intuitive understanding of the shape and speed of response. Only experienced people are able to achieve good setpoint response this way.

Unfortunately, once a loop is tuned for good setpoint response, the response to upset is usually very sluggish. Good setpoint tuning does not automatically result in good recovery from upsets. Unfortunately, it is upsets that usually are the source of off-spec product and poor variability.

Using modern tools to analyze a loop will give the engineer or senior technician helpful hints about the process: numbers and graphics will inform the user about design, equipment performance, and interactions with other loops. Modern tools also let the engineer or the technician select appropriate tuning parameters for the control objective. And since the algorithms used in PID controllers are different from one manufacturer to another, in many cases the algorithm is user selectable.

WHAT DO YOU WANT?



The same loop can be tuned for robustness (green), neutral response (blue), or speed (red) depending on the objectives.

The characteristics of good control (Table I) are difficult to obtain. When tuning a loop, one must make compromises between robustness and speed of response. Robustness is the ability of the control loop to remain stable when the process (mainly dead time or process gain) changes. Usually, to obtain robustness:

- Speed of response is longer,
- Errors are greater when a disturbance occurs, and
- Disturbances are not easily rejected.
- If the response is fast, it usually indicates:
- The loop is less robust,
- Errors are small when a disturbance occurs, and
- Disturbances are quickly rejected.

The trends in Figure 1 show the same flow loop tuned for different objectives.

A control loop consists of the process, measurement, controller, usually a current to pneumatic (I/P) transducer,

WHAT IS GOOD CONTROL?
Good setpoint response without overshoot.
Good setpoint response with a maximum overshoot.
Response time matched with another loop so loops will be synchronized.
Response time long enough to ensure the loop will not react with another loop.
Load disturbance quickly rejected.
Load disturbance rejected without cycling.
Robust tuning so the loop will remain stable when the process changes.
Aggressive tuning so the error will remain small enough to keep the product in specs.

and valve. Optimal process control depends on all of these components working properly. Hence, before tuning a loop, one must verify if each component is operating properly and if the design is appropriate.

Choosing the optimal PID tuning should be done after making sure all of the other components are working properly. The optimal tuning parameters ensure your equipment is used at maximum efficiency.

Questions to Be Answered

The following steps outline a procedure for approaching and optimizing a process control loop. Optimization requires observation in manual and automatic modes, and at various operating conditions. We need to answer the following questions:

1. Process gain: Is the control valve sized properly? Often, valves are oversized. If so, the controller output will be at one end of the range when the loop is in automatic. Also, oversizing the valve will amplify nonlinearities such as hysteresis, stiction, different response to small and large changes, and operating near the seat.

The process gain should be between 0.3 and 3. The ideal process gain is 1. A process gain too high will not permit the controller to work at its full potential: the controller will have to be tuned with a small proportional gain.

2. Hysteresis/stiction: Does the control valve have harmful hysteresis and/or stiction? Hysteresis is a difficulty but stiction is really the main problem. Stiction occurs when friction is present.

Hysteresis should be less than 3%, significantly less if the loop is to be tuned tightly. Stiction should be less than 1% and often 1% is too much.

3. Sensor/transmitter: Is the measurement sensor working properly? From your experience, do the numbers make sense? For example, is the dead time small enough? If a transmitter is not properly installed, the dead time can be too long; if a filter is added in the transmitter, the equivalent dead time could be longer.

4. Noise band: Is there an excessive amount of noise in the loop? When disturbances occur too fast to be removed by the PID controller, they are called noise. Filtering may help. The filter should be small enough to not increase the equivalent dead time and large enough to reduce the noise.

Selecting the filter time constant is a tradeoff between increasing the equivalent dead time and reducing the amount of noise. When the noise is reduced, the controller output is smoother.

5. Nonlinearities: How nonlinear is the loop? A loop is nonlinear when the process gain varies. All loops are somewhat nonlinear. It is the degree of nonlinearity that we are interested in. If the loop gain varies by more than a factor of two or three, then linearization will help optimize the loop.

6. Asymmetry: Does the loop respond differently in one direction than in the other? Often, a valve responds more quickly in one direction than the other. Also, in temperature processes using one fluid to add heat and another to remove heat, the two fluids are different and the characteristics of the process are different.

If the equivalent dead time or the equivalent time constant are different depending on the direction, use the worst case to tune the loop or use a special algorithm.

7. Tuning: Is the loop optimally tuned? If the loop is tuned aggressively to minimize error, the robustness is small; if the loop is tuned sluggishly to reduce variability, the recovery time after a disturbance is long.

Tuning parameters are selected to make a compromise between robustness and performance. The loops upstream could interact--selecting the appropriate tuning parameters will allow decoupling. At the opposite, if loops need to be synchronized, selecting the appropriate tuning parameters will ensure they work in accordance.

Next: Diagnosis

Each of these problems has a characteristic signature, which can be found by performing a series of tests and analyzing the results. The tests, which will be covered in detail in the next installment of this series, start with collecting process variable and controller output data with the controller in automatic at normal operating conditions, then introducing a setpoint change. Data is also collected with the loop in manual mode.

You will be able to see how the operating range for the valve and its performance can tell you if the valve is sized correctly; whether loop cycling is being caused by hysteresis, nonlinearities, or poor tuning; and the other critical aspects of loop performance that must be understood before tuning the controller.

Michel Ruel, P.E., at TOP Control Inc., St. Romuald, Quebec, an engineering company specializing in optimization of continuous and batch proces control. Ruel has 22 years of plant experience at companies including Monsanto, Domtar Paper, Dow Corning, and Shell Oil. Author of several publications on instrumentation and control and frequent university lecturer, Ruel is experienced in solving unusual process control problems and a pioneer in implementing fuzzy logic in process control. His e-mail address is mruel@topcontrol.com

Use Derivative Action Responsibly

Derivative action can give you the fastest response in loop tuning, but only if you use it responsibly.

- Process and advanced control
- Software and information integration
- Loop-tuning software
- PID (proportional-integral-derivative)
- Process control valves

Too much derivative will make the loop unstable. Too little deviative will not benefit response, and could result in, for instance, a jittery control valve, creating greater wear and shortening time to replacement.

Stabilize with derivative

In PID (proportional-integral-derivative) loop tuning, adjustments can be made to one or more variables to create changes.

Using the appropriate amount of derivative (or D) allows using more proportional and especially more integral actions, resulting in a much faster-reacting control loop. Properly applied derivative action creates the most pronounced changes on second-order processes like temperature loops. But D action also can help the response of most loops.

For example, in an easy comparison, two controllers (see screen capture) running a temperature loop have the same proportional and integral terms. As an upset is applied, the PI-only curve swings visibly more widely at least four times, compared to the second, which adds derivative action.

A plot of dead time against process gain for each controller shows a 53% increase in robustness (see second screen capture). Clearly, D action can smooth the response, creating a more robust loop.

Controllers, processes differ

If setpoint response is important to a loop, then the effects of derivative on the setpoint should be examined. Some controllers allow removing derivative action from setpoint changes. In a simulated comparison of Honeywell Plantscape A and B algorithms, each with derivative, algorithm A reacts slightly more quickly with a large initial spike in the controller output. Algorithm B doesn't react as quickly when it does respond (see third screen capture).

Most processes can be helped by derivative action, except those with almost pure dead time, which should not use D. These are somewhat rare. An old rule of thumb is to not use D on noisy loops.

Process control always has tradeoffs. If the loop is noisy, D action will make the control valve move more, causing more wear on the valve, decreasing its life. This is why using just a little bit of D can harm the loop: it does little to improve performance and wears out the valve. When using D, use the full and proper amount. Additional filtering can help counteract control valve wear, if the filter is the right size.

Other cautions; bad rap

Apply derivative only on controllers that limit the derivative gain. Manufacturers limit derivative gain by applying a first- or second-order filter to the process variable or to the error signal when the user enters a setting for D.

Without such filters, using D action in the presence of any amount of noise would continually the smash controller output into the upper or lower limits. In the lower plot on the fourth screen capture, the blue line shows such an event. The example demonstrates, in part, why D often receives a bad rap. The red plot shows the same controller with the proper D gain limit.

There is another caution with parallel-type controllers. ("Comparison of PID Control Algorithms," *Control Engineering* , March '87, explains PID controller types.) On many of these controllers, the D gain limit changes with the dialed-in value of controller gain.

Dial a gain of 1, and everything works as expected. Dial in progressively smaller controller gains, and the D gain limit slowly vanishes. Put in a large controller gain, and the D gain limit is so close to the actual D that it ends up canceling the D action. With parallel controllers, do not use D unless the controller gain is close to 1.

Filtering helps

A temperature-loop simulation applied to two identical PID controllers shows how additional filtering can make a difference. With a filter, performance is unchanged, robustness hurt slightly, but valve travel and reversals are dramatically reduced. In this case, the filter is probably worth it. Make sure that any filter added isn't "big" enough to hurt loop performance.

Properly set, derivative action improves the response of most loops. Even with the proper setting, be careful to examine tradeoffs in valve wear. Analysis software can help in comparing options, and provide decision-making tools for loop tuning.

PID ALGORITHMS and TUNING METHODS

The PID control algorithm is used for the control of almost all loops in the process industries, and is also the basis for many advanced control algorithms and strategies. In order for control loops to work properly, the PID loop must be properly tuned. Standard methods for tuning loops and criteria for judging the loop tuning have been used for many years, but should be reevaluated for use on modern digital control systems.

While the basic algorithm has been unchanged for many years and is used in all distributed control systems, the actual digital implementation of the algorithm has changed and differs from one system to another and from commercial equipment to academia.

We will discuss controller tuning methods and criteria. Also discussed will be the digital PID control algorithm, how it works, the various implementation methods and options, and how these affect the operation and tuning of the controller.

Table of Contents

4. Additional PID concepts

Interactive or Noninteractive algorithm
Converting between interactive and non-interactive
External feedback
Saturation Properties

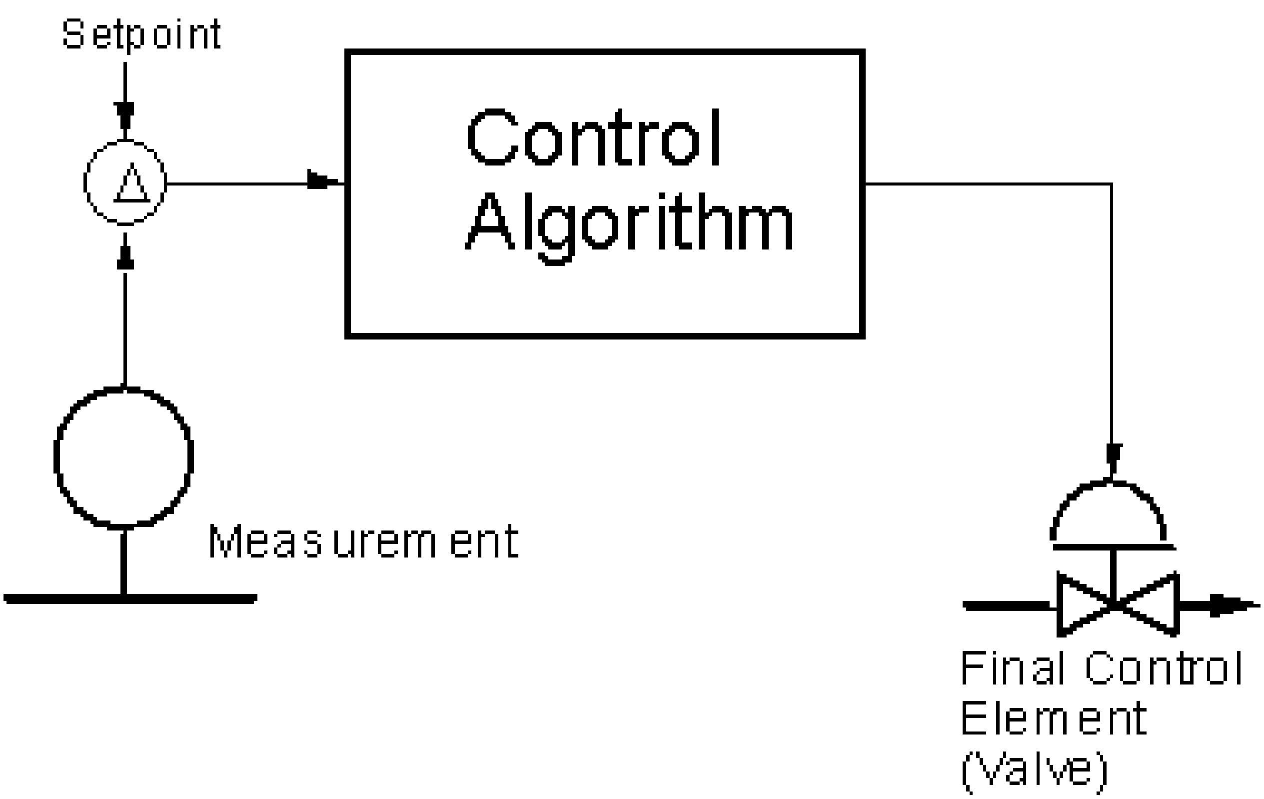
5. Other controller features

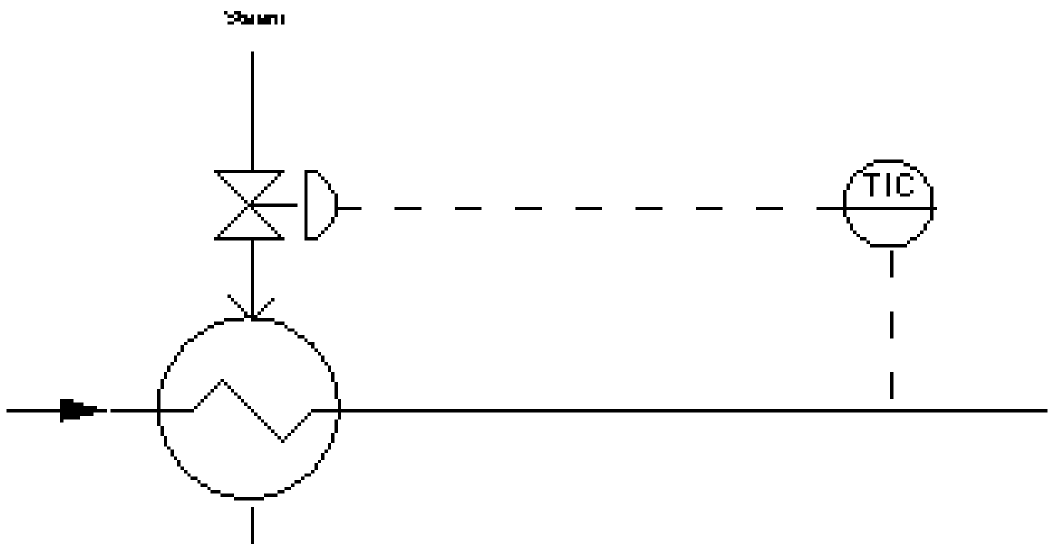
Gain on process rather than error
Derivative on process rather than error

6. Loop tuning

Tuning Criteria
Mathematical criteria
On-line trial tuning
Ziegler Nichols tuning method: open loop reaction rate
Ziegler Nichols tuning method: open loop point of inflection
Ziegler Nichols tuning method: open loop process gain
Ziegler Nichols tuning method: closed loop
Controllability of processes
Flow loops

Chapter 1
The Feedback Control Loop





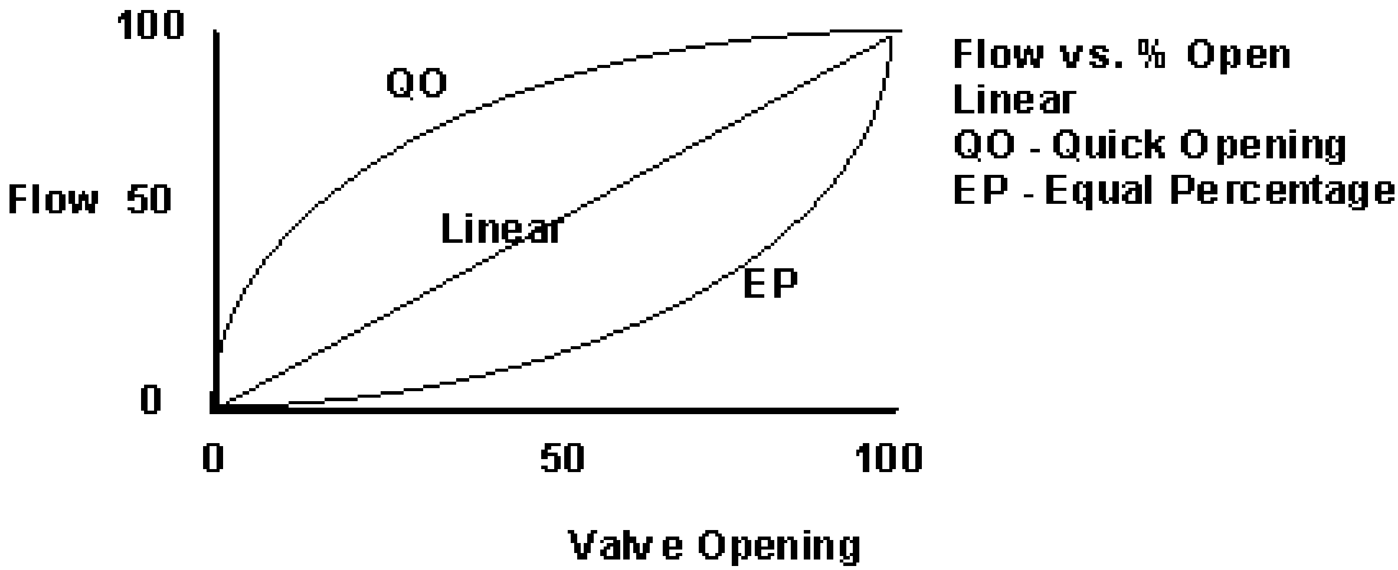
The system measures the process, compares it to a setpoint, and then manipulates the output in the direction which should move the process toward the setpoint.

Valve Linearity

Valves are usually non-linear. That is, the flow through the valve is not the same as the valve position. Several types of valves exist:

- Linear
    - Same gain regardless of valve position
- Equal Percentage
    - Low gain when valve is nearly closed
    - High gain when valve is nearly open
- Quick Opening
    - High gain when valve is nearly closed

Low gain when valve is nearly open

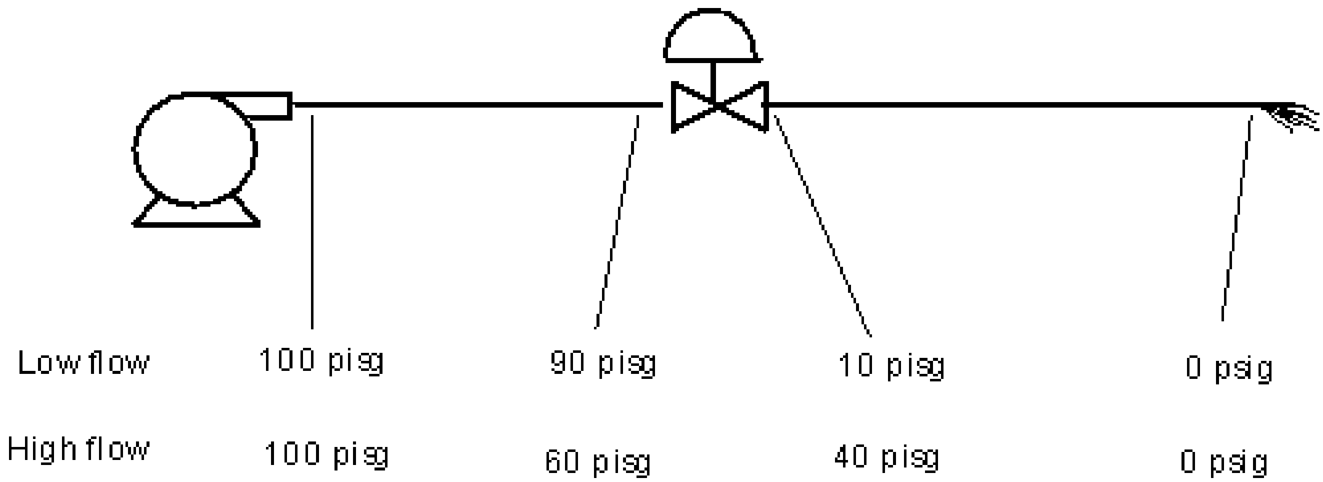


As we will see later, the gain of the process, including the valve, is very important to the tuning of the loop.

• If the controller is tuned for one process gain, it may not work for other process gains.

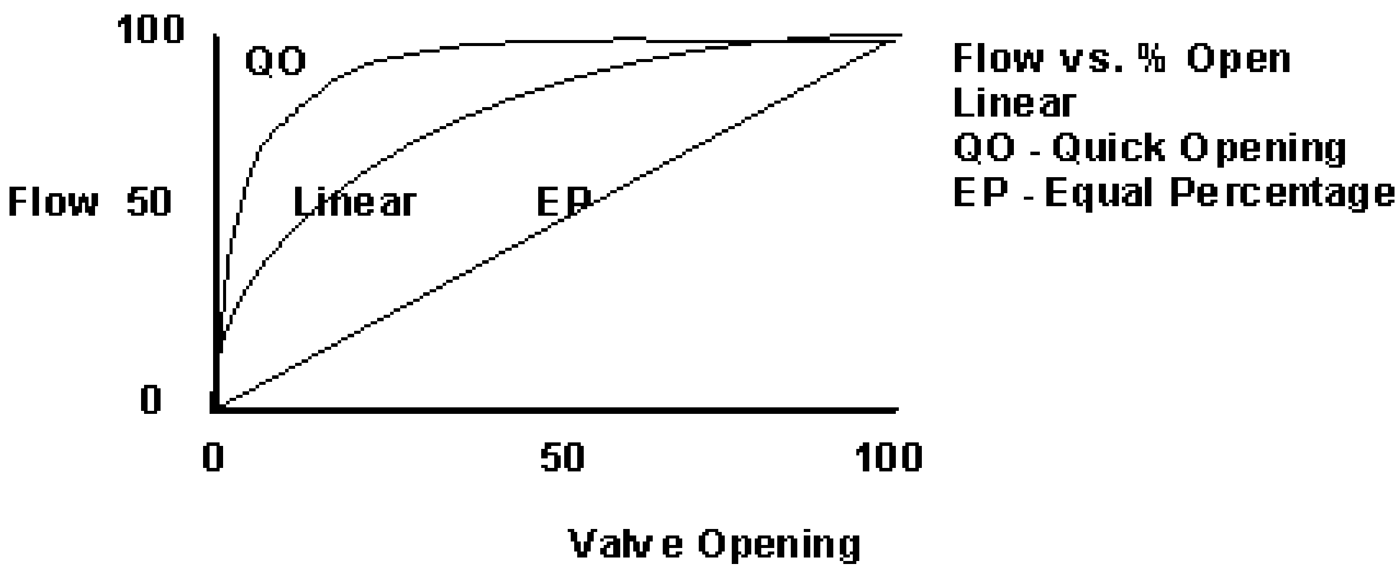
Valve Linearity:
Installed characteristics

The flow vs. percent open curve changes due to the head loss in the piping



At **low** flow, the head loss through the pipes is **less**, leaving a **larger** differential pressure across the valve.

At **high** flow, the head loss through the pipe is **more**, leaving a **smaller** differential pressure across the valve.



The effect is to increase the non-linearity of most valves.

Fail Open Valves

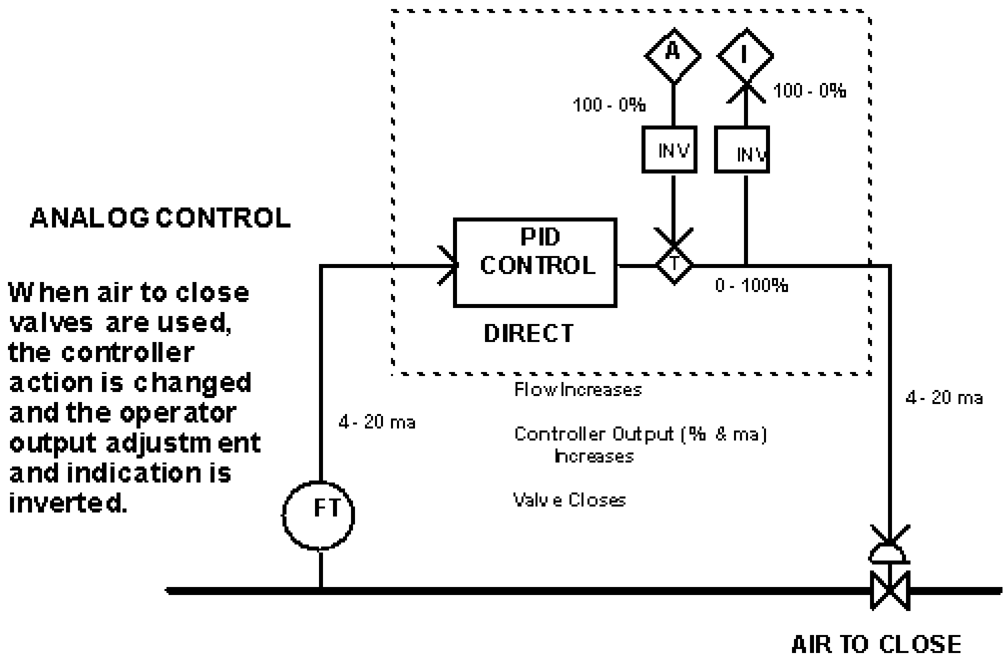
Valves are usually either: Fail Closed, air to open or

Fail Open, air to close

• Regardless of the way the valve operates, the operator is interested in the knowing and adjusting the position of the valve, not the value of the signal. **"Up is always open"**

• All controllers have some means of indicating the controller output in terms of the valve position. When the operator increases the output as indicated on the controller, the valve opens.

Indication Inversion



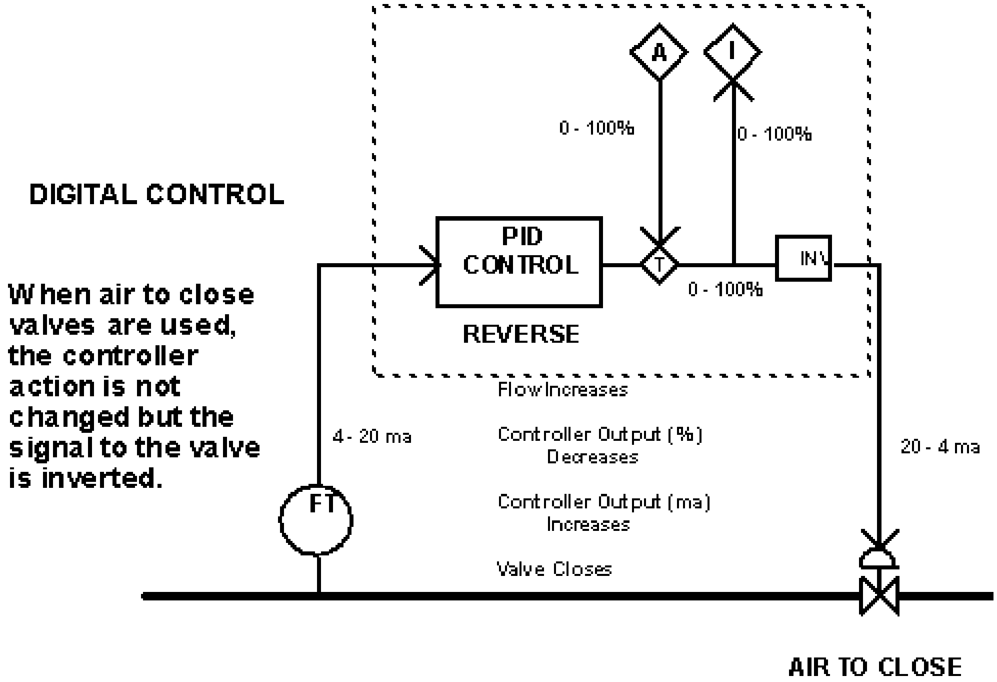
The output **indication** is inverted.

The controller action takes the valve action into acount.

The flow loop is **direct** acting.

Most analog controllers work like this.

Signal Inversion



The output **signal** is inverted.

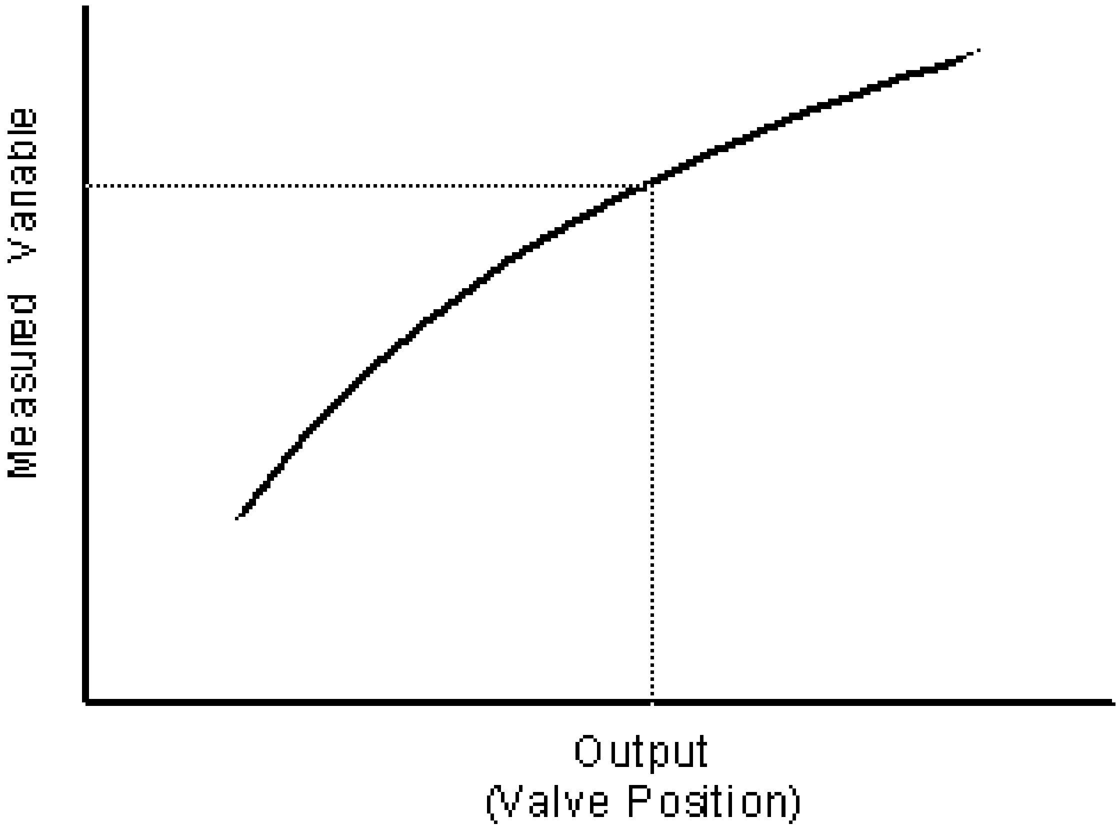
The controller action ignores the valve action.

The flow loop is **reverse** acting

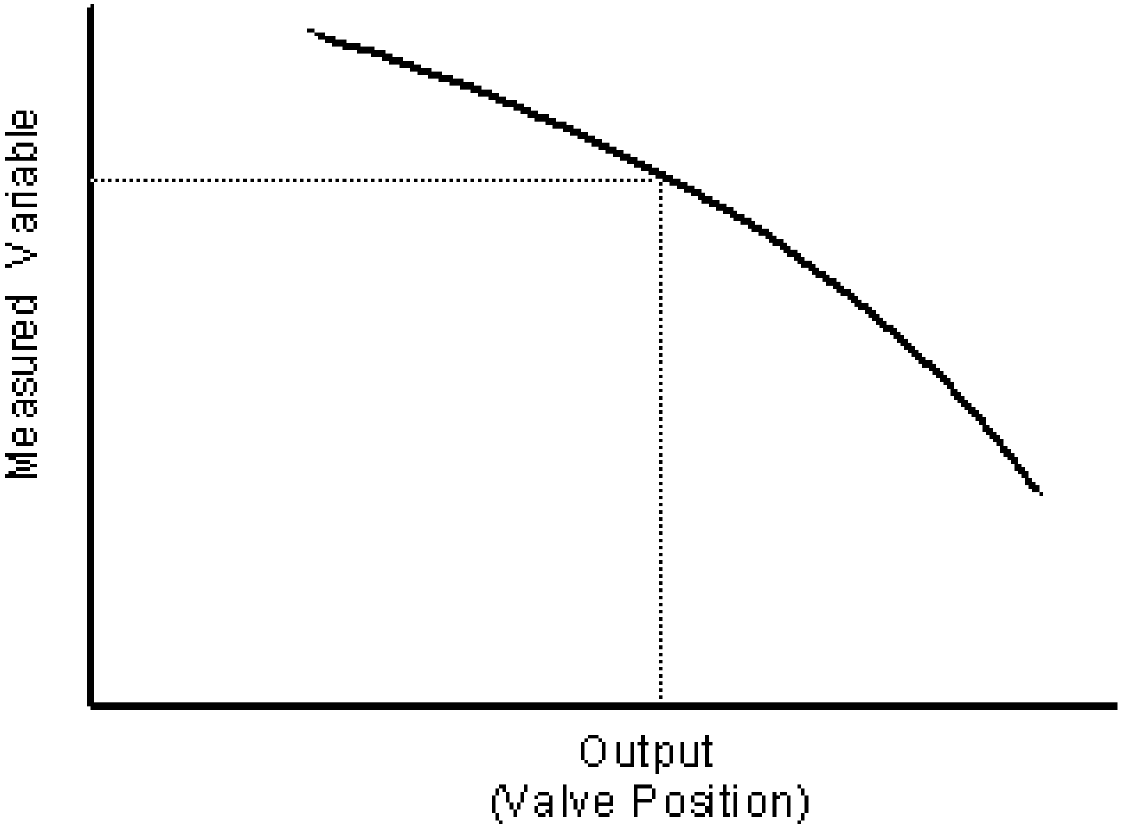
Some distributed control systems work like this.

Chapter 2
The Process Response to the Controller

Steady state relationships:
Relating valve change to measurement change

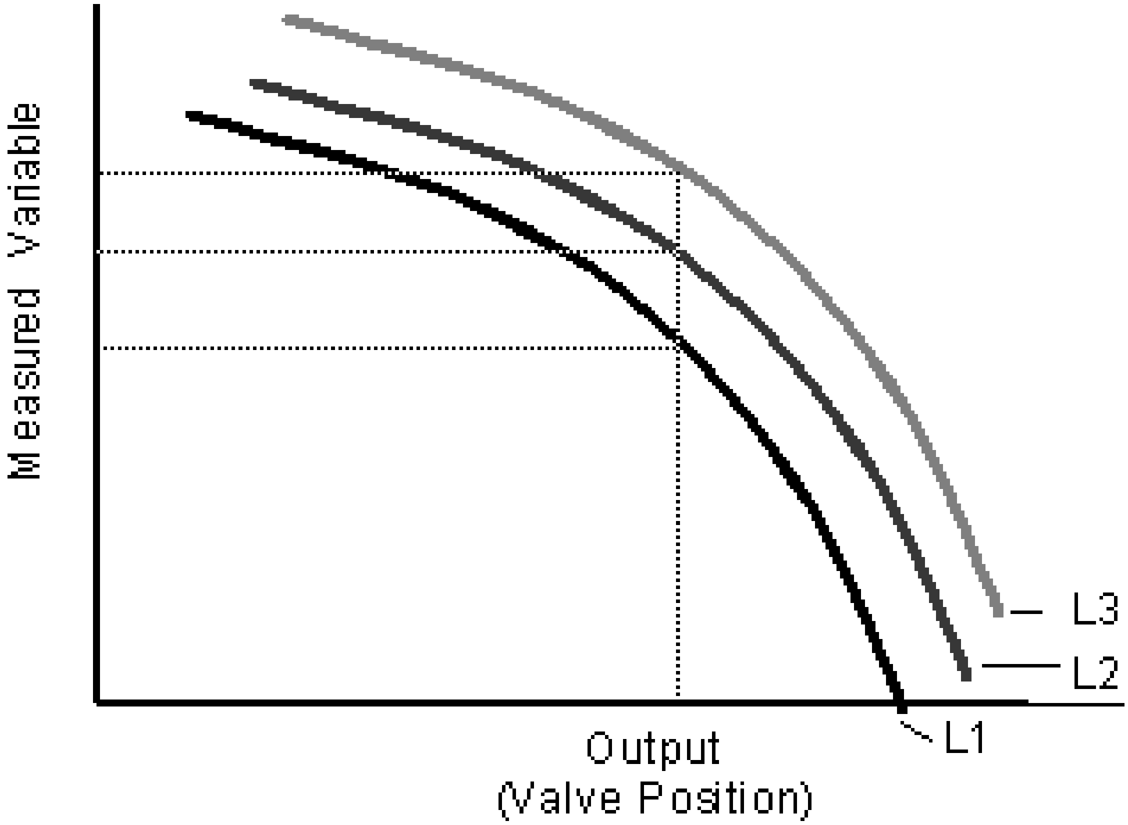


Direct Acting Process

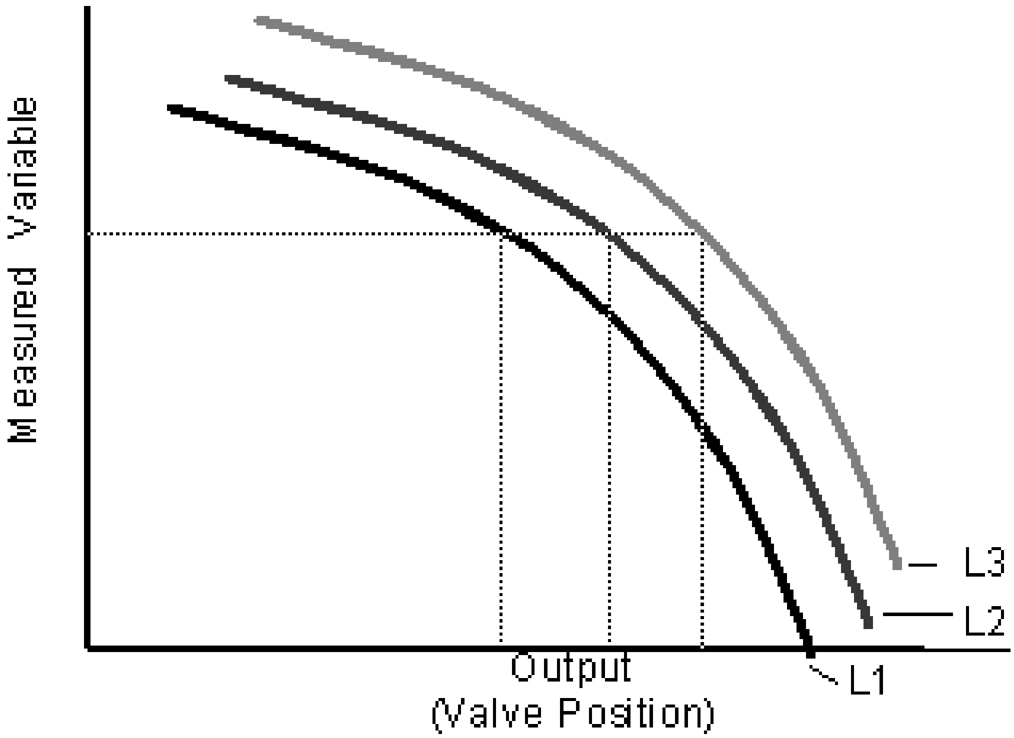


Reverse Acting Process

Steady state relationships: changing load



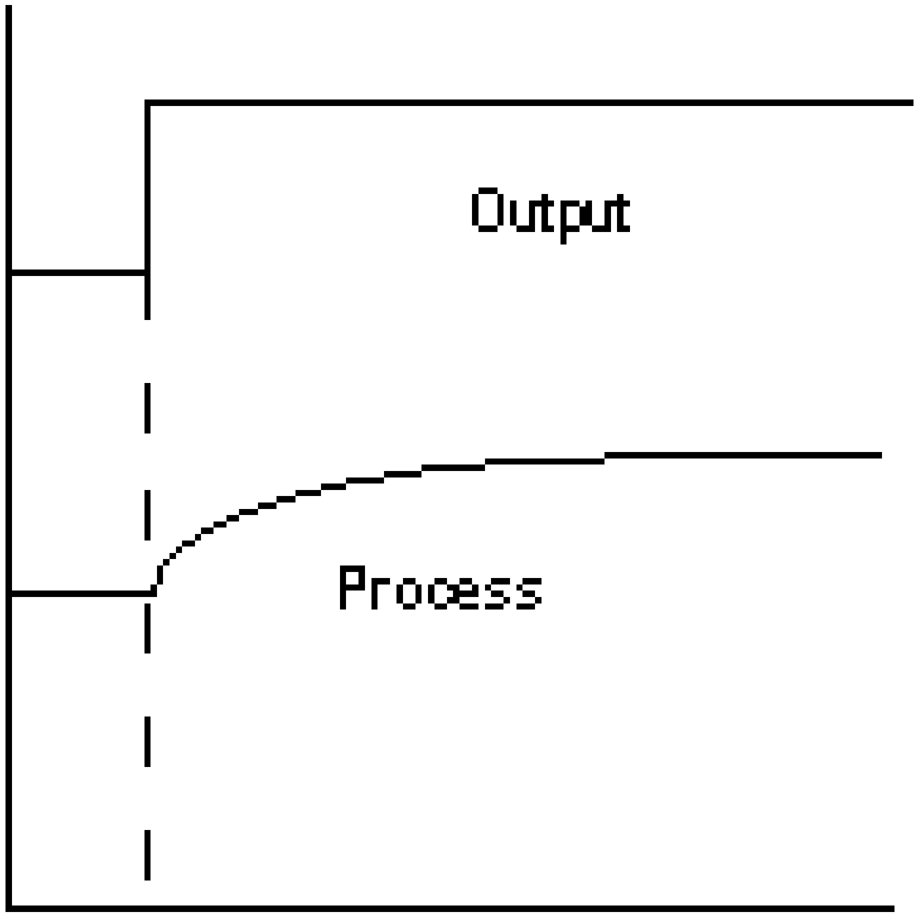
Load change with constant output



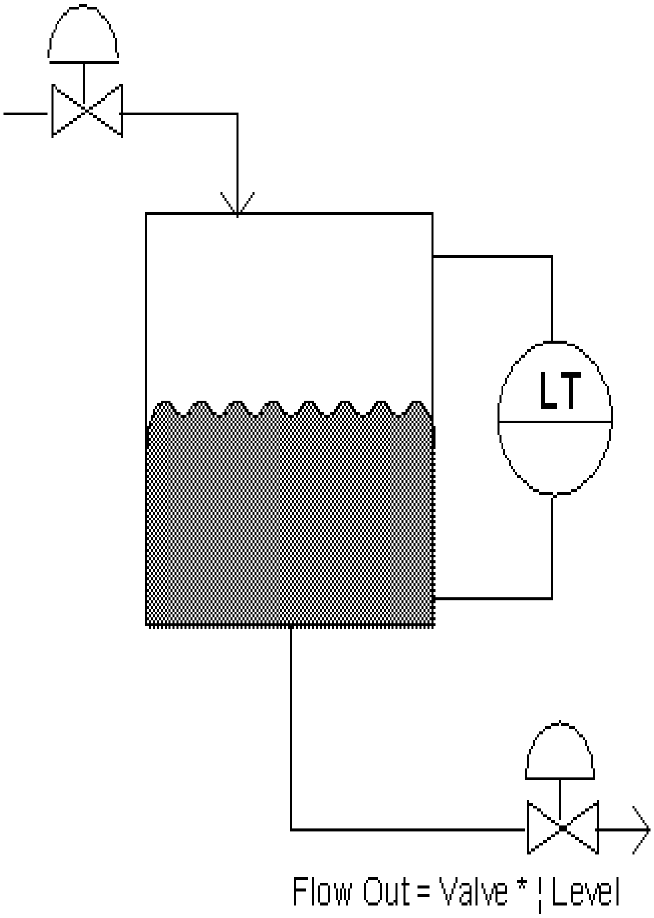
Load change, manipulating output to
hold process measurement constant

When the load changes, either the process value changes or the valve position must be changed to compensate for the load change.

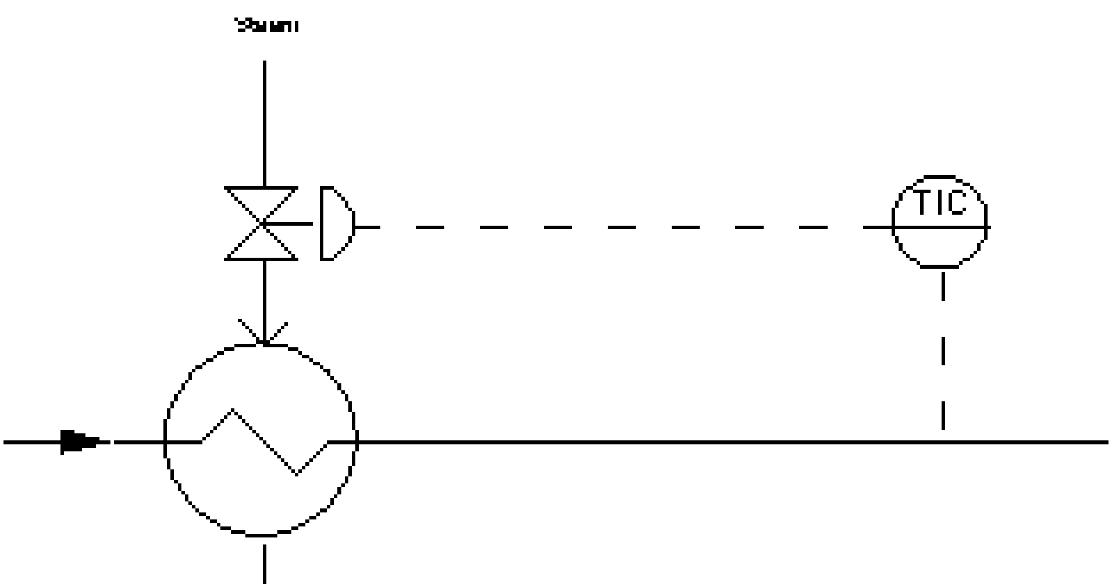
Process Dynamics: Simple lag



Process with lag only



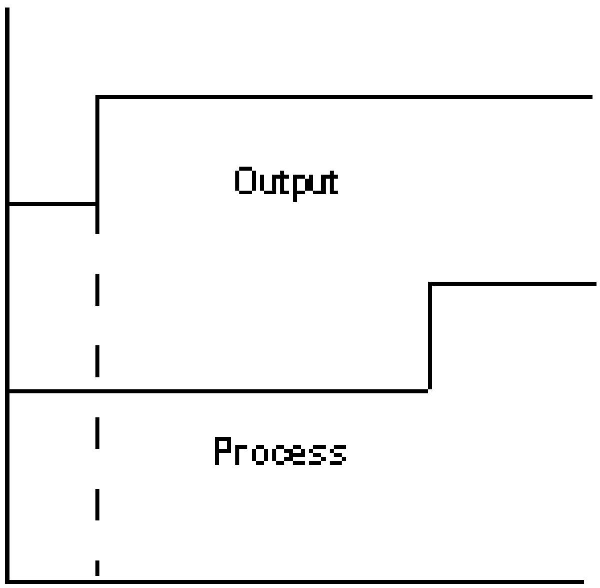
Process Dynamics: Dead time



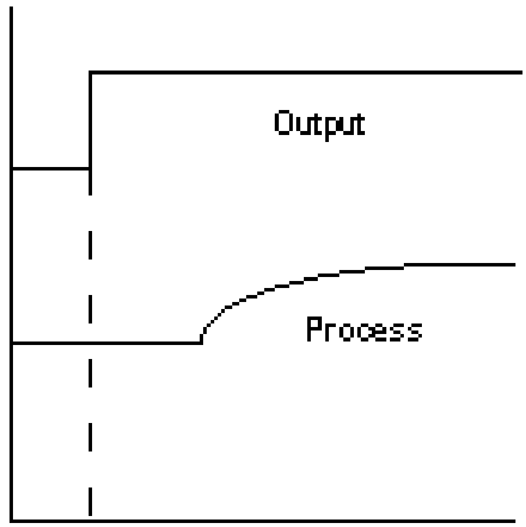
Dead Time: A delay in the loop due to the time it takes material to flow from one point to another

Also called: Distance Velocity Lag

Transportation Lag



Process with pure dead time



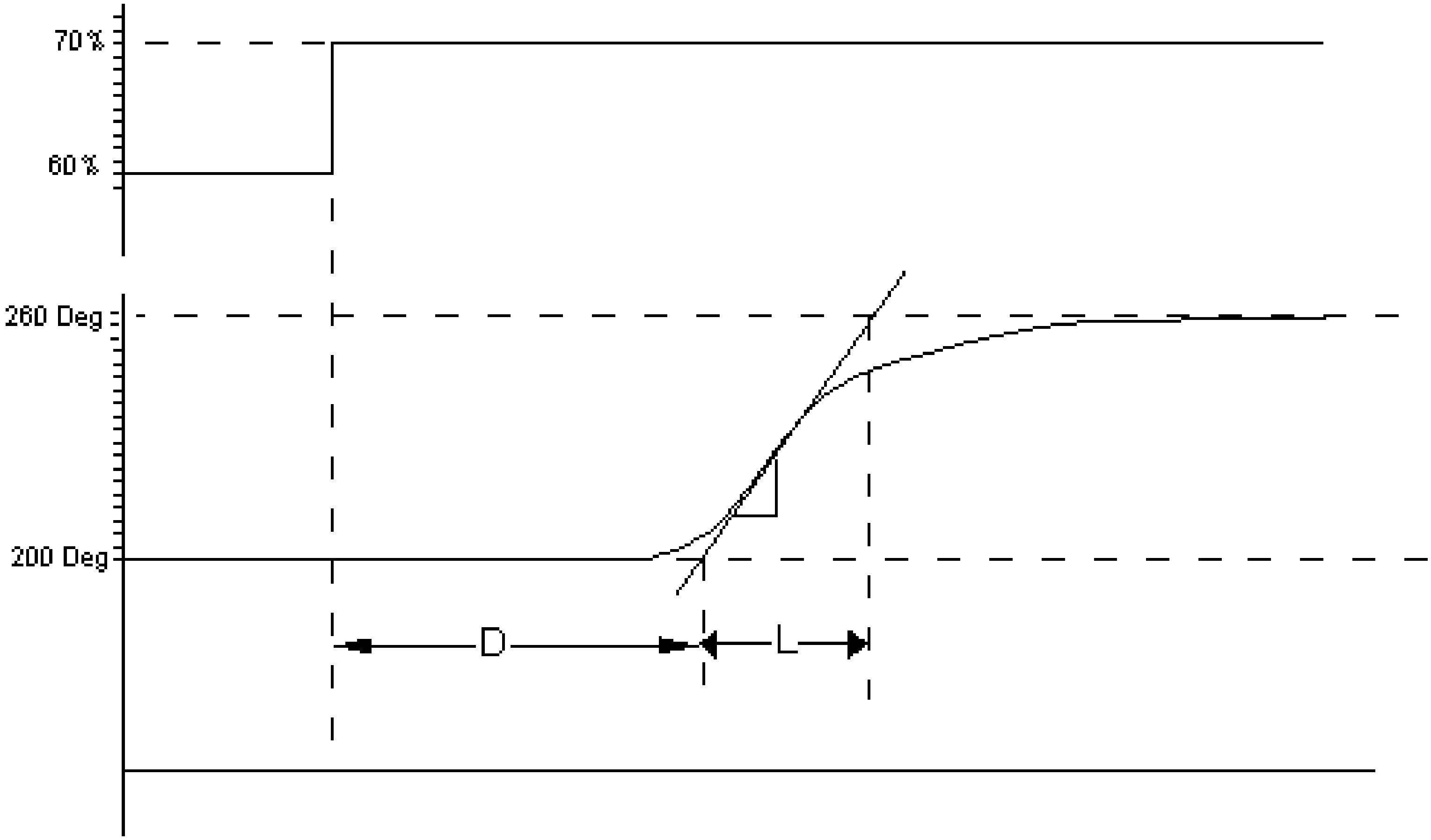
Process with lag and dead time

Most loop combine dead time and lag.

Measurement of dynamics

The dynamics differ from one loop to another.

However, they usually result in a response curve like this:



L is Lag—the largest lag in the process loop.

D is "Pseudo Deadtime"—the sum of the deadtime and all lags other than the largest lag.

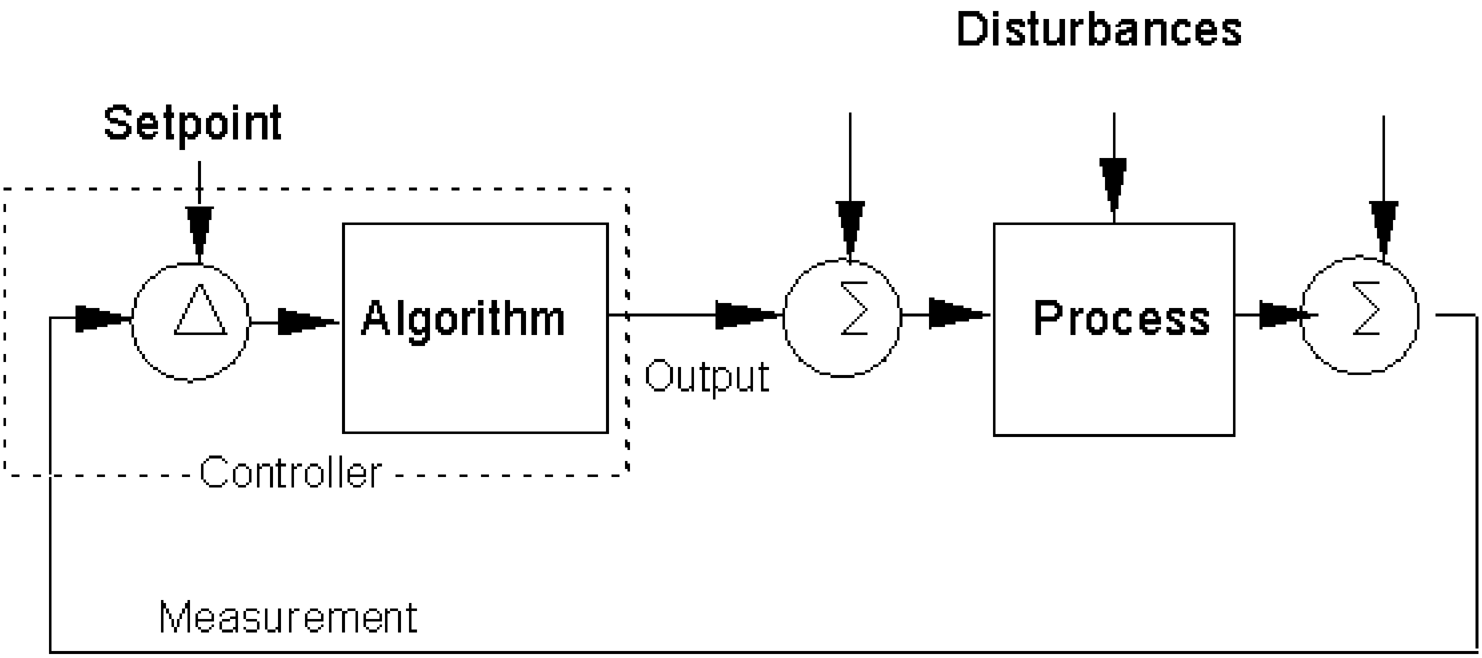
Disturbances

Almost all processes contain disturbances.

Disturbances can enter anywhere in the process.

The effect of the disturbance can depend on where it enters the loop.

Most disturbances cannot be measured.


Disturbances
Setpoint
Algorithm
Output
Process
Controller
Measurement

Chapter 3
The PID algorithm

Action

PROCESS ACTION

Defines the relationship between changes in the **valve** and changes in the **measurement**.

DIRECT **Increase** in valve position causes an **increase** in the measurement.

REVERSE **Increase** in valve position causes a **decrease** in the measurement.

CONTROLLER ACTION

Defines the relationship between changes in the **measured variable** and changes in the **controller output**.

DIRECT **Increase** in measured variable causes an **increase** in the output.

REVERSE **Increase** in measured variable causes a **decrease** in the output.

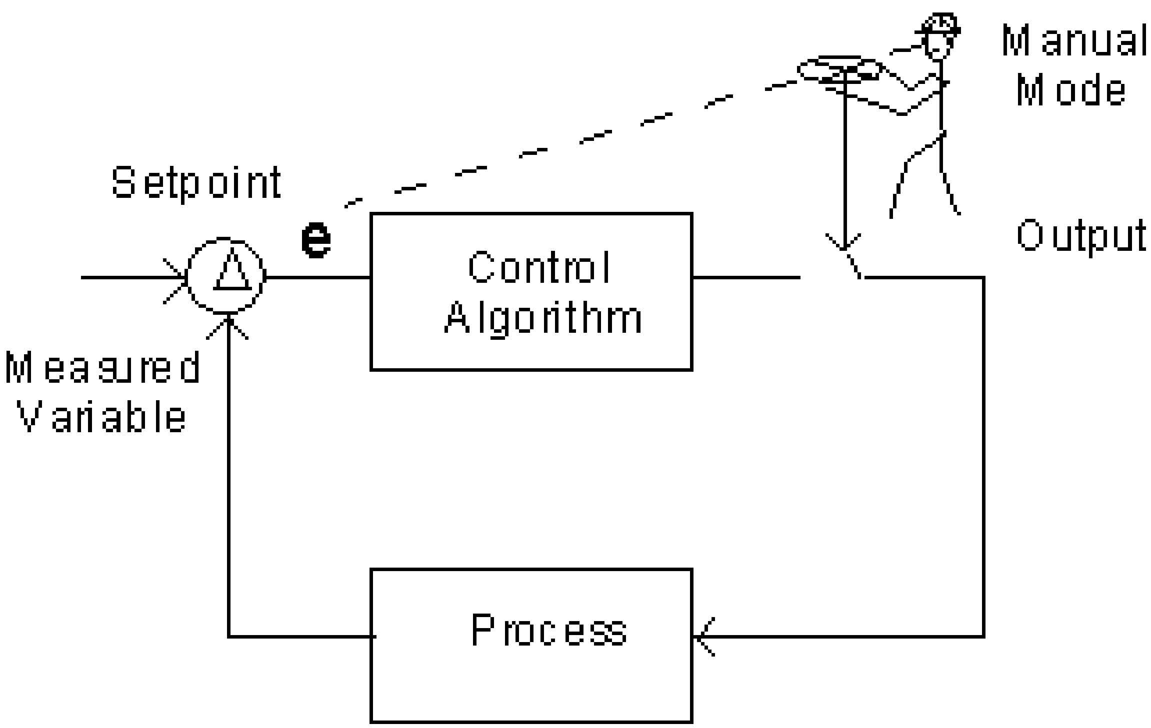
The controller action must be the **opposite** of the process action.

Auto/Manual

Manual Mode:

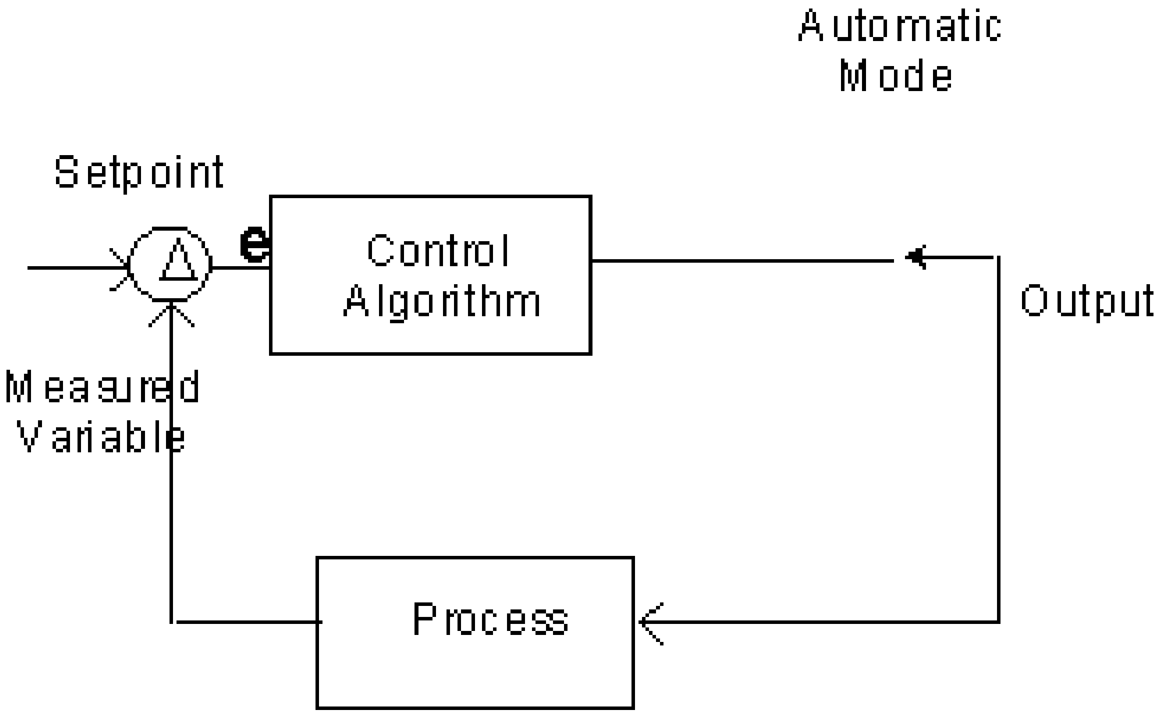
The operator adjust the output to operate the plant.

During startup, this mode is normally used.



Automatic Mode:

The control algorithm manipulates the output to hold the process measurements at their setpoints.



This should be the most common mode for normal operation.

Key concepts

The PID control algorithm does not "know" the correct output to bring the process to the setpoint.

• It merely continues to move the output in the direction which should move the process toward the setpoint.

• The algorithm must have feedback (process measurement) to perform.

The PID algorithm must be "tuned" for the particular process loop. Without

such tuning, it will not be able to function.

• To be able to tune a PID loop, each of the terms of the PID equation must be understood.

• The tuning is based on the dynamics of the process response.

The PID Control Algorithm

The PID control algorithm comprises three elements:

- **P**roportional - also known as Gain
- **I**ntegral - also known as Automatic Reset or simply Reset
- **D**erivative - also known as Rate or Pre-Act (TM of Taylor Instrument Co.)

The algorithm is normally available in several combinations of these elements:

- Proportional only
- Proportional and Integral (most common)
- Proportional, Integral, and Derivative
- Proportional and Derivative

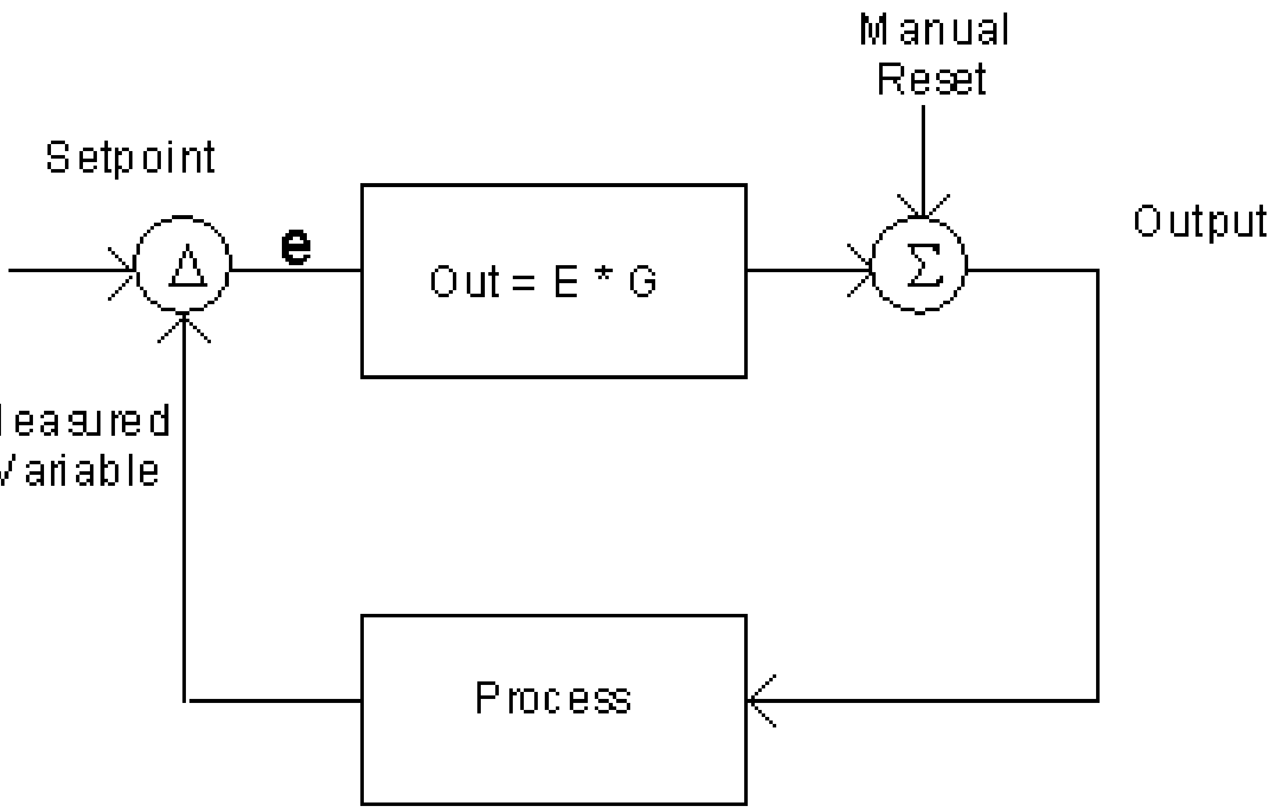
We will examine each of the three elements below:

Proportional

E = Measurement - Setpoint (direct action)

E = Setpoint - Measurement (reverse action)

Output = E * G + k



The output is equal to the error times the gain plus the manual reset.

If the manual reset stays constant, there is a fixed relationship between the setpoint, the measurement, and the output.

Proportional—units

The proportional or gain term may be calibrated in two ways:

Gain and **Proportional Band**

Gain = Output/Input

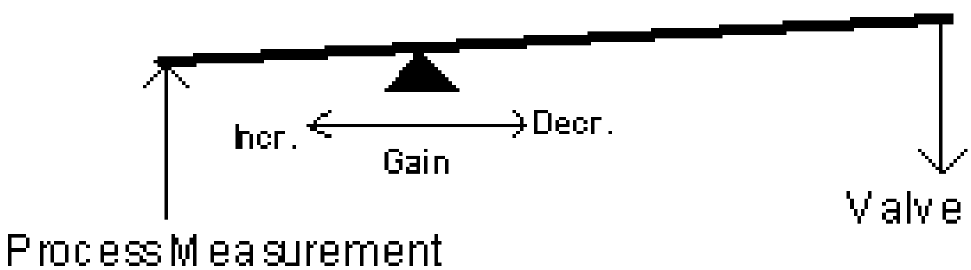
Increasing the gain will cause the output to move more.

Proportional band is the % change in the input which would result in a 100% change in the output.

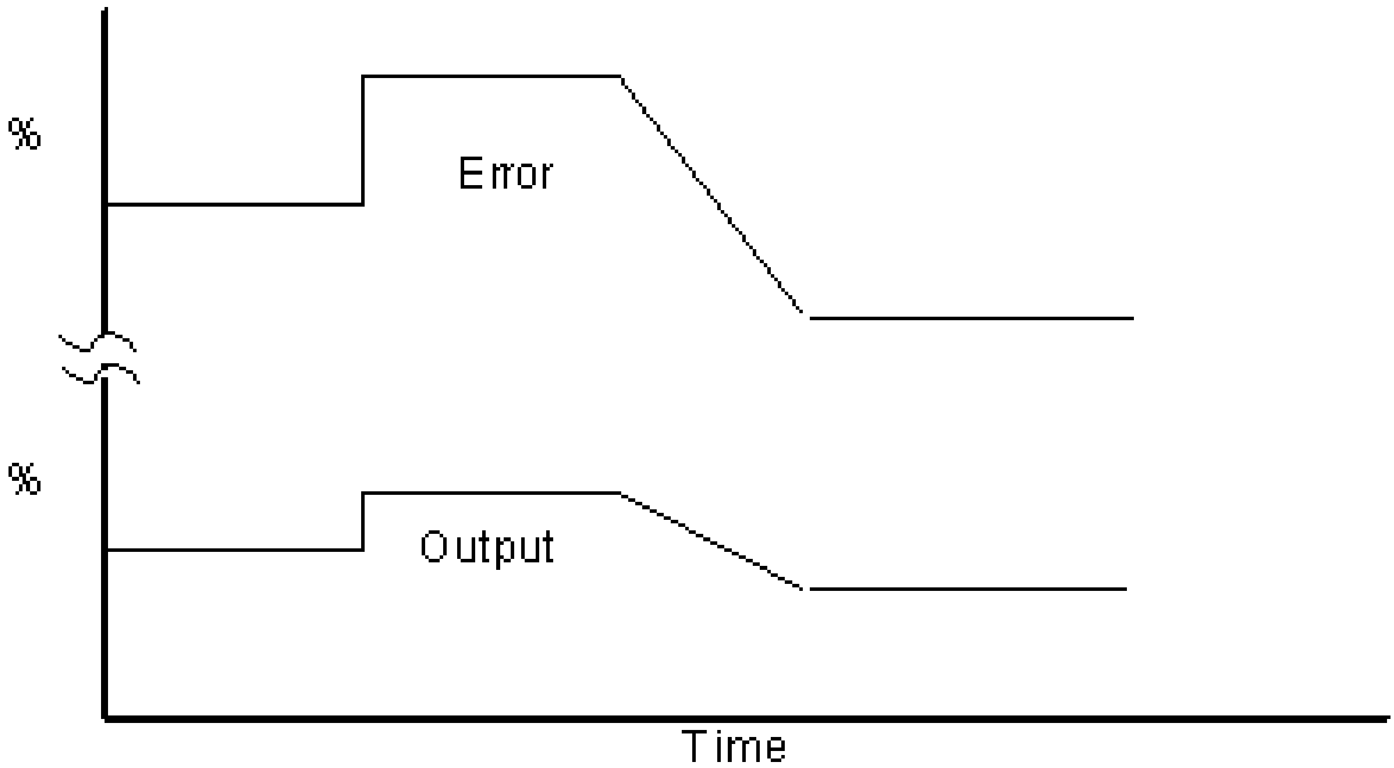
Proportional Band = 100/Gain

We will use gain in this document.

Proportional—Output vs. Measurement



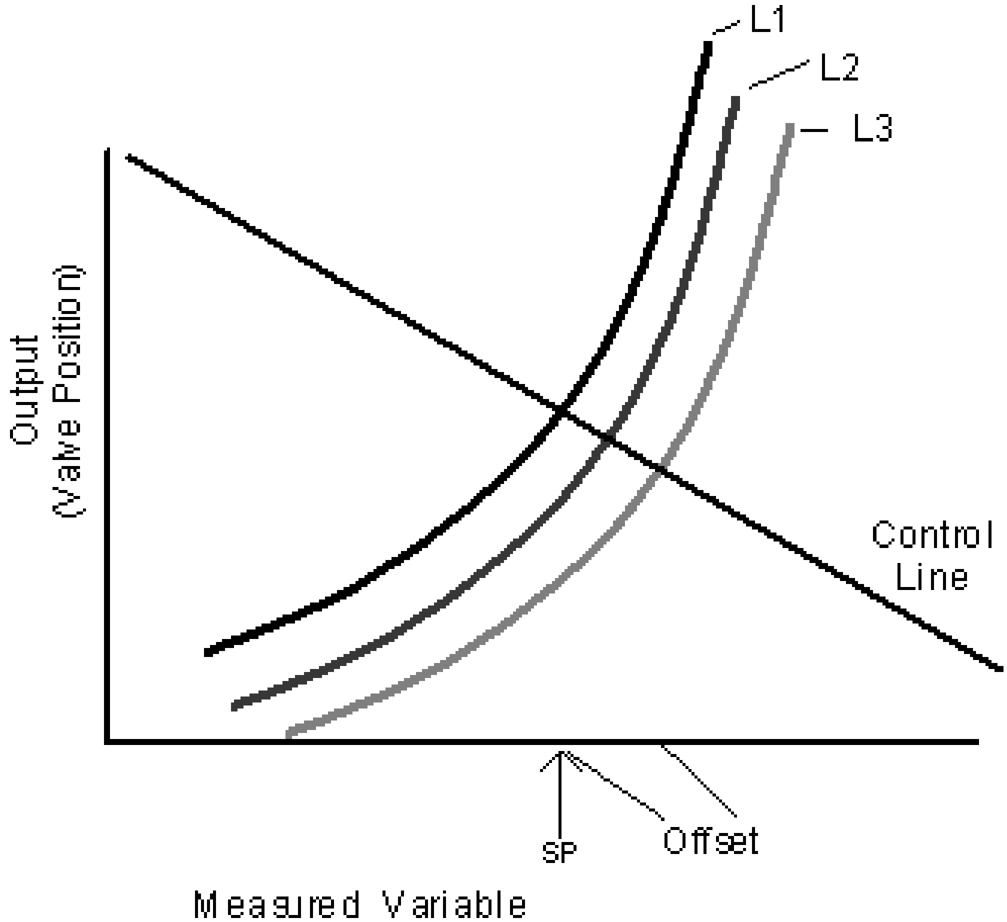
(Reverse acting)



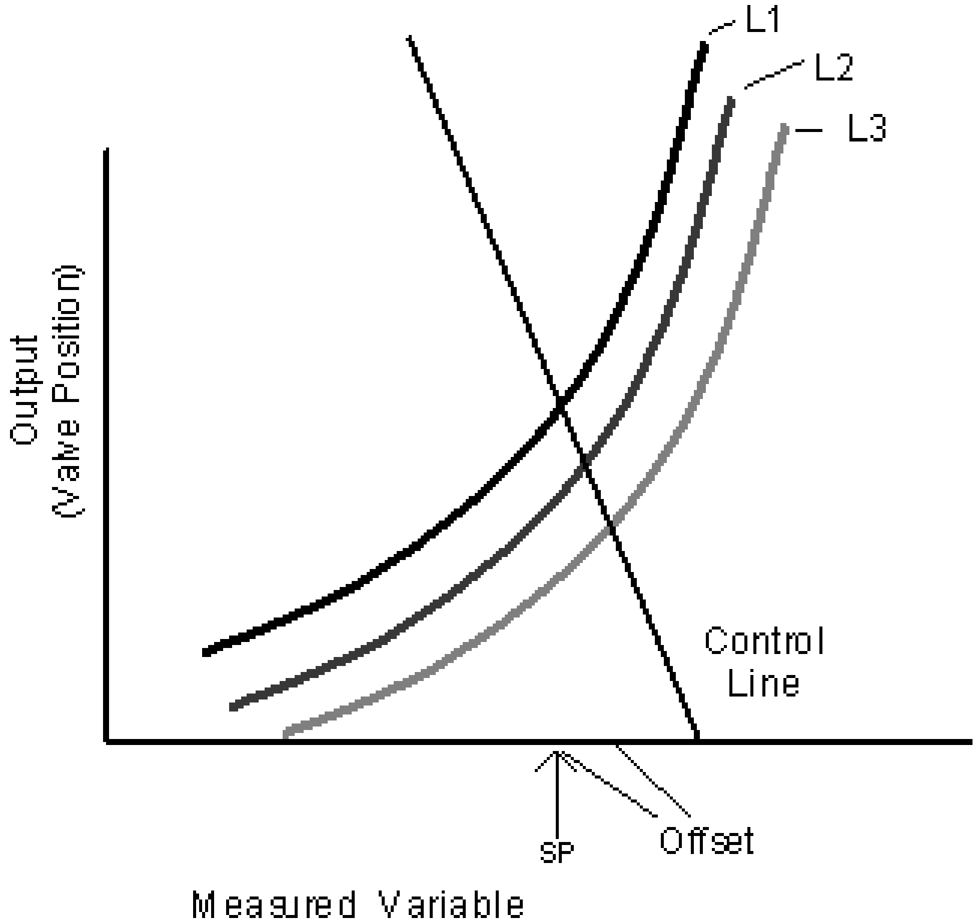
Proportional only control produces an offset. Only the adjustment of the manual reset removes the offset.

Proportional—Offset

Offset can be reduced by increasing gain.



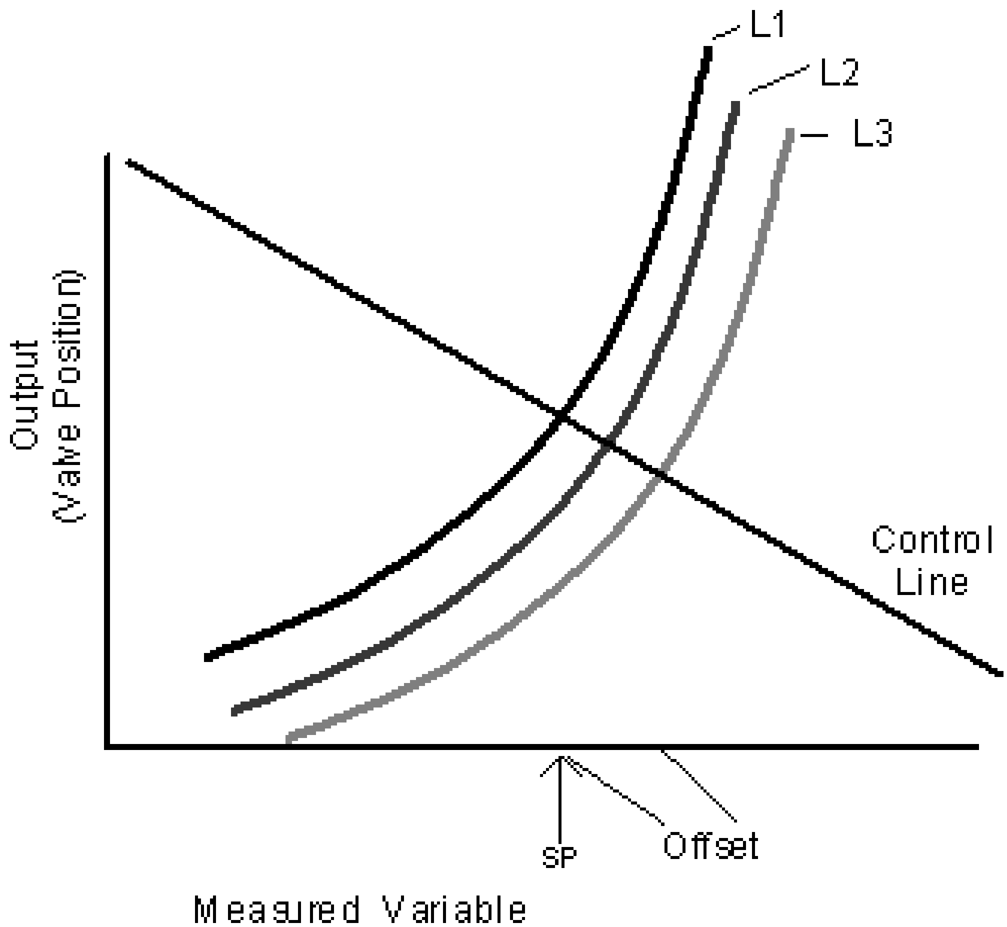
Proportional control with low gain



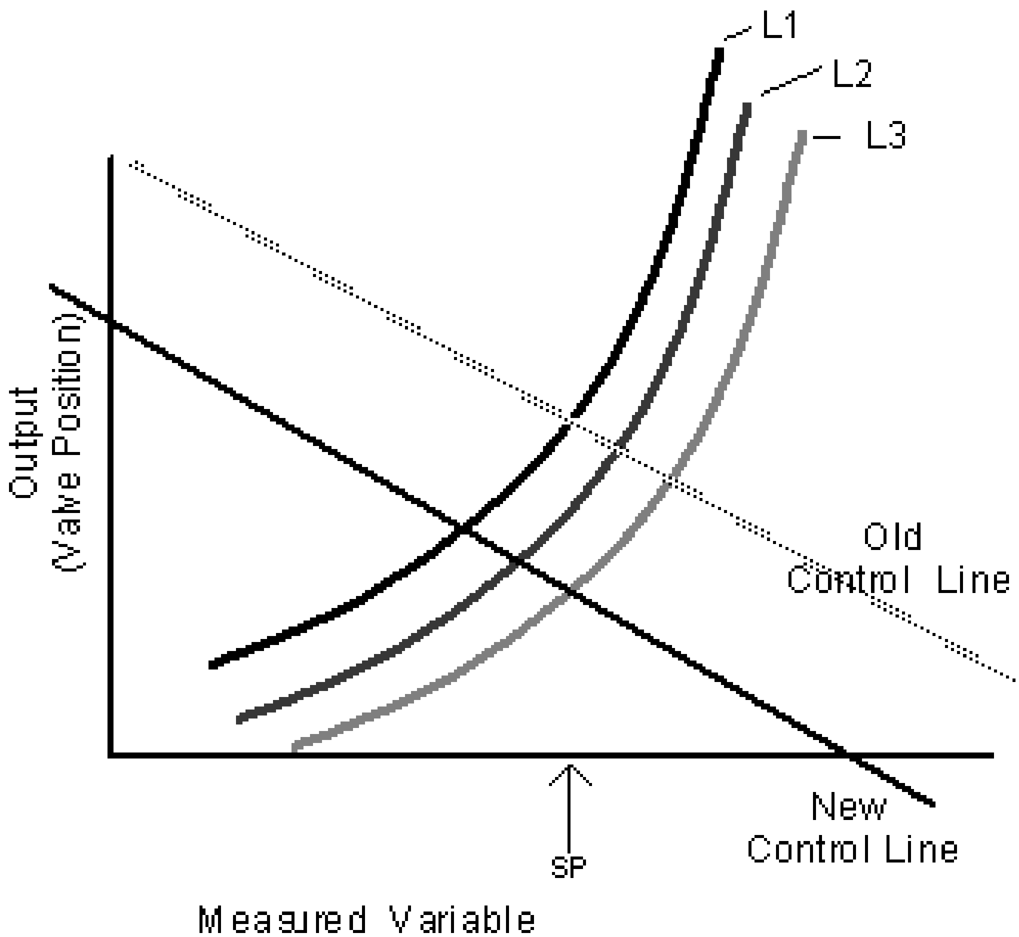
Proportional control with higher gain

Proportional—Reducing offset with manual reset

Offset can be eliminated by changing manual reset.



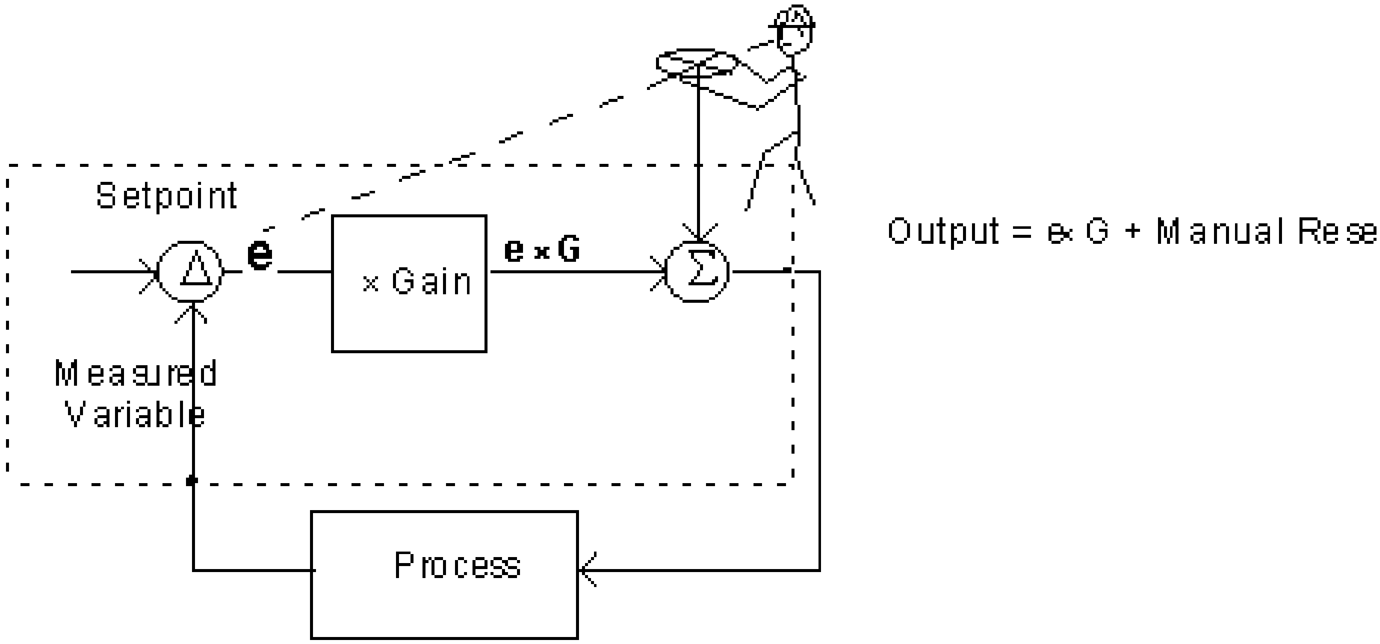
L1
L2
L3
Output
(Valve Position)
Control
Line
SP
Offset
Measured Variable



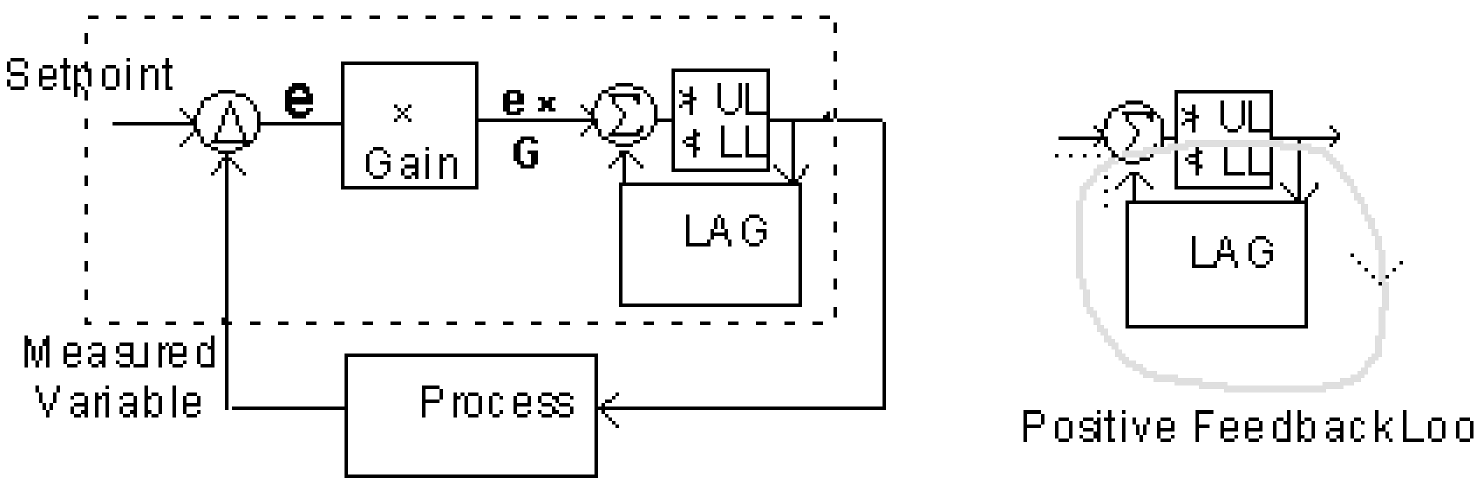
Proportional control different manual reset

Adding automatic reset

With proportional only control, the operator "**resets**" the controller (to remove offset) by adjusting the **manual reset**:



This manual reset may be replaced by **automatic reset** which continues to move the output whenever there is any error:



This is called "**Reset**" or **Integral Action**.

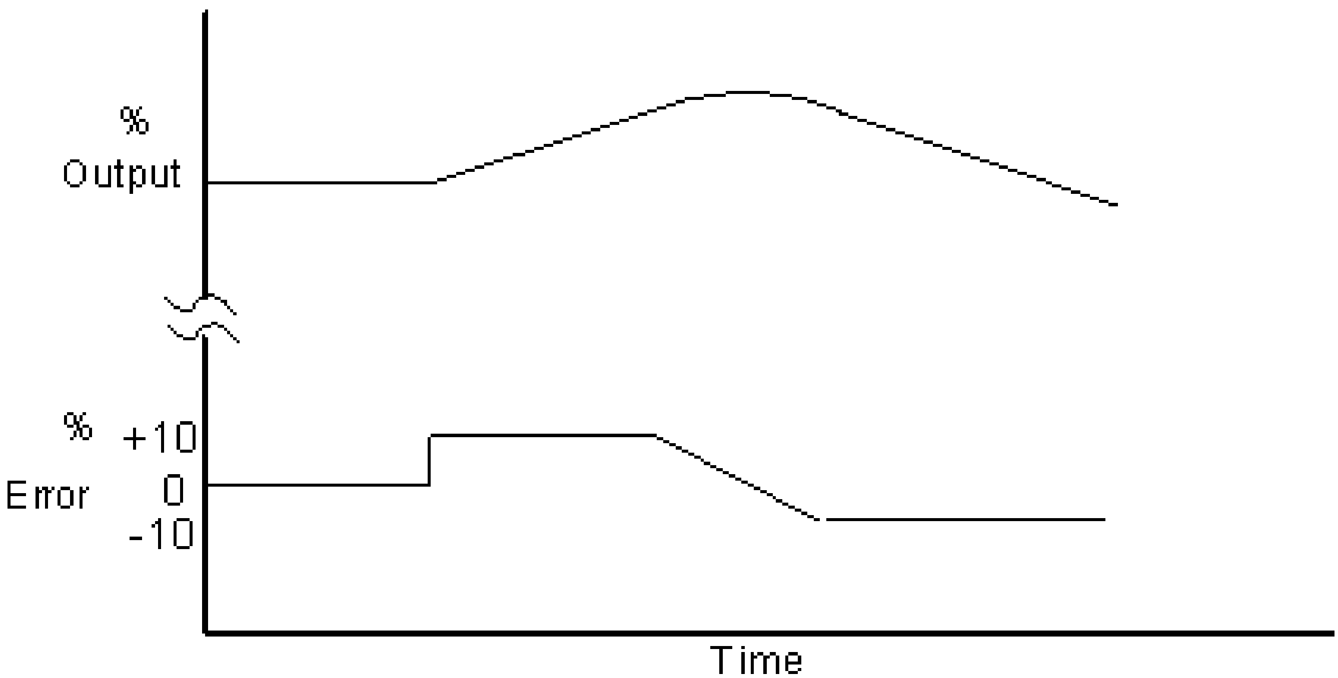
Note the use of the positive feedback loop to perform integration.

Reset or integral mode

Reset Contribution:

Out = g X Kr X integral of error

where **g** is gain, **K**$_{\mathbf{r}}$ is the reset setting in repeats per minute.

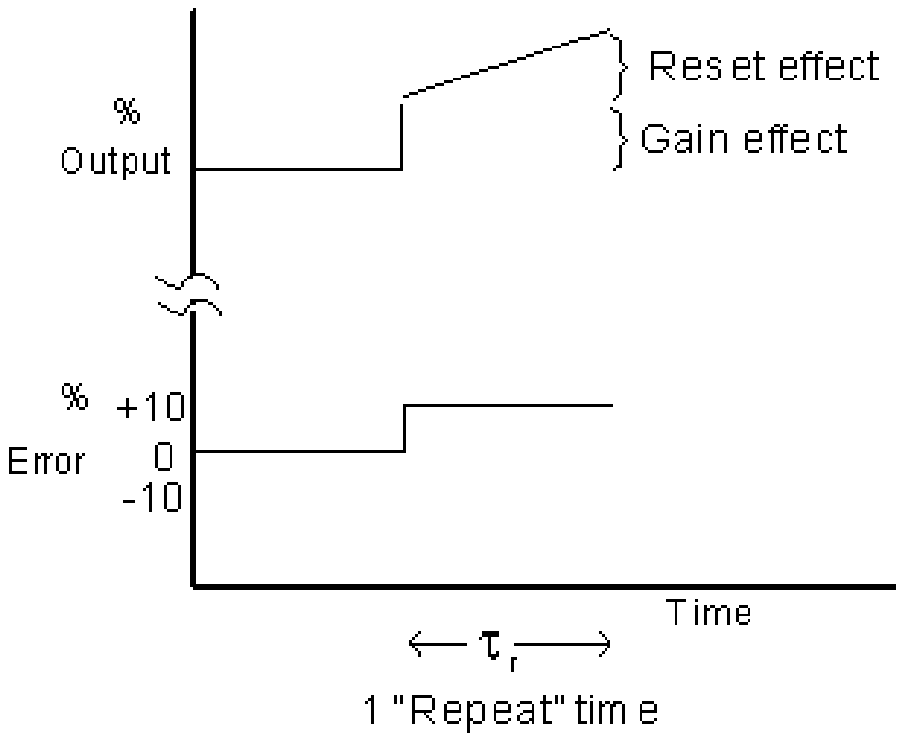


Units used to set integral or reset

Assume a controller with proportional and integral only.

Calculation of repeat time: (gain and reset terms used in controller)

With the error set to zero (measurement input = setpoint), make a change in the input and note the immediate change in output. The output will continue to change (it is integrating the error). Note the time it takes the output to, due to the integral action, repeat the initial change made by the gain action.



Some control vendors measure reset by **repeat time** in minutes. This is the time it takes the reset (or integral) element to repeat the action of the proportional element.

Others measure reset by "**repeats per minute**".

- Repeats per minute is the inverse of minutes of repeat

This document will use repeats per minute.

Derivative

First used as a part of a temperature transmitter ("Speed-Act™" - Taylor Instrument Companies) to overcome lag in transmitter measurement.

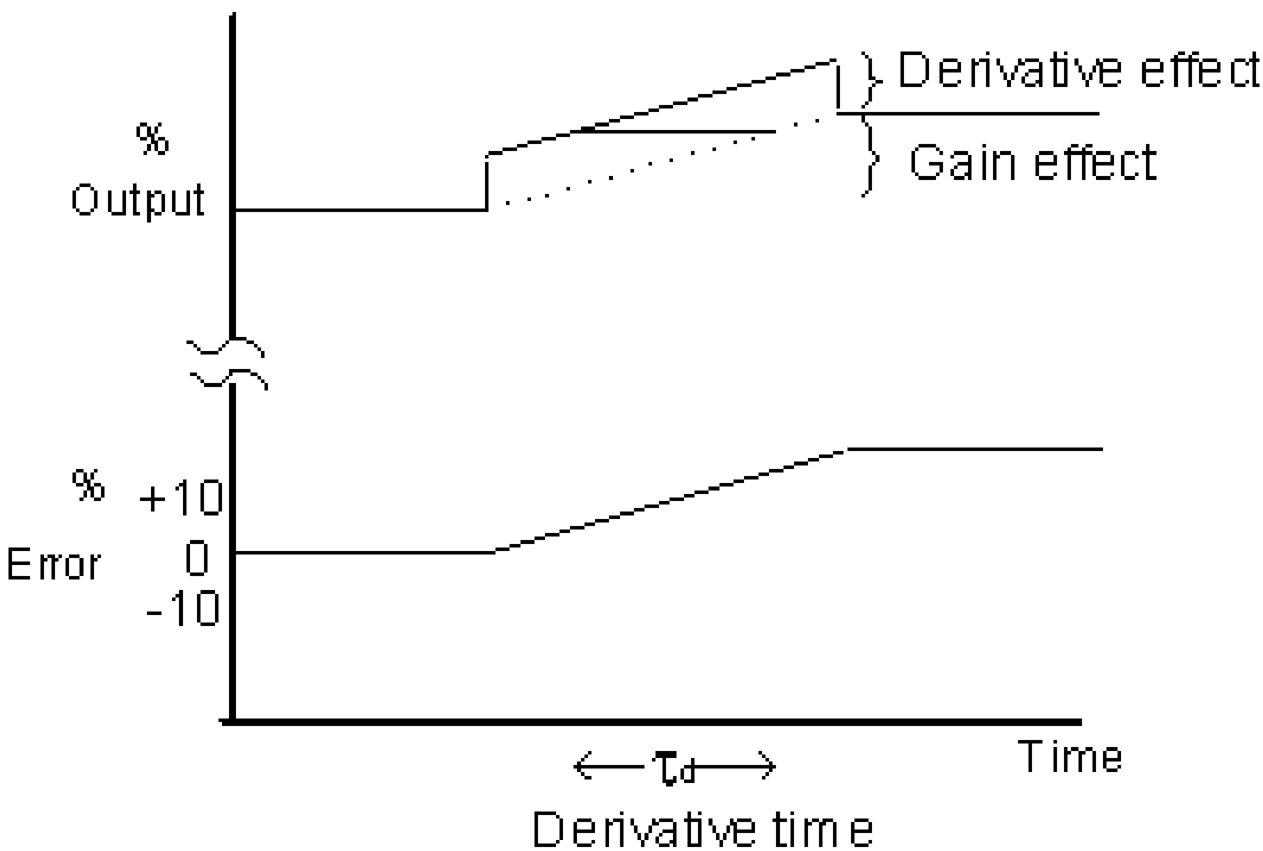
Also known as Pre-Act and Rate.

Derivative Contribution:

Out = g x **Kd** x **de/dt**

where **g** is gain, **Kd** is the derivative setting in minutes.

Response of controller with proportional and derivative:

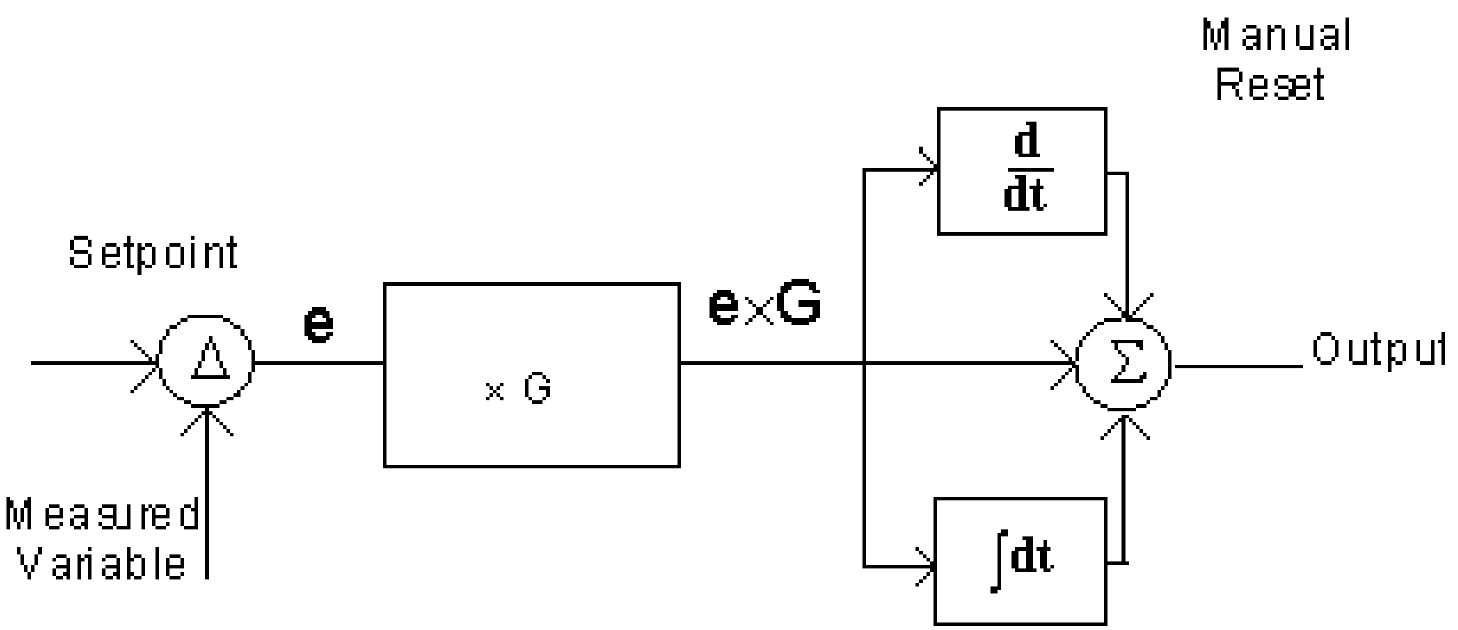


The amount of time that the derivative action advances the output is known as the "derivative time" measured in minutes.

All major vendors measure derivative (Derivative, Rate) the same.

Complete PID response

Non-Interactive (text book) form:



Out = G(e + R+ D)

Where

G = Gain

R = Reset (repeats per minute)

D = Derivative (minutes)

Note: See Interactive vs. Noninteractive (below)

Chapter 4
Additional PID Concepts

Interactive or Noninteractive algorithm

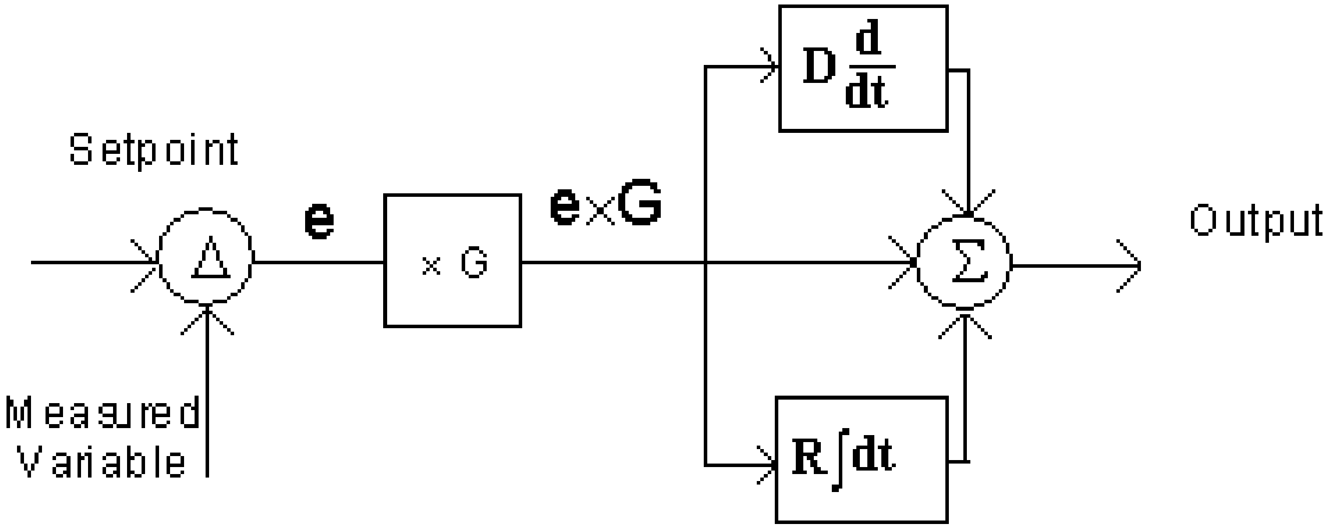
"Interactive" and "Noninteractive" refer to interaction between the reset and derivative terms. This is also known as "series" or "parallel" derivative.

Almost all analog controllers are interactive.

Many digital controllers are non-interactive, some are interactive

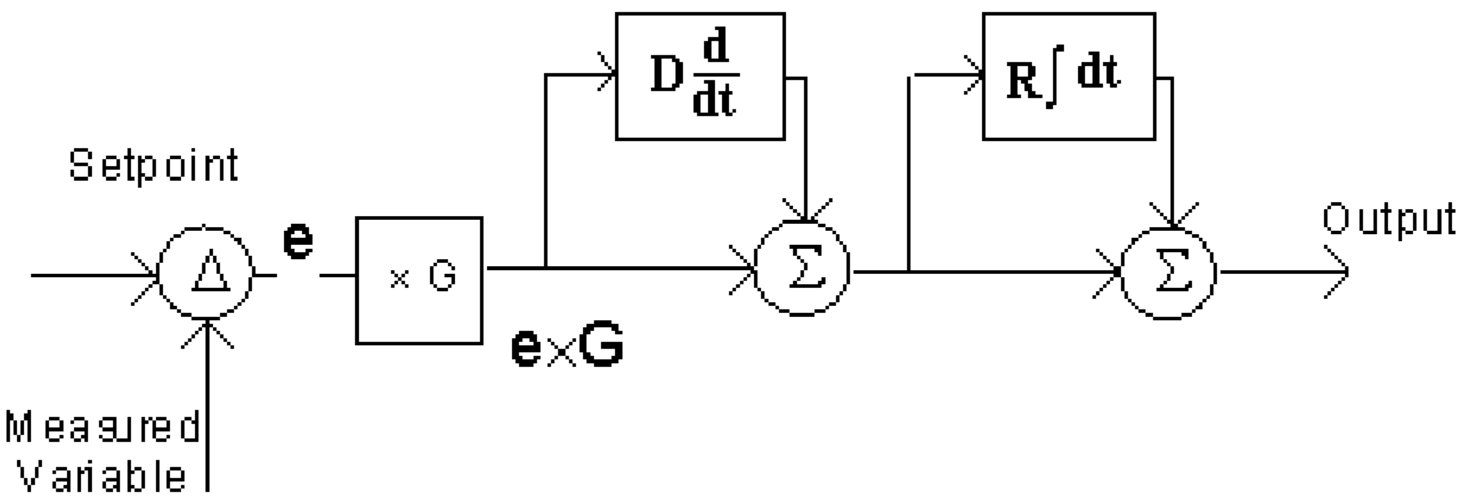
The only difference is in the tuning of controllers with derivative.

Non-Interactive (Parallel):



Out = G(e + R+ D)

Interactive (series):



Out = (RD+1)G(e + R+D)

Converting between interactive and non-interactive

Applies only to 3-mode controllers

To convert from non-interactive to interactive:

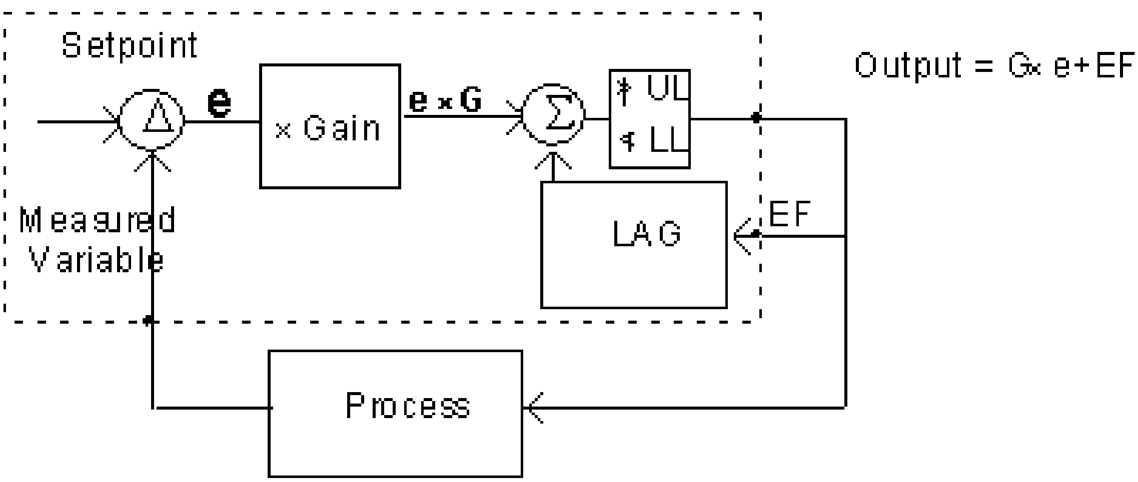
$$Gn = Gi\ (1 + Ri\ Di)$$

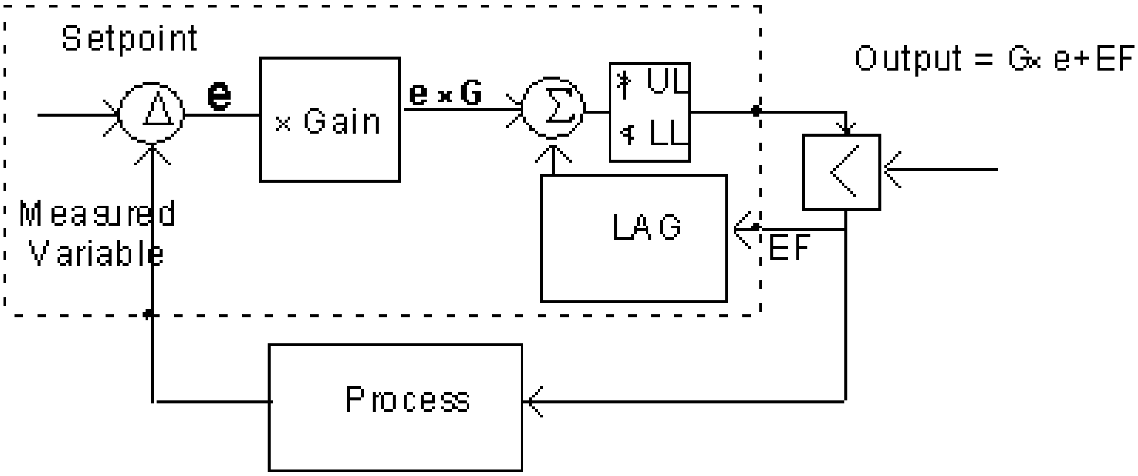
$$Rn = Ri/(1 + Ri\ Di)$$

$$Dn = Di/(1 + Ri\ Di)$$

• In other words, with a non-interactive controller the gain should be higher, the reset *rate* lower, and the derivative lower than on a commercial interactive controller.

External feedback





Saturation Properties

Another difference is in the "Saturation Properties"

eg. what happens when output has been at the upper or lower limit.

Standard algorithm

Described on previous page.

Output stays at limit until measurement crosses setpoint.

"Integrated velocity form"

Similar to equation:

Output = Last output + gain x (error - last error + reset x error)

Output pulls away from limit one reset time before measurement crosses setpoint.

• For most applications, there is no difference. For some batch startup problems, the "integrated velocity form" algorithm works best.

• Standard works best for high gain/low reset rate applications.

Chapter 5
Other Controller Features

Gain on process rather than error

In applications with high gain, a step change can result in a sudden, large movement in the valve.

• Not as severe as the derivative effect, but still can upset the process

• Solution: let gain act only on process rather than error.

Derivative on process rather than error

A step change in the setpoint results in a step change in the error.

• The derivative term acts on the rate of change of the error.

• The rate of change of a step change is very large.

• An operator step change of the setpoint would cause a very large change in the output, upsetting the process.

• Solution: let derivative act only on process rather than error.

Chapter 6
Loop Tuning

Tuning Criteria

or

"How do we know when its tuned"

Elementary methods

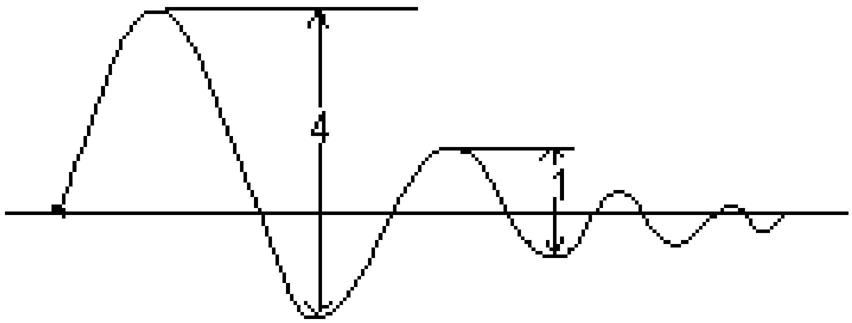
1 The plant didn’t blow up.

2 The process measurements stay close enough to the setpoint.

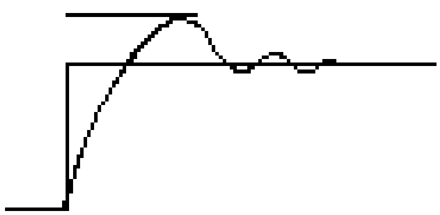
3 They say it’s OK and you can go home now.

Informal methods

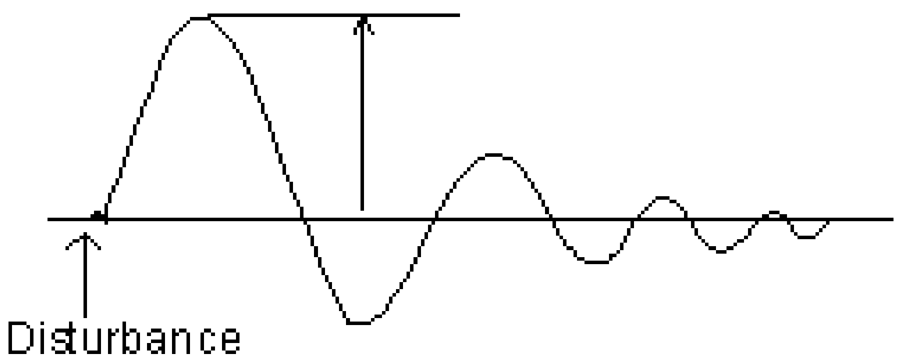
1 Optimum decay ratio (1/4 wave decay).



2 Minimum overshoot.



3 Maximum disturbance rejection.



The choice of methods depends upon the loop’s place in the process and its relationship with other loops.

Mathematical criteria

Mathematical methods—minimization of index

IAE - Integral of absolute value of error

ISE - Integral of error squared

ITAE - Integral of time times absolute value of error

ITSE - Integral of time times error squared:

• These mathematical methods are used primarily for academic purposes, together with process simulations, in the study of control algorithms.

On-line trial tuning

or

The "by-guess-and-by-golly" method

1. Enter an initial set of tuning constants from experience. A conservative setting would be a gain of 1 or less and a reset of less than 0.1.

2. Put loop in automatic with process "lined out".

3. Make step changes (about 5%) in setpoint.

4. Compare response with diagrams and adjust.

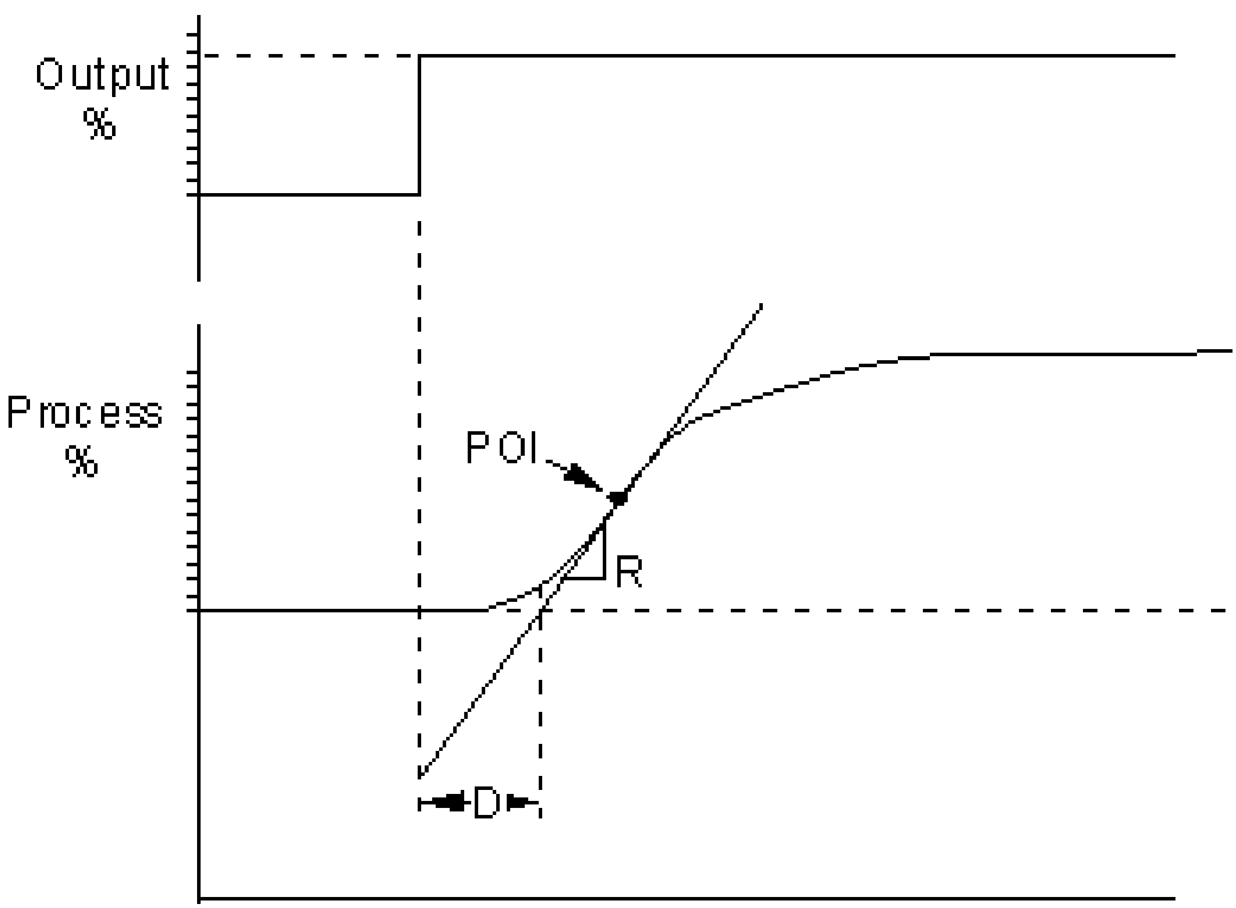
Ziegler Nichols tuning method: open loop reaction rate

Also known as the "reaction curve" method

The process must be "lined out"—not changing.

With the controller in manual, the output is changed by a small amount.

The process is monitored.



The following measurements are made from the reaction curve:

X % Change of output

R %/min. Rate of change at the point of inflection (POI)

D min. Time until the intercept of tangent line and original process value

The gain, reset, and Derivative are calculated using:

	Gain	Reset	Derivative
P	X/DR	—	—
PI	0.9X/DR	0.3/D	—
PID	1.2X/DR	0.5/D	0.5D

Ziegler Nichols tuning method: open loop point of inflection

Another means of determining parameters based on the ZN open loop.

After "bumping" the output, watch for the point of inflection and note:

T_i min Time from output change to POI

P % Process value change at POI

R %/min Rate of change at POI (Same as above method)

X % Change in output. (Same as above method)

D is calculated using the equation:

$D = T_i - P/R$

D & X are then used in the equations on the previous page.

Ziegler Nichols tuning method: open loop process gain

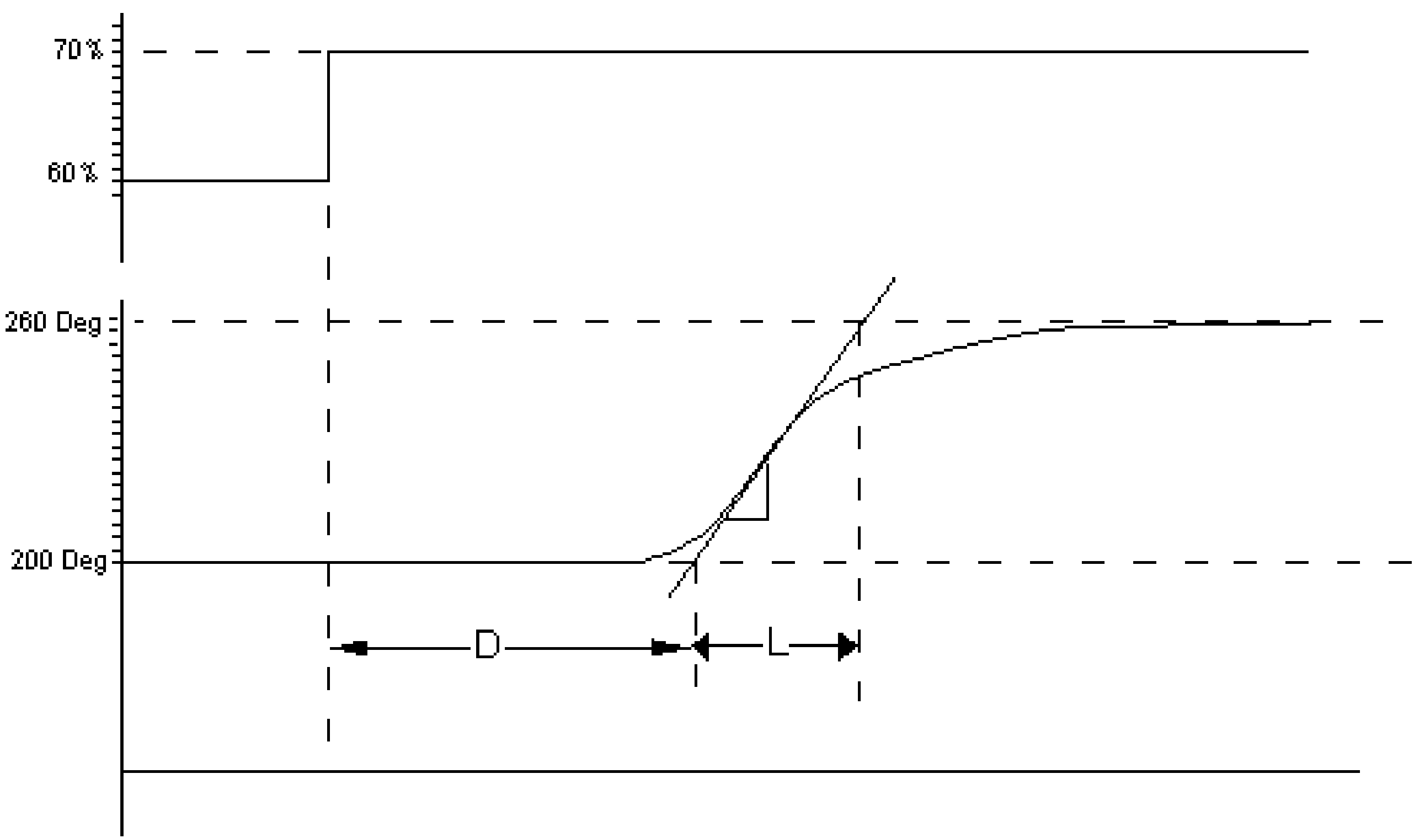
Mathematically derived from the reaction rate method.

Used only on processes that will stabilize after output step change.

The process must be "lined out"—not changing.

With the controller in manual, the output is changed by a small amount.

The process is monitored.



Gp is the process gain - the change in measured value (%) divided by the change in output (%)

The gain, reset, and Derivative are calculated using:

	Gain	Reset	Derivative
P	L/GpD	—	—
PI	0.9 L/GpD	0.3/D	—
PID	1.2 L/GpD	0.5/D	0.5D

Ziegler Nichols tuning method: closed loop

Steps

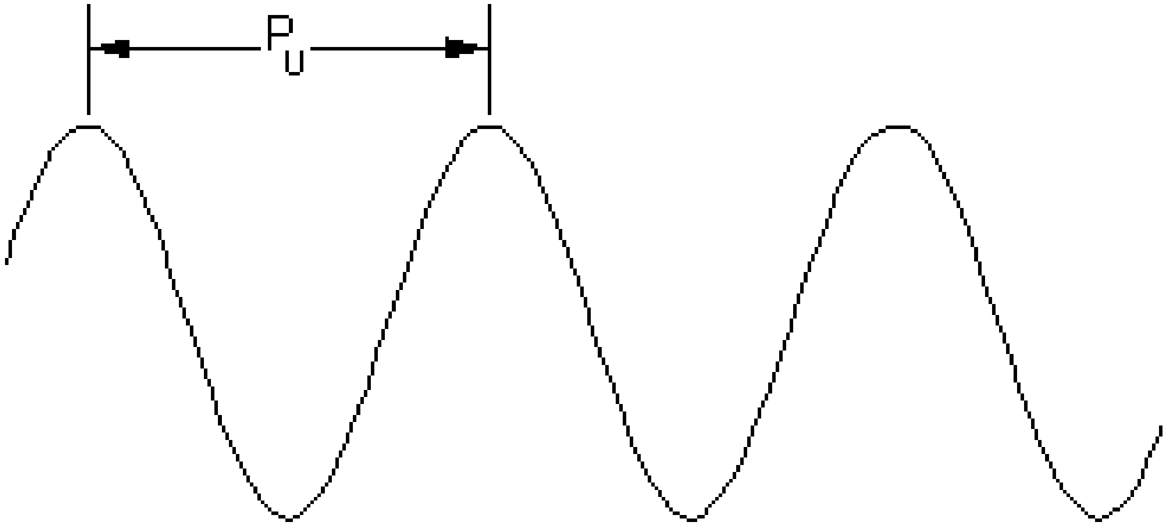
Place controller into automatic with low gain, no reset or derivative.

Gradually increase gain, making small changes in the setpoint, until oscillations start.

Adjust gain to make the oscillations continue with a constant amplitude.

Note the gain (Ultimate Gain, Gu,) and Period (Ultimate Period, Pu.)

The Ultimate Gain, Gu, is the gain at which the oscillations continue with a constant amplitude.



The gain, reset, and Derivative are calculated using:

	Gain	Reset	Derivative
P	0.5 Gu	—	—
PI	0.45 Gu	1.2/Pu	—
PID	0.6 Gu	2/Pu	Pu/8

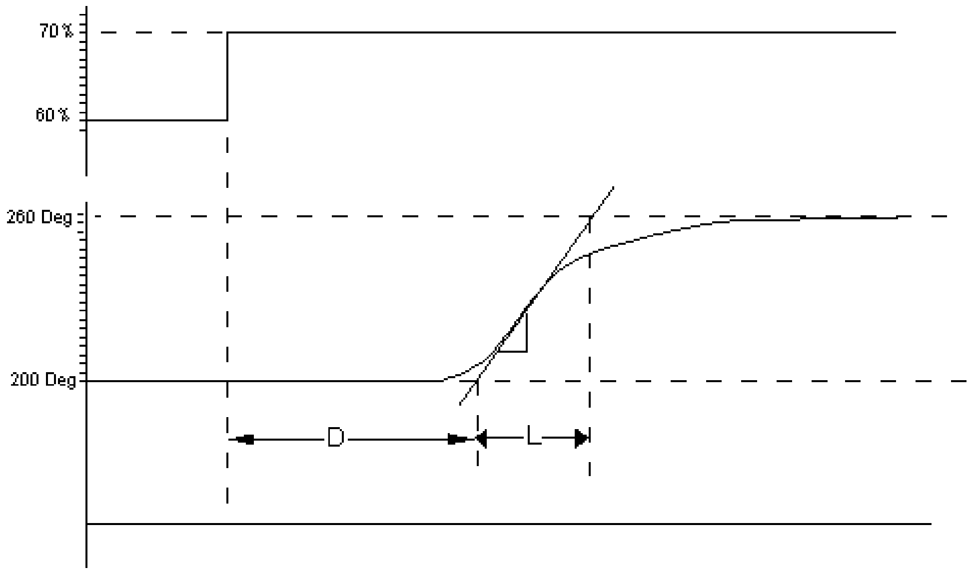
Controllability of processes

The "controllability" of a process is depends upon the gain which can be used.

The higher the gain:

• the greater rejection of disturbance and

• the greater the response to setpoint changes.



The **predominate lag** L is based on the largest lag in the system.

The **subordinate lag** D is based on the deadtime and all other lags.

The maximum gain which can be used depends upon the ratio .

From this we can draw two conclusions:

• Decreasing the dead time increases the maximum gain and the controllability.

• Increasing the ratio of the **longest** to the **second longest** lag also increases the controllability.

Flow loops

Flow loops are too fast to use the standard methods of analysis and tuning.

Analog vs. Digital control:

- Some flow loops using analog controllers are tuned with high gain.
- This will not work with digital control.

With an analog controller, the flow loop has a predominate lag (L) of a few seconds and no subordinate lag.

With a digital controller, the scan rate of the controller can be considered dead time.

Although this dead time is small, it is large enough when compared to L to force a low gain.

Typical digital flow loop tuning: Gain= 0.5 to 0.7

Reset=15 to 20 repeats/min..

no derivative.

The PID control algorithm

The following is a brief description of the standard PID control algorithm used in most controllers.

Proportional Contol (gain)

The first element of PID control to be developed is Proportional control. The equation is simple:

error = measurement - setpoint (direct action)

or

error = setpoint - measurement (reverse action)

Note the action may be either direct or reverse. In a direct acting control loop an increase in the process measurement causes an increase in the ouput to the final control element.

The proportional only equation is:

output = gain x error + bias

The bias is sometimes known as the manual reset. Some control systems (such as Foxboro products, use proportional band rather than gain. The proportional band and the gain are related by:

$$\text{Gain} = \frac{100\%}{\text{Proportional Band}}$$

$$\text{Proportional Band} = \frac{100\%}{\text{Gain}}$$

Gain is the ratio of the change in the output to the change in the input.

$$\text{Gain} = \frac{\text{Output change}}{\text{Input change}}$$

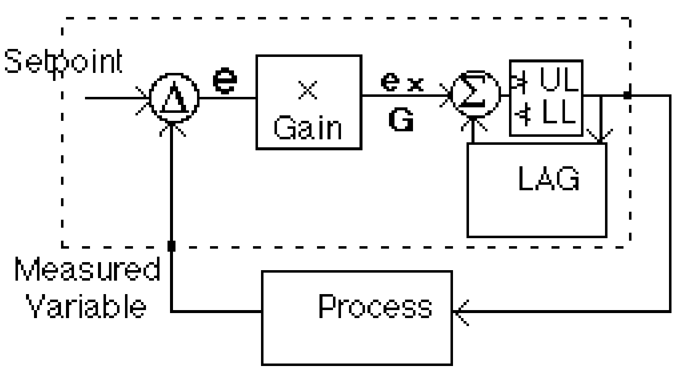
Proportional band is the amount the input would have to change in order to cause the output to move from 0 to 100% (or vice versa)

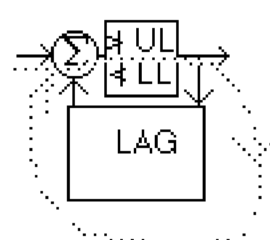
With proportional only control the controller will not bring the process measurement to the setpoint with out a manual adjustment to the bias (or manual reset) term of the equation. In the early days of control the operator, upon observing an offset in the control loop would correct the offset by manually "reseting" the controller (adjusting the bias).

Integral Control (automatic reset)

Rather than to require that the operator "manually reset" the control loop whenever there was a load change control functions were developed to "automatically reset" the controller by adjusting the bias term when ever there was an error. This "automatic reset" is also known simply as "reset" or as "integral".

The most common way to implement integral mode in analog controllers is to use a positive feedback into the output.





Positive Feedback Loop

The equation for PI control is:

$$Out = g \times K_r \times \int e \, dt$$

out = gain x (error + integral(error)dt)

The amount of reset used is measured in terms of "reset time" in minutes or its inverse, "reset rate" in repeats per minute. The following test can be perfored on a controller which is not connected to the process:

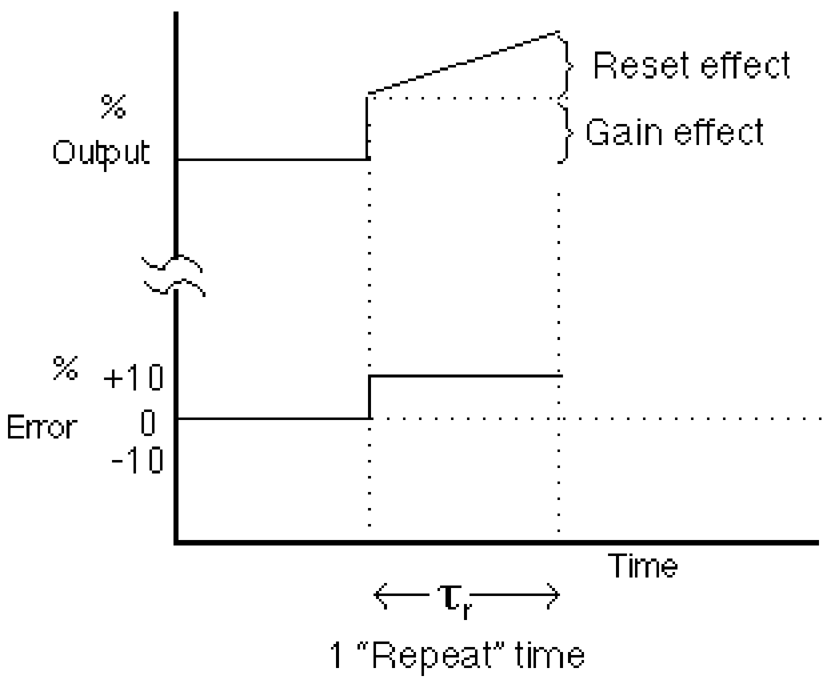
1. an adjustable signal is connected to the input.

2. the output is indicated or recorded.

3. wtih the controller manual the setpoint and the input are set to the same value.

4. the controller is switched to automatic. Becuase the error is zero, the output does not change.

5. The input to the controller is changed by a small amount. The output will move suddenly due to the gain. The output will continue to change at a constant rate. The time is measured from the time of the intitial change until the time that the instant change is repeated by the constant movement. The repeat time, or reset time, is the time it takes for the reset effect to repeat (or move the output the same amount as) the gain effect. Its inverse is reset rate, measured in repeats per minute.



Derivative Control (Pre-Act tm or Rate)

The third term of PID control is derivative, also known as Pre-Act (trade mark of Taylor Instrument Companies (now ABB Kent Taylor), and rate.

The derivative term looks at the rate of change of the input and adjusts the output based on the rate of change. The derivative function can either use the time derivative of the error, which would include changes in the setpoint, or of the measurement only, excluding setpoint changes.

The equation for the derivative contribution (assuming derivative on error) is:

$$\text{Out} = g \times K_d \times \backslash\text{F(de,}$$

The amount of derivative used is measured in minutes of derivative. To illustrate the meaning of minutes of derivative, consider the following open loop test:

1. Connect a signal generator with a ramp cability to the input of a controller. The controller output is connected to a recorder. Configure the controller with some gain, no reset, and no derivative.

2. With a constant output from the signal generator and the controller in manual, adjust the setpoint to be equal to the input from the signal generator.

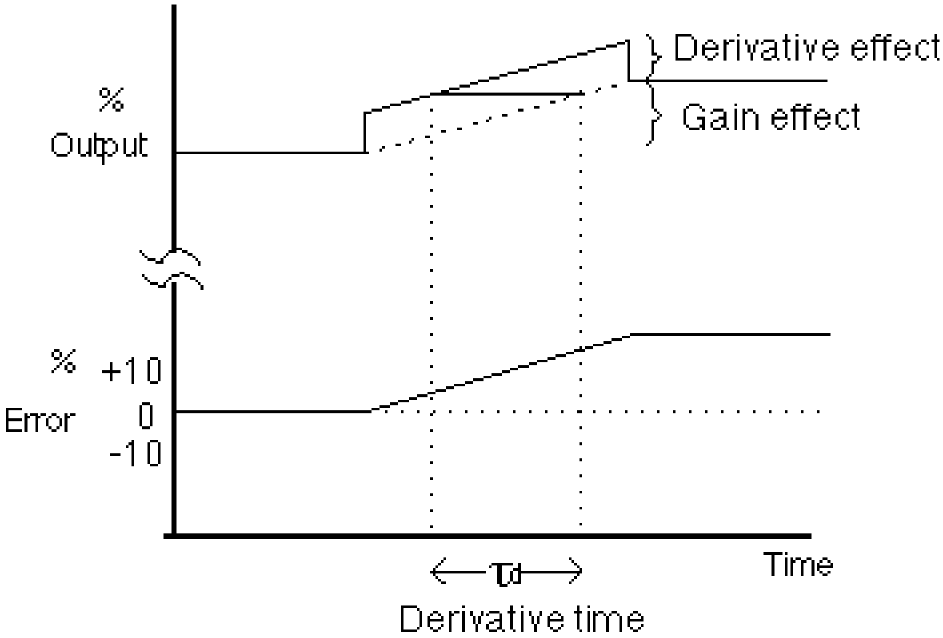
3. Place the controller into automatic mode.

4. Start the ramp.

5. Later stop the ramp.

6. Repeat the above steps with some derivative. Compare the trend records of the controller's input and output.

On the following trend record



note that when the ramp is started, with no derivative (dashed line) the output ramps up due to the change in input and the gain. Using derivative (solid line) the output jumps up, rises in a ramp, then jumps down. The difference ***in time*** between the solid line and the dashed line represents the amount of derivative, in units of time (usually minutes).

Putting it together: PID control

Combining the three elements, gain, integral, and derivative, we have the equation:

$$\mathbf{Out} = \mathbf{G}\left(\mathbf{e} + \mathbf{R}\int \mathbf{e}\,\mathbf{dt} + \mathbf{D}\frac{\mathbf{de}}{\mathbf{dt}}\right)$$

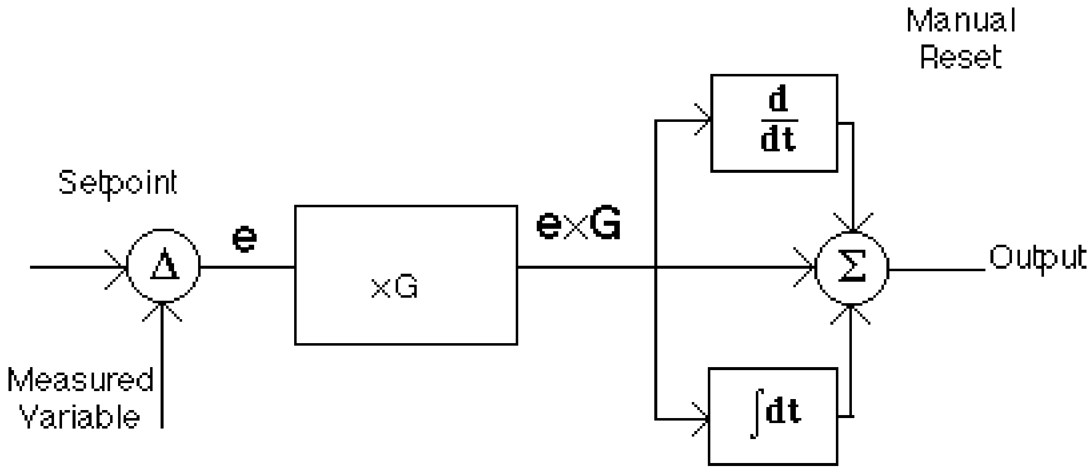
Where

G = Gain

R = Reset (repeats per minute)

D = Derivative (minutes)

Shown graphically:



Note that in the equation the gain is multiplied by all three terms. This is important for the PID equation to be able to be tuned by any of the standard tuning methods.

The PID Algorithm

There is no single PID algorithm. Different fields using feedback control have probably used different algorithms ever since math was introduced to feedback control. This Web page (a single file, of four pages, no pictures) is written for people in the process industries, for that is the only field in which I (David W. St. Clair) have experience. Even in that single field, which has been served by companies such as ABB (formerly Taylor), Bailey, Fisher, Foxboro, Honeywell, Moore Products, Yokogawa and others there is no standard algorithm. Perhaps years ago there was (or for most practical purposes was), but today there are many algorithms. Also there is no standard terminology. For the person interested in tuning controllers for the process industries it has become a bit more complicated, because the rules and procedures you would use to tune with one algorithm are not the ones you would use to tune with another. Also, with the added features available with computers, some of the configurations can become quite complex. This page does not begin to address those, but you certainly need to understand what your basic building block is.

The purpose of this Web page is to focus on the fact that there are differences and to describe them (or at let to alert you to look for them). No reference to the algorithm of specific manufacturers is given. If you are tuning controllers you must know the algorithm of the equipment you are using. For that you should read the information provided by the manufacturer. Even the words used to identify an algorithm are ambiguous. You should look at the equation. This is unfortunate because many persons assigned the responsibility of tuning process industry controllers are not comfortable with equations. If you are reading this as preparation for writing a PID algorithm, it will alert you to the fact that there is more to it than you might have thought. Indeed, the feedback I have from knowledgeable people is that even the experts can slip up.

I have asked several friends and acquaintances to review this write-up before (and after) putting it on the Web. This does not necessarily mean they agree with what I have written (some discussions are still taking place), but at least I have sought their advice. I hope it has no mistakes in it. If you feel you have something to contribute in the way of corrections or additions, please write me. I have nothing to sell by providing this page, except better control and hopefully less confusion.

Presently there are three basic forms of the PID algorithm. These will be discussed in turn. After that there is a short discussion of other aspects of any algorithm which must be considered to write the digital program for one. A section on references and links is at the end.

Expressed by their Laplace transforms the three forms are:

First form: Kc(1 + 1/Tis)(1 + Tds)/(1 + Tds/Kd)
Second form: Kc'(1 + 1/Ti's + Td's)

Third form: Kc" + 1/Ti"s+Td"s

where

Kc, Kc' and Kc" relate to the P part of PID
Ti, Ti' and Ti" relate to the I part of PID
Td, Td' and Td" relate to the D part of PID
s is the Laplace notation for derivative with respect to time
Kd is the derivative gain

I have deliberately not assigned a name to any of these forms yet. Also I have not given a name to the variables. Both will come later as each algorithm is discussed. The second and third forms can be made equivalent to the first form (provided derivative is handled appropriately), but the first form cannot duplicate all combinations available in the second and third forms. The second and third forms can be made equal to each other. For most practical purposes one algorithm is not better than another, just different.

THE FIRST FORM OF THE PID ALGORITHM

This first form is called "series" or "interacting" or "analog" or "classical". The variables are:

Kc = controller gain = 100/proportional band
Ti = Integral or reset time = 1/reset rate in repeats/time
Td = derivative time
Kd = derivative gain

Early pneumatic controllers were probably designed more to meet mechanical and patent constraints than by a zeal to achieve a certain algorithm. Later pneumatic controllers tended to have an algorithm close to this first form. Electronic controllers of major vendors tended to use this algorithm. It is what process industry control users were used to at the time. If you are unsure what algorithm is being used for the controller you are tuning, find out what it is before you start to tune.

I did not follow closely the evolution of algorithms as digital controllers were introduced. It is my understanding that most major vendors of digital controllers provide this algorithm as basic, and many provide the second form as well. Also, many provide several variations (I'm told Allen-Bradley has 10, and that other manufacturers are adding variations continually).

The choice of the word interacting is interesting. At least one author says that it is interacting in the time domain and noninteracting in the frequency domain. Another author disagrees with this distinction. This really becomes a discussion of what interacts with what. To be safe, think of the word interacting as one to *identify* the algorithm, not to *describe* it.

SECOND FORM OF THE PID ALGORITHM

The second form of the algorithm is called "noninteracting, or "parallel" or "ideal" or "ISA" . I understand one manufacturer refers to this as "interacting", which serves to illustrate that terms by themselves may not tell you what the algorithm is. This form is used in most textbooks, I understand. I think it is unfortunate that textbooks do not at least recognize the different forms. Most if not all books written for industry users rather than students recognize at least the first two forms. The basic difference between the first and second forms is in the way derivative is handled. **If the derivative term is set to zero, then the two algorithms are identical.** Since derivative is not used very often (and shouldn't be used very often) perhaps it is not important to focus on the difference. But it is important to anyone using derivative, and people who use derivative should know what they are doing. The parameters set in this form can be made equivalent (except for the treatment of gain-limiting on derivative) to those in the first form in this way:

$Kc' = ((Ti + Td)/Ti))Kc$, "effective" gain.
$Ti' = Ti + Td$, "effective" integral or reset time
$Td' = TiTd/(Ti + Td)$, "effective" derivative time

These conversions are made by equating the coefficients of s. Conversions in the reverse direction are:

$Kc = FKc'$
$Ti = FTi'$
$Td = Td'/F$

where

$F = 0.5 + sqrt(0.25 - Td'/Ti')$

Typically Ti is set about 4 to 8 times Td, so the conversion factor is not huge, but it is important to not loose sight of the correction. With this algorithm it is possible to have very troublesome combinations of Ti' and Td'. If $Ti' < 4Td'$ then the reset and derivative *times*, as differentiated from *settings*, become complex numbers, which can confuse tuning. Don't slip into these settings inadvertently! A very knowledgeable tuner may be able to take advantage of that characteristic in very special cases, but it is not for everyone, every day. Some companies advise to use the interacting form if available, simply to avoid that potential pitfall.

This algorithm also has no provision for limiting high frequency gain from derivative action, a virtually essential feature. In the first algorithm Kd is typically fixed at 10, or if adjustable, should typically be set somewhere in the range of 6 to 10. This desirable limiting of the derivative component is sometimes accomplished in this second form by writing it as:

$Kc'(1 + 1/Ti's + Td's)/(1 + Td's/Kd)$

or

$Kc'(1 + 1/Ti's + Td's/(1 + Td's/Kd))$

There are likely many variations on the theme.

The variables Kc', Ti' and Td' have been called "effective". In the Bode plot, **IF** Ti'>4Td', **THEN** Kc' is the minimum frequency-dependent gain (Kc is a frequency-independent gain). This is at a frequency which is midway between the "corners" defined by Ti and Td, which is also midway between the "effective " corners associated with Ti' and Td'. Ti' is always larger than Ti and Td' is always smaller than Td, which recognizes the slight spreading of the "effective" corners of the Bode plot as they approach each other.

This algorithm is also called the "ISA" algorithm. The ISA has no association with this algorithm. Apparently this attribution got started when someone working on the Fieldbus thought it would become "THE" algorithm. It didn't. Or hasn't. ANSI/ISA-S51.1-1979 (Rev. 1993) is a standard on Process Instrumentation Terminology. While this is a standard on terminology, not algorithms, it uses the first form of the algorithm for examples and in its Bode plot for a PID controller. Another term used to identify this algorithm is "ideal". Think of this word as one to *identify* the algorithm, not *describe* it. It is true that it can do everything the first form can do, and more, provided the gain for derivative is handled appropriately. But settings which produce complex roots should be used only by the very knowledgeable.

THIRD FORM OF THE PID ALGORITHM

It is hard to know what to call this third form since it is so close to the second. It has been called "parallel", "ideal parallel", "noninteracting", "independent" and "gain independent". In one sense this third form is the second form rewritten. I understand this is the algorithm taught Electrical Engineers. The second and third forms can be made equal to each other by using the following substitutions:

Kc" = Kc'
Ti" = Ti'/Kc'
Td" = Kc'Td'

They would only differ in what you call the tuning parameters. They are not gain, integral time and derivative time as those words are traditionally used in this field. Also, the option for limiting the gain from derivative action should be handled somehow, perhaps the same way as for form two. One option is as follows:

Kc" + 1/Ti"s + Td"s/(1+Td"s/K"d)

The constraint in the second form that Ti'>4Td' to keep the roots real becomes Kc"Ti">4Td"/Kc", which is a bit more complicated.

PROGRAMMING CONSIDERATIONS

There are many considerations in writing the program for a controller besides the decision on which basic algorithm to use. These include:

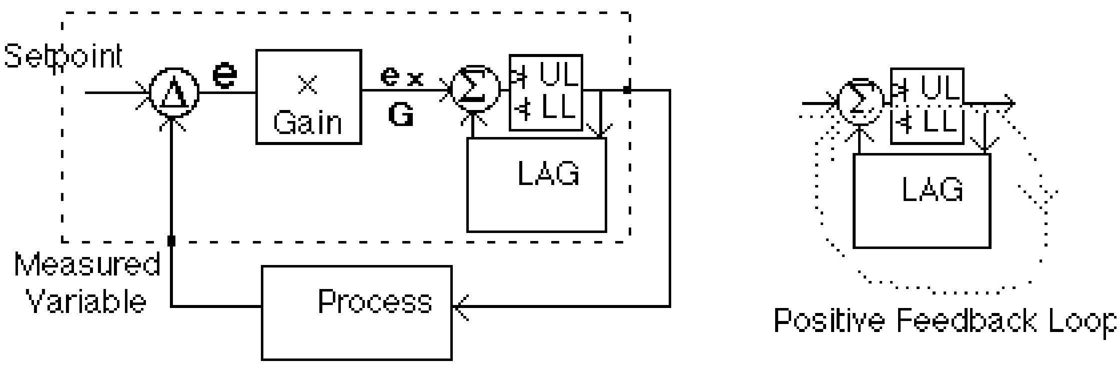
1. The option to have the derivative function act only on the process variable, not on set point changes.
2. The same option with regard to the proportional action. This option may be tied to the first, in that if you choose to have derivative act only on the process variable you get proportional action only on that also.
3. Provision for setpoint and process variable tracking, to permit bumpless automatic/manual transfers. You can have bumpless transfers without setpoint tracking. You can also transfer from manual to automatic without any bump due to proportional action. Aren't all these options wonderful!
4. Provision for reset windup protection.
5. Provision for a filter besides the one used to limit the derivative gain.
6. It is no simple matter to get digital derivative action to approach the quality of analog derivative action. No program can match it. This space is not intended to amplify on that problem, but simply to emphasize that it is a problem. It relates to sampling frequency and noise on the signal. Some algorithms use more than one back value of the controlled variable I believe. Also some manufacturers limit how low a derivative time may be set. It is very difficult for the user to know whether the derivative provided is doing a good job of achieving what could be achieved with derivative action.
7. Integral/reset action with digital controllers is not perfect. There is a phenomenon related to quantizing error, sampling time and long integral/reset times and calculating precision which prevents integrating to zero error. Apparently with more digits in the A/D converter and in the computer's math, this is becoming less and less of a problem.
8. There is the choice of having the algorithm be "velocity", sometimes called "incremental" (each calculation period a *change* in the output is calculated), or "position" (each calculation period the actual desired output is calculated). Apparently at one time there was a perception that the velocity algorithm did not have a reset windup problem, but this is not the case. The choice between the incremental and position algorithms seems to be a choice based on many considerations which are beyond the scope of this write-up.
9. There are options on filtering noise, such as providing a dead zone or a zone of low gain around the setpoint.
10. There are options to be considered in special cases, such as preventing reset windup in override and cascade situations.
11. Provision needs to be made for manual bias.
12. There must be other points to make to caution the novice. Does anyone want to suggest some?

Equivalent code

The PID algorithm, as implemented in a typical digital control system, can be understood by reference to a small basic program that is the equivalent to the PID algorithm in its most common application.

[Further explaination of the PID algorithm]

1. A positive feedback integral algorithm is used.



2. Derivative is on process.

3. Relative moderate tuning coefficients (<10) are used.

4. Output limits are 0 and 100%

5. The loop is scanned every second

Variables:

`Input`	*The process input, in percent*

`InputD` *Process input after derivative calculation*
`InputLast` *Process input on the previous pass*
`InputDF` *Input after derivative calculation and filter*
`Feedback` *internal feedback for reset after filter*
`Derivative` *Derivative time in minutes*
`Gain` *Gain, negative if controller is reverse acting*
`ResetRate` *Reset Rate in repeats per minute*
`DFilter` *Derivative filter time constants, in minutes*
`OutputTemp` *Result of the PID calculation*
`Output` *The final output*

The PID emulation code:

```
InputD=Input+(Input-InputLast)*Derivative *60        Derivative calculation
InputLast=Input
InputDF=InputDF+(InputD-InputDF)*DFilter/60          Derivative filter
OutputTemp=(InputDF-SetPoint)*Gain+Feedback          Basic gain calculation
IF OutputTemp >100 THEN OutputTemp= 100              Output Limits
IF OutputTemp <0 THEN OutputTemp= 0                  Values other than 0 and
100 may be used
Output=OutputTemp                                    The final output
Feedback=Feedback+(Feedback-Output)*ResetRate/60  Filter for reset feedback
```

Comparison of PID Control Algorithms
(All Controllers Are Not Created Equal)

Modified from an article published in Control Engineering March, 1987. This article updated and re-written for the Web.

One fine day, a plant engineer, replaced his controllers. Even though he used the same settings on the new controllers, the retrofitted loops went out of control in automatic. He tried to tune these controllers the same way he had tuned the old ones. The loops seemed to get more unstable.

This mysterious and very real situation is the result of two manufacturer's using different PID algorithms. Read on to solve this and other common mysteries about PID controllers.

In practice, manufacturers of controllers don't adhere to any industry wide standards for PID algorithms. Different manufacturers and vendors use different PID algorithms and sometimes have several algorithms available within their own product lines.

The figures and graphs used in this article were produced using the ExperTune Loop Simulator. For PID loop tuning, analysis and simulation contact ExperTune.

The Name Game

Just as there are no adhered to industry standards for PID controllers, nomenclature and *action* for similar modes varies.

```
P   Proportional Band = 100/gain
I            Integral = 1/reset
D          Derivative = rate = pre-act
```

Some manufacturers call Proportional Band the Proportional Gain. Manufacturers interchange names and units for integral or reset action. In this article, integral action is defined in time/repeat and reset in repeat/time. One is the reciprocal of the other. The action of either reset or integral can be reversed depending on the manufacturers units.

The Algorithms

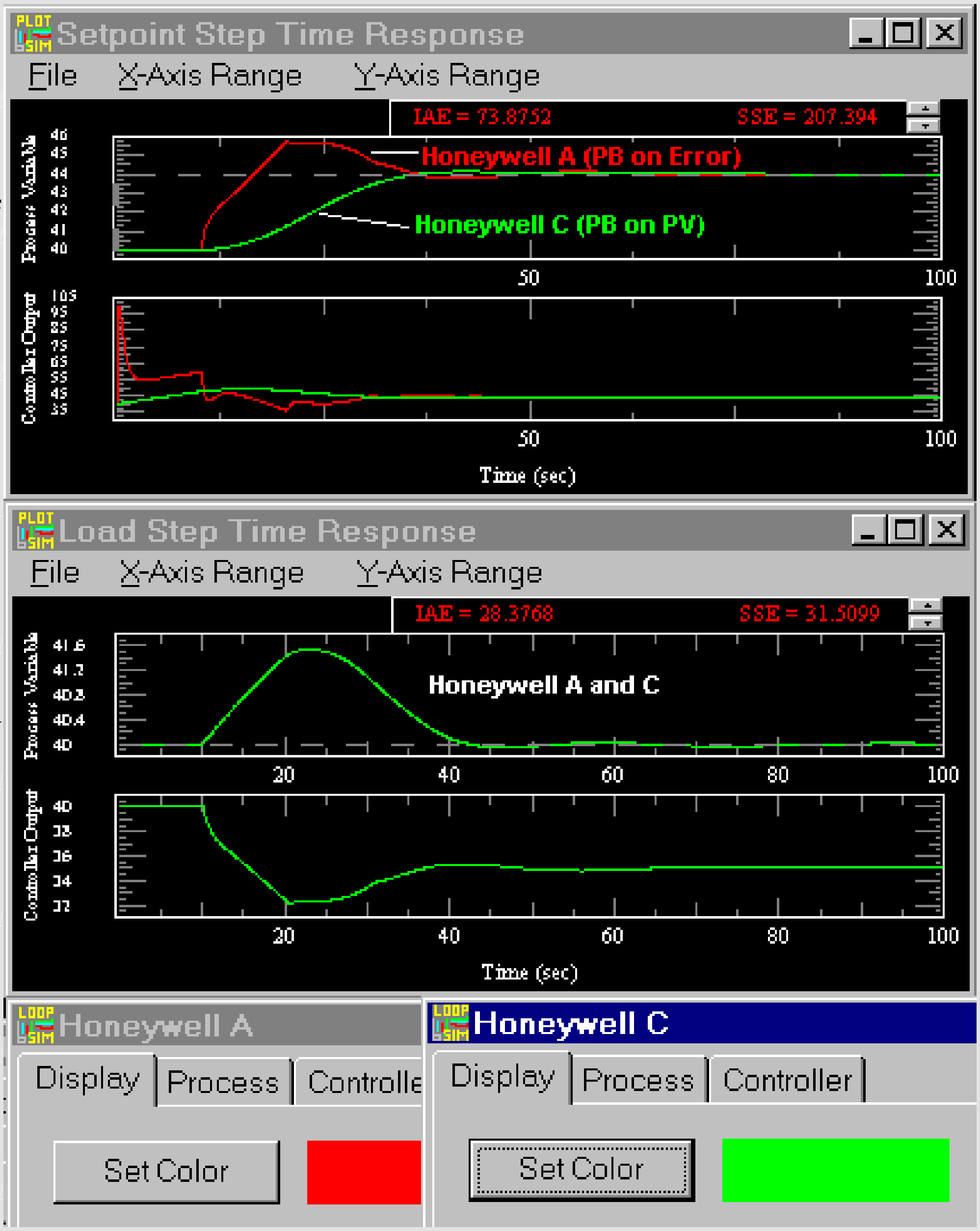
There are three major classifications of PID algorithms that most manufacturer's algorithms fit under. These three are: series, ideal, and parallel. Again, manufacturers vary on the their names for these categories. The only way to really tell which one you have is to look at the equation for the controller. In simple form these are:

Ideal algorithm:	$\text{OUTPUT} = Kc \left[e(t) + \frac{1}{I} \int e(t)d(t) + D \frac{d\, e(t)}{dt} \right]$
Parallel:	$\text{OUTPUT} = Kp\, [e(t)] + \frac{1}{I} \int e(t)d(t) + D \frac{d\, e(t)}{dt}$
Series (Interacting):	$\text{OUTPUT} = Kc \left[e(t) + \frac{1}{I} \int e(t)d(t) \right] \left[1 + D \frac{d}{dt} \right]$

Kc, Kp are gain; I, Ip are integral and D, Dp are derivative settings. The series controller's strange looking form makes it act like an electronic controller. A three term controller can be made with only one pneumatic (or electronic) amplifier using the series form. Thus, pneumatic controllers and early electronic controllers often used the series form to save on amplifiers which were expensive at the time. Some manufacturers use the series form in their digital algorithms to keep tuning similar to electronic and pneumatic controllers.

Differences in Proportional Band Or Gain

If you use only proportional action, the main difference between series and ideal algorithms is that some manufacturers use proportional band while others use gain. On a controller using the gain setting, *increasing* this setting makes the loop more sensitive and less stable. While *decreasing* proportional band on controllers using it will have the same effect.



Some manufacturers allow more flexibility with P action by letting you choose whether gain (proportional) action works on set point changes. For example, Honeywell TDC has two types of algorithms that work differently on set point changes. With their type A algorithm, gain action acts on set point changes and with their type C it does not. For load upsets, type A and C act the same, but for set point changes, the difference is dramatic.

ExperTune Loop Simulator Windows compare responses of Honeywell TDC type A (proportional action on error) and C (proportional action on PV only) controllers on a simulated temperature loop. Bottom graph shows either type controller response to a step load change. Top plot red is response of Type A, top plot green of Type C. With the type C algorithm, damping and overshoot to step set point changes are similar to damping and overshoot on step load disturbances. Type C may be desirable over type A, since tuning for load or set point changes is similar with type C.

Because of the sensitive set point response, you may want to use type A for the inner loop or slave in a cascade. Type C with smooth set point response may be better for the outer or master loop.

Bailey's "error input" and "PV and SP" algorithms are analogous to Honeywell's type A and C. Bailey's "error input" has sensitive set point response while Bailey's "PV and SP" has smooth set point responses.

Like Honeywell's, the two Bailey algorithms give identical load responses. Because the load responses are the same for the different Bailey and Honeywell algorithms, they have the same stability. For a fast analysis of the stabilities, ExperTune allows a comparison of the robustness plots of the algorithms.

Differences In Integral Action

Once you convert integral and reset values to the same units, PI controllers respond mostly the same for load disturbances. The proportional action may be different as described above. Anti-reset windup is usually done differently, but the effect of these differences is usually minor compared to other differences between algorithms.

Differences In Derivative Action

The largest variation among controllers from different manufacturers is the way they handle derivative action. Virtually no two are the same. This is part of the reason why many people don't use derivative action. The differences are caused from different methods of filtering or not filtering at all, whether the derivative works on set point changes or not, and how derivative interacts or does not interact with the integral action.

On controllers, when you set derivative to something besides zero, you get derivative action. In a series controller, when you use both integral and derivative actions, the integral and derivative modes interact. Interaction causes the effective controller action to be different from what it would be in a ideal controller.

```
The effective proportional band is:

  PB (effective)  = PB/(1 + D/I)

The effective integral time is:

  I (effective) =  I + D

The effective derivative time is:

  D (effective) = 1 / (1/I + 1/D)
```

Where PB, I and D are the proportional band, integral and derivative values you set or entered into the series controller. The effective values are equivalent ideal controller settings.

These equations show that for the series controller you cannot make the effective derivative time greater than 1/4 the effective integral time. The largest effective derivative occurs when D=I. When D is set larger than I, the effective integral time is adjusted more with D and the effective derivative is adjusted more with I! Therefore it is usually good control practice to keep the values of D less than I for a series controller.

For example: Foxboro and Fisher use a series algorithm. AEG Modicon, Texas Instruments controllers use the ideal type. Honeywell has both series and ideal algorithms. Bailey, Allen Bradley, and GE have both ideal and parallel algorithms.

Other Differences in Derivative

Besides the interaction differences described above, derivative action among the series and ideal groups varies.

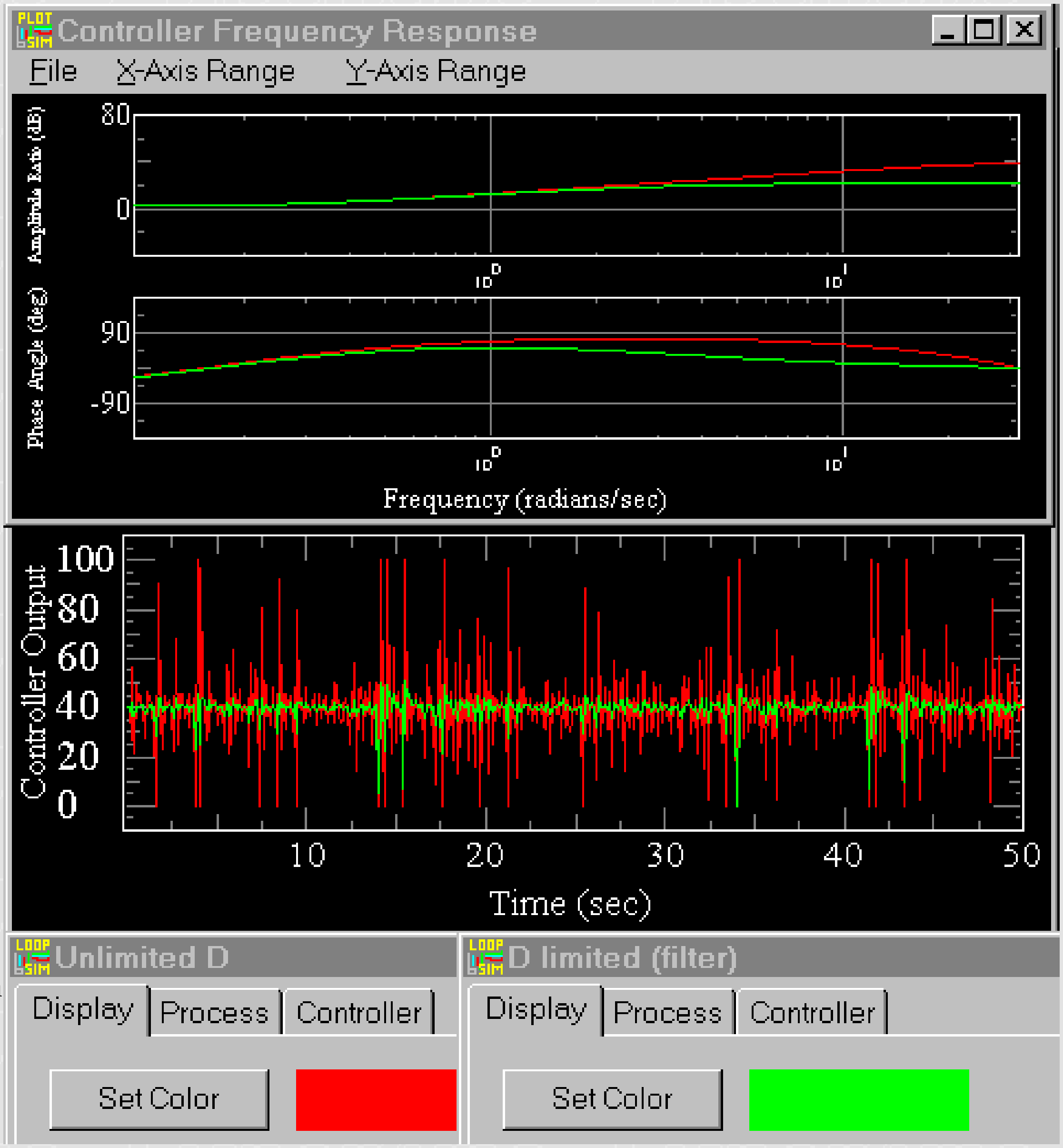
With most controllers, derivative works only on measurement. On some controllers however, derivative action works on set point changes. Although response to a load disturbance will be the same, set point response on these controllers can get out of hand.

Since most controllers are used for regulating disturbances, derivative action working on set point changes is usually not a problem except in cascade loops or ones where the set point is being manipulated.

Of more significance is whether and how filtering is done when you dial in derivative.

The Unlimited Derivative Problem

Some manufacturers do not filter or limit derivative action. Thus, at high frequencies, the amplitude ratio gets large. In the Figure the red line shows the amplitude ratio of unlimited derivative action.



The Figure shows the same PV noise added to a PID controller with (green) and without (red) PV filtering. PV filtering limits the derivative gain.

Unlimited derivative action does not help good loop control but does amplify measurement noise in the controller output. The result of unlimited derivative is a "jumpy" or nervous and noisy controller output. The lower graph in the Figure - red line is the time response of a controller to measurement noise. This can wear out valves, or drive a slave loops set point crazy. Worse yet, the noise can drive the controller into saturation which causes the anti-reset code to take over. No wonder derivative is seldom used!

Filtering Limits Derivative Noise

On the controllers that use filtering with derivative, usually the measurement signal gets the filtering. The time constant of filtering is usually calculated by these algorithms based on the derivative value dialed in. The amount of filtering changes with the amount of derivative. This has the effect of limiting derivative action at high frequencies. In Figure the green line shows the amplitude ratio and controller output using limited derivative action. Control loop performance is the same on both since unlimited derivative does not improve control loop performance.

Parallel Controllers

With parallel controllers, controller gain is not multiplied by the error signal only. Integral and derivative actions are "independent" of the controller gain.

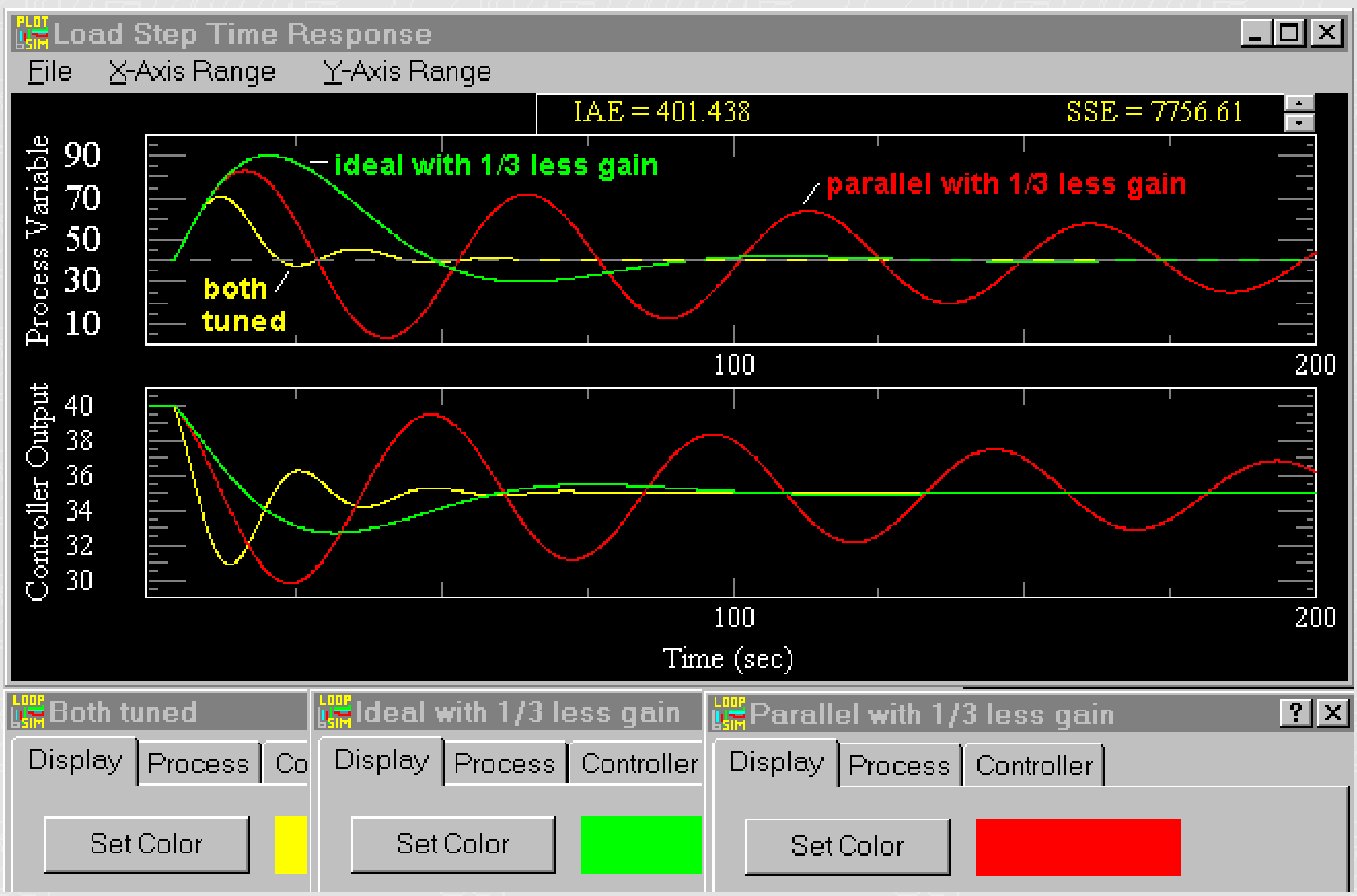
At first it might seem that the parallel controller is easier to work with because of this "independence". But, parallel algorithms require very different integral and derivative tuning parameters than other controllers. These equations show how to convert from parallel to ideal settings:

```
Kc = Kp

I = Ip Kp (units of time/repeat)

D = Dp/Kp (units of time)
```

There is more of a difference between parallel versus ideal controller tuning than series versus ideal tuning. The intuitive feel for tuning a parallel controller is very different from the others. The Figure below shows load response in a level loop. The yellow curve is for a tuned (ideal or parallel) controller. Normally it would seem that lowering the controller gain will make the loop more stable as in fact it does with the ideal controller in the Figure. However, the parallel controller gets *less* stable with lower gain!. Like all controllers, it also gets less stable with more gain. So *either increasing or decreasing* the gain on a parallel controller can drive the loop unstable! The controllers in the Figure are PI controllers. The situation is more pronounced when you use derivative.



With the parallel controller, the effective integral and derivative values change with the gain setting. So, lowering the controller gain also lowers the effective I; increasing controller phase. Lowering gain also increases the effective D, moving the derivative phase to higher frequencies; eliminating its stabilizing effect on controller integral action. The overall effect is unstabilizing as the Figure shows. For example: Bailey and Allen Bradley both have a parallel algorithm available that they describe as a "non-interacting" algorithm. They call the ideal algorithm an "interacting" one.

Conclusions

Choosing the best algorithm for your process is dependent on your process control needs and objectives. Different algorithms perform better in different situations. By using ExperTune simulation software, these differences are easier to understand.

PID Control Technical Notes

General

PID (Proportional-Integral-Derivative) control action allow the process control to accurately maintain setpoint by adjusting the control outputs. In this technical note we have attempted to explain what PID is in practical terms. We have available further technical references for our customers.

ISE has a complete line of PID controls suitable for virtually an application. We also have numerous tools (such as software, data loggers and recorders) to help to optimize any control application. Our application engineers have extensive practical knowledge in the tuning of PID controls to all types of applications.

While controls can be used for many different process variables for clarity we have chosen to use temperature as the process variable throughout these notes. Other processes can utilize these control concepts and the effects will be the same.

PROPORTIONAL & PID CONTROL ACTION

Proportioning control continuously adjusts the output dependent on the relative positions of the process temperature and the setpoint. PID (Proportioning/Integral/Derivative) are control functions commonly used together in today's controls. These functions when used properly allow for the precise control of difficult processes.

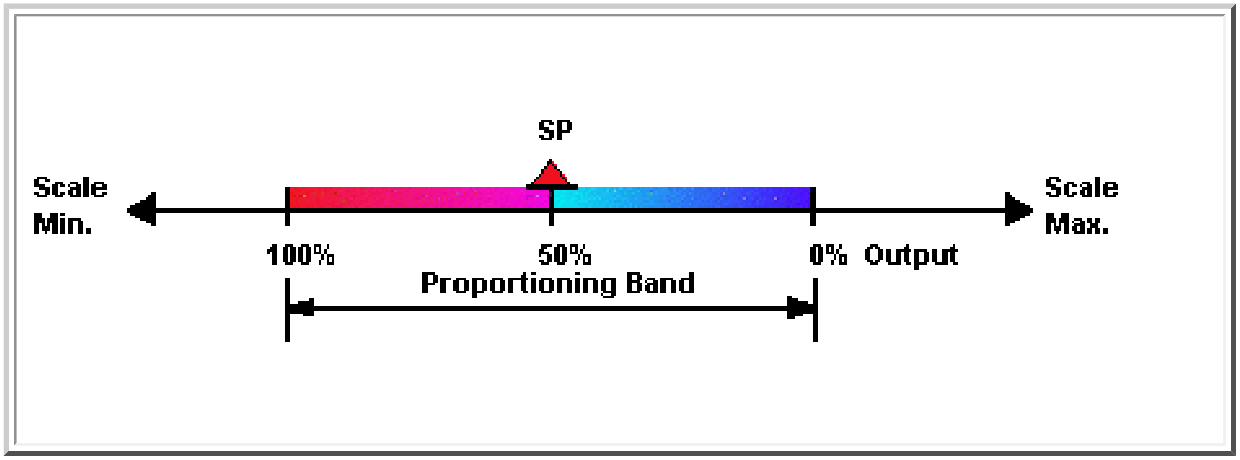
General:

1) Allows for the output to be a value other than 100% or 0%.

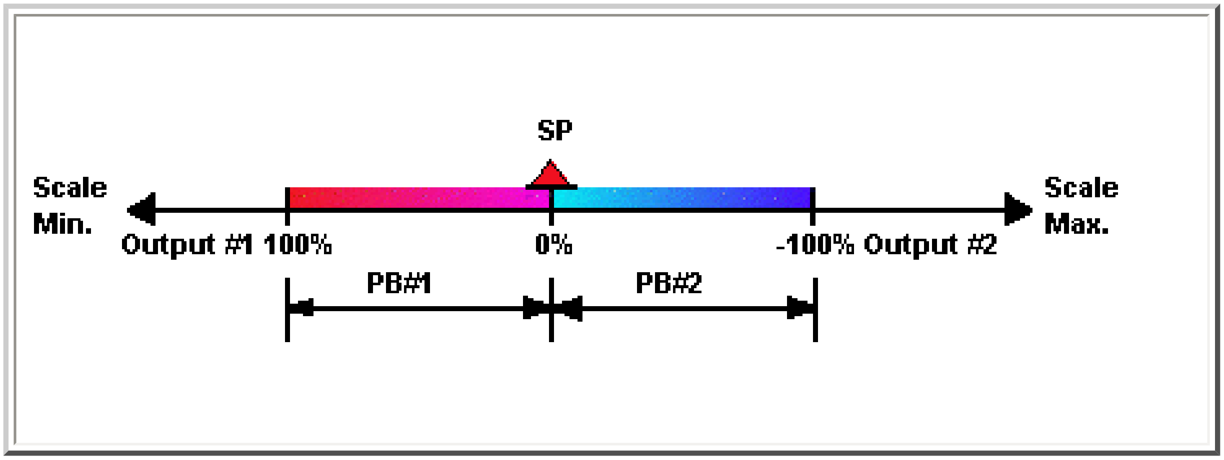
2) Temperature can be controlled without oscillations around the setpoint.

Definitions:

1) **Proportioning Band:** is the area around the setpoint where the controller is actually controlling the process; The output is at some level other than 100% or 0%. The band is generally centered around the setpoint (on single output controls) causing the output to be at 50% when the setpoint and the temperature are equal.



On (2) two output controls (i.e.: heat/cool) there are two proportioning bands. One is for heating and one is for cooling. In this case the bands generally end at the setpoint as shown below.



Proportioning bands are normally expressed in one of three ways:

- As a percentage of full scale
- As a number of degrees (or other process variable units)

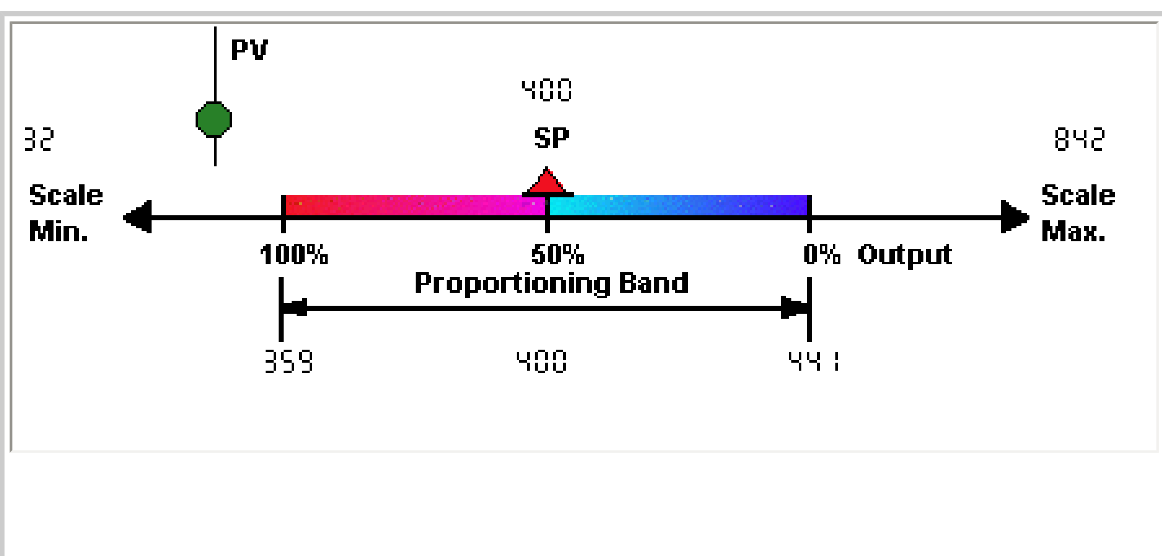
- Gain which equals 100%/proportioning band% (example PB% = 5; Gain = 20)

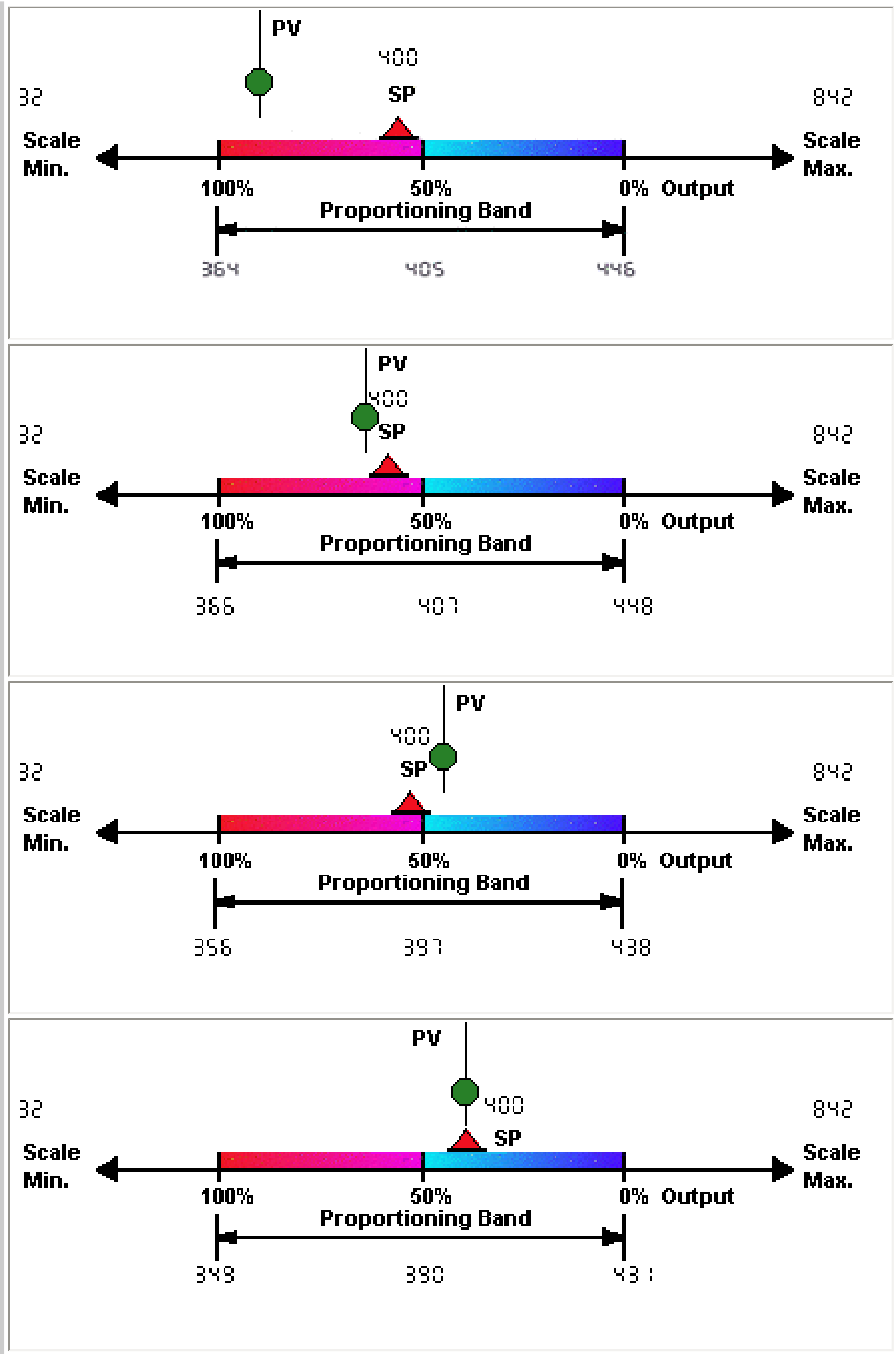
If the proportioning band is to narrow an oscillation around the setpoint will result. If the proportioning band is to wide the control will respond in a sluggish manner, could take a long time to settle out at set point and may not respond adequately to upsets.

Manual Reset: Virtually no process requires precisely 50% output on single output controls or 0% output on two output controls. Because of this many older control designs incorporated an adjustment called manual reset (also called offset on some controls). This adjustment allows the user to redefine the output requirement at the setpoint. A proportioning control without manual or automatic reset (defined below) will settle out somewhere within the proportioning band but likely not on the setpoint. Some newer controls are using manual reset (as a digital user programmable value) in conjunction with automatic reset. This allows the user to preprogram the approximate output requirement at the setpoint to allow for quicker settling at setpoint.

Automatic Reset (Integral): Corrects for any offset (between setpoint and process variable) automatically over time by shifting the proportioning band. Reset redefines the output requirements at the setpoint until the process variable (temperature) and the setpoint are equal. Most current controls allow the user to adjust how fast reset attempts to correct for the temperature offset. Control manufacturers display the reset value as minutes, minutes/repeat (m/r) or repeats per minute (r/m). This difference is extremely important to note for repeats/ minute is the inverse of minutes or minutes/ repeat). The reset time constant must be greater (slower larger number m/r smaller number r/m) than the process responds. If the reset value (in minutes/repeat) is to small a continuous oscillation will result (reset will over respond to any offset causing this oscillation). If the reset value is too long (in minutes/ repeat) the process will take to long to settle out at setpoint. Automatic reset is disabled any time the temperature is outside the proportioning band to prevent problems during startup.

Below is an example of a single output (heat only temperature control) with a 10% proportioning band and a setpoint of 400. Note how reset shifts the proportioning band when the temperature (PV) enters the proportioning band.





Reset stops moving the proportioning band as soon as the setpoint and PV are equal. In the above example reset determined approximately 38% output is required to maintain setpoint. Stable control is achieved and the temperature matches the setpoint of 400.

Rate (Derivative): Shifts the proportioning band on a slope change of the process variable. Rate in effect applies the "brakes" in an attempt to prevent overshoot (or undershoot) on process upsets or startup. Unlike reset rate operates anywhere within the range of the instrument. Rate usually has an adjustable time constant and should be set much shorter than reset. The larger the time constant the more effect rate will have. Too large of a rate time constant will cause the process to heat too slowly. Too short and the control will be slow to respond to upsets. The time constant is the amount of time any effects caused by rate will be in effect when rate is activated due to a slope change.

Self Tuning /Adaptive Tuning / Pre-Tuning

Many control manufactures provide various facilities in their controls that allow the user to more easily tune (adjust) the PID parameters to their process. Below is a description of same.

Tuning On Demand with Upset: This facility typically determines the PID parameters by inducing an upset in the process. The controls proportioning is shut off (on-off mode) and the control is allowed to oscillate around a setpoint. This allows the control to measure the response of the process when heat is applied and removed (or cooling is applied). From this data the control can calculate and load appropriate PID parameters. Some manufactures perform this procedure at setpoint while others perform it at other values. Caution must be excersized for substantial swings in the process variable values will likely occur while the control is in this mode.

Adaptive Tuning: This mode tunes the PID parameters without introducing any upsets. When a control is utilizing this function it is constantly monitoring the process variable for any oscillation around the setpoint. If there is an oscillation the control adjusts the PID parameters in an attempt to eliminate them. This type of tuning is ideal for processes where load characteristics change drastically while the process is running. It cannot be used effectively if the process has externally induced upsets for which the control could not possibly tune out. For example: A press where a cold mold is inserted at some cyclic rate could cause the PID parameters to be adjusted to the point where control would be totally unacceptable.

Some manufactures call Tuning on demand Self Tune, Auto Tune or Pre-Tune. Adaptive tuning is sometimes called Self Tune, Auto Tune or Adaptive Tune. Since there is no standardization in the naming of these features questions must be asked to determine how they operate.

General Control Types

ON-OFF CONTROL ACTION

On-Off control is the most basic form of temperature control.

1) Changes output only after temperature crosses the setpoint.

2.) Should only used on non-critical applications, The process temperature never stabilizes at the setpoint due to process inertia.

3.) Also used in alarms and safety circuits.

4.) Most PID controls operate in this mode if the proportioning band is set to "0".

Time Proportioning Controls

1.) Vary the output by cycling a relay, SSR or logic voltage on and off.

2.) Proportions by varying the On Time versus Off Time.

3.) Usually include a parameter such as Cycle Time which is the total of the On Time and the Off Time. Example of its operation is as follows: With a Cycle Time of 10 seconds if the control decides it wants 42% heat applied to the process the output would be On for 4.2 seconds and Off for 5.8 seconds giving an effective output of 42%.

Linear Output Controls

1.) Provides a DC voltage or current output related to the required output demand. For example: If the control has an output range of 4-20 mA and decides it wants 50% power to the process the control would output 12.0 mA

2.) Normally connected to an SCR Power control or other solid state device. The power handling device then converts this signal to a relative power output.

Closed Loop Valve Motor Controls

1.) These controls are used in conjunction with motor actuators in gas heating applications. The control has (2) outputs (typically relays) one for clockwise rotation and one for counter clockwise rotation.

2.) Feedback as to motor position is provided by a potentiometer attached to the motor.

Open Loop Valve Motor Controls

1.) These controls are used in conjunction with motor actuators in gas heating applications. The control has (2) outputs (typically relays) one for clockwise rotation and one for counter clockwise rotation.

2.) No feedback as to motor position is provided. The user enters a value for motor travel time in the control. This allows the control to determine how long to operate the motor in either direction.

High & Low Limit Controls

1.) Usually used as safety devices upon the failure of the primary control device or some other failure in the system.

2.) Once the process variable goes through the limit setpoint the controllers output switches. The output will not revert back to normal until the process variable returns to a safe value and a reset button is pressed.

3.) Most insurance companies require FM approved limit devices on certain applications particularly on gas fired and on applications that are left unattended.

4.) For complete safety a separate sensor and contactor is required. On electric applications utilizing SCR power controls a contactor connected to the incoming power of the SCR should be used to protect against SCR failure.

Ziegler-Nichols Closed Loop Tuning

The Ziegler-Nichols Closed Loop method is one of the more common methods used to tune control loops. It was first introduced in a paper published in 1942 by J.G. Ziegler and N.B. Nichols, both of whom at the time worked for the Taylor Instrument Companies of Rochester, NY.

The open loop method is useful for most process control loops. To use the method the loop is tested with the controller in automatic. The Closed Loop method determines the gain at which a loop with proportional only control will oscillate, and then derives the controller gain, reset, and derivative values from the gain at which the oscillations are sustained and the period of oscillation at that gain.

The ZN Closed Loop method should produce tuning parameters with will obtain quarter wave decay. This is considered good tuning but is not necessarily optimum tuning.

Steps

- Ensure that the process is "lined out" with the loop to be tuned in automatic with a gain low enough to prevent oscillation.
- Increase the gain in steps of one-half the previous gain. After each increase, if there is no oscillation change the setpoint slightly in order to trigger any oscillation.
- Adjust the gain so that the oscillation is sustained, that is, continues at the same amplitude. If the oscillation is increasing, decrease the gain slightly. If it is decreasing, increase the gain slightly.
- Make note of the gain which causes sustained oscillations and the period of oscillation. These are the "Ultimate Gain" (GU) and the "Ultimate Period" (PU) respectively.
- Calculate the tuning for the following set of equations. Use the set which corresponds with the desired configuration: P only, PI, or PID.

Tuning Equations

- P Only: Gain = 0.5 GU
- PI: Gain=0.45 GU, Reset=1.2/PU
- PID:Gain=0.6GU, Reset=2/PU, Derivative=PU/8

A practical approach for large-scale controller performance assessment, diagnosis, and improvement

Abstract

Eastman Chemical Company has developed a large-scale controller performance assessment system spanning over 14,000 PID controllers in 40 plants at 9 sites worldwide. Controllers can be sorted in order of performance to quickly identify which need attention. Performance history is available to track improvement or degradation in performance for a single controller or an entire plant. Diagnostic aids are available for both novices and experts to substantially reduce troubleshooting time. E-mail reports are automatically generated and sent to subscribers to keep them informed of relevant changes with minimal investment of their time. The user interface is web-based to allow universal access to any employee. Use of the system has dramatically increased controller optimization productivity.

Keywords: PID controller; Performance assessment; Troubleshooting; Diagnosis

1. Introduction

The late 1990s were a time of change at Eastman Chemical Company. Product prices were dropping due to ever-present competition as well as world over-capacity built with strong profits from the last industry upcycle. Raw material prices were increasing due to increasing cost of petroleum feedstocks. Investors were much more interested in chasing the rising spiral of technology stocks rather than sectors like basic materials. The joint arrival of these market forces triggered development of a "perfect storm" for chemicals manufacturing. The business strategies put in place to deal with the situation had a very common theme, "do more with less". Higher quality and production rate was demanded with fewer people, less energy, less raw materials, and importantly, lower capital investment.

For the Advanced Controls Technology group at Eastman, these changes meant a dramatic reduction in control strategy development for new plants. Fortunately, the demand for process control work in existing plants was higher than ever, as production managers were eager to gain the benefits of control improvements that could be delivered with little or no cash out the door. The renewed focus on existing operations revealed numerous opportunities for process control improvement. During these improvement projects, it was very typical to find poorly performing control loops. The most common problem was oscillation as a result of valve hysteresis or stiction. Indeed, the problem was alarmingly common. It quickly became clear that looking for oscillating or poorly performing control loops one-by-one could consume the full resources of the control group and limit investigation to a fraction of all processes within the company. A vision began to form of a controller performance assessment tool that would enable efficient detection and diagnosis of problems in the many thousands of control loops in service worldwide at Eastman.

Studies of controller performance assessment algorithms began to appear in the early 1990s after the work of Harris [1], in which closed-loop time-series controller data were analyzed to benchmark controller performance against minimum variance control. Desborough and Harris [2], Kozub and Garcia [3], and Stanfelj et al. [4] extended this concept with a performance index. Comprehensive reviews of performance index algorithms are provided by Qin [5] as well as Huang and Shah [6].

* Corresponding author. Tel.: +1-423-224-0409; fax: +1-423-224-0453.

E-mail address: jwcox@eastman.com (J.W. Cox).

0959-1524/02/$ - see front matter
PII: S0959-1524(02)00018-5

A performance assessment system making use of a performance index and other analyses of closed-loop process data was reported by Jofriet et al. [7]. However, this system required the deadtime of each loop to be specified by the user, creating a significant burden. An open loop bump test and analysis for each controller was suggested to determine this parameter. Kozub [8] describes an alternative industrial application where a deadtime specification is not necessarily required, but a specification of the desired autocorrelation function of the controller error is required. In some sense, this variation requires even more effort per loop to configure, but it also makes the benchmark performance more relevant to industrial operation. Thornhill et al. [9,10] introduced a significant advance in which default parameters for the performance index algorithm were shown to be useful for various generic categories of control loops. This work lowered the barrier to large-scale implementation of a performance index.

Thornhill and Hagglund [11] extended performance assessment in a different direction with a method to detect and diagnose oscillations in control loops. Together with a performance index, this paper presented an outline of a complete approach to controller assessment.

Miller et al. [12] described a comprehensive system for controller performance assessment developed by Honeywell Hi-Spec Solutions (Thousand Oaks, CA). This system is now offered to the process industries as a service called Loop Scout™. Other industrial applications of performance assessment as well as the challenge of developing a large-scale, automated system are discussed by Harris et al. [13].

Eastman considered using the Loop Scout™ service in 1999, but chose not to for three primary reasons:

- Automated data collection was limited to Honeywell control systems.
- Substantial amounts of process data would have to be sent to Honeywell, requiring complicated approvals.
- The cost to assess loops worldwide was prohibitively high given the emphasis on reducing business expenses.

At that time, we refined our vision for large-scale controller performance assessment at Eastman. Features we felt were necessary included:

- Interfaces to all Eastman's DCS systems, PI systems, and our in-house advanced control platform.
- A friendly user interface, providing accessibility to all company personnel.
- Minimal client and server configuration.
- Periodic assessment of loops and retention of performance history.
- Ranking loops by performance.
- Preliminary problem diagnosis for poorly performing loops.
- Reports which could be generated interactively or predefined reports which could be e-mailed to users on a schedule.
- Ability for users to add comments and documents to the system and link them to loops or loop tuning changes.

A search of the marketplace in 1999 did not reveal any system that we could purchase that included most or all of these features. As a result, we chose to develop one of our own.

2. System architecture

A schematic diagram of our system is shown in Fig. 1. Selected portions of the system are numbered in the figure and are discussed in more detail below.

2.1. Block 1—data interface

Closed-loop controller data is automatically captured directly from Eastman control systems at regular intervals. We have developed data interfaces for most of the control systems in use at Eastman. We considered keeping the data interface simpler by using data from Eastman's company-wide system of PI historians, but the typical data compression was far too great for the assessment analyses. As OPC becomes more prevalent, a universal data interface may become practical. Data capture is handled in a distributed manner, with processing power at each control system being used to collect the data and push it to the assessment system. This approach helps the system scale nicely to many thousands of loops. An interface to PI and to Eastman's advanced control platform is also available for assessments on demand. Controller tuning is captured on a daily basis and pushed to the assessment system.

2.2. Block 2—computation engine

Computational software of our own design does performance assessment on controller data shortly after capture. The full assessment includes:

- Time-series trends (full and zoomed).
- Descriptive statistics for controller setpoint, measurement, output, and error.
- Setpoint crossings (number of times measurement crosses setpoint).
- Univariate and bivariate density of error and output.
- Closed-loop impulse response.

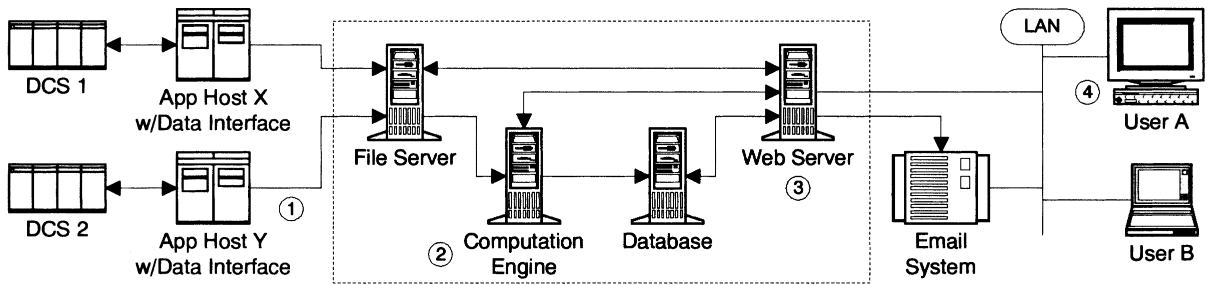


Fig. 1. Schematic diagram of the controller performance assessment system.

- Harris-style extended horizon performance index.
- Power spectrum.
- Oscillation detection and characterization.
- Controller idle index [14].
- Cross-correlation between error and output.
- Diagnosis information in text form.
- A single composite score ranking the loop performance.

Numeric and text information from the assessments is written to a database. The computation engine is also available for assessments on demand. We can quickly and easily modify assessment analyses or add new ones since we are in control of the source code.

2.3. Block 3—web server

A web server is at the heart of the assessment system. A web-based user interface was chosen to eliminate the need for distribution and installation of client software and also because users are very familiar with web pages, reducing the learning curve for use of the system. Very capable software is available for web-database integration and web server-side programming, making a web server a strong application platform.

2.4. Block 4—user interface

The user interacts with the system via a web browser. Hyperlinked web pages allow the user to navigate the system. Web forms are used for user input. Web pages with text, tables, images, and applets are used to present the system output. In addition, users can subscribe to a variety of daily or monthly reports for process areas in which they have interest. These reports are sent to the user via HTML e-mail.

2.5. Scaling the system

The dotted box in Fig. 1 shows separate computers performing individual functions within the assessment system. This configuration is shown for illustration only and is flexible. For a small system, all functions could be performed in a single computer or some number less than four as shown. For a very large system, individual functions such as web serving could take place in multiple computers with load-sharing capability.

3. Features and capabilities

The assessment system has many features designed to help users identify controllers needing attention, diagnose controller problems, and track controller performance over time. These features include e-mail reports, detailed performance assessments, and detailed controller information (includes tuning history and analysis, performance history and problem diagnosis, loop configuration information, etc.). E-mail reports for individual process areas are sent daily or monthly, based on the report type, to interested users who subscribe. To illustrate the capabilities of the assessment system, consider the following descriptions of how these features can be used.

3.1. Daily tuning change report

A daily tuning change report is sent to a subscriber's e-mail inbox once per day and lists controllers having tuning changes during the past day. With this report, users are quickly alerted to controller tuning changes in their area. This is important because tuning changes are often indications of significant problems, and communication of tuning changes to all appropriate personnel by other means has typically been poor, at least at Eastman. Tuning changes are often the first attempt to remedy controller problems because they are easy and cheap. However, in most cases, tuning changes are not the best response to the event that made good controller performance become bad. The daily tuning change report is a reliable and timely way to alert users about potential problems in their process area.

The example in the next section illustrates the use of an actual daily tuning change report. A report excerpt is shown in Fig. 2. For the process area in question, loop LCDC801 had a tuning change in the last day.

3.2. Tuning and performance history

For individual loops, users can pull up a web page that contains tuning history, tuning analysis, performance history, performance analysis, and loop configuration information. The web page can be requested interactively through the performance assessment web site or it can be requested from a hyperlink that appears in HTML e-mail or other documents capable of including links. In the case of the tuning change report in Fig. 2, the link is the tag name. To further investigate the LCDC801 tuning change, the user simply clicks on the tag name while viewing the e-mail report. A web browser pops up and loads a web page full of tuning, performance, and configuration information. An excerpt from this web page is shown in Fig. 3. The following abbreviations are used in the performance history table in Fig. 3: PV—process variable (measurement), OP—controller output, Err—controller error, Std Dev—standard deviation, CLPA—closed-loop performance assessment (a Harris-style performance index), Osc Diag—diagnosis from oscillation analysis, SP Cross—count of times the measurement crosses the setpoint.

The tuning history table shows that the previous tuning had been in place for over 2 years. It seems unlikely that the tuning would have persisted that long if it were bad. The tuning analysis does not indicate that the tuning was changed to unreasonable values. It was simply a small change, probably an experiment.

A glance further down the page at the performance history table reveals that perhaps the tuning is not the problem after all. Performance measures are available weekly, except for the time between 25 May and 19 August, when the process was shut down. There was a dramatic change in the controller performance when the process came back up. The CLPA statistic jumped towards 1, indicating a substantial departure from efficient control. The oscillation detection analysis also began to signal a hardware problem (“hdw” in the “Osc Diag” column), where before it had indicated no problem or some oscillating disturbances.

However, the table also shows that the mean error for the loop has not increased substantially. It is a little higher than previous, but still averaging about 1% of the level span. This level performance is adequate for the reactor. Thus, for this case, the investigation reveals that the loop might have some problems, but they do not appear to be serious at this point. It would be appropriate to follow up on the LCDC801 loop at a convenient time in the future to find out if the valve may need some maintenance.

The following loops had tuning changes in the past day or are new to the database in the past day. You can click on the tagname to get more information. If you made a tuning change, please click the tagname, claim the tuning in the tuning history table, and add a few notes about why you changed it.

Tagname	Description	Gain %/%	Integral rpt/sec	Derivative sec	Filter smooth
LCDC801	DR1 LEVEL CONTROL	0.50	0.00100	0.00	0.00

Fig. 2. Excerpt from a daily tuning change report delivered by HTML e-mail.

3.3. Performance detail report

The performance detail report is another available e-mail report that is extremely useful to plant personnel interested in minimizing process variability and optimizing performance of critical loops. This monthly report contains summary statistics, loop performance rankings, and a section highlighting changes in loop performance for an entire plant or process area. While this report can be used in many different ways, its strength is in identifying the worst performing loops out of the many loops (hundreds to thousands) in the process area. Technical personnel familiar with the process area can pick out loops that need attention (those rated as poor or fair performers which are critical to the process operation) in a matter of minutes. Once a problem loop has been identified, historical information useful for problem diagnosis is easily accessible, again with a hyperlink to the web-based loop information page.

Consider the following example to illustrate the use of an actual performance detail report. A very small excerpt from the main section of the report (which lists loops in ascending order of performance) is shown in Fig. 4. The following abbreviations are used in Fig. 4 in addition to those already listed for Fig. 3: Perf. Class—performance classification, Crit—criticality, Pct Osc Hdw Diag—percentage of assessments where the oscillation analysis results in a diagnosis of hardware problems, Num Assess—number of assessments.

As a user scans the report, attention is drawn to loop TCGH606. This loop is noted because it is a reactor temperature loop; its performance is critical to the process performance. A loop of this importance should be doing better than “fair”. The report currently shows the criticality of this loop to be “Avg” (average). All loops default to average criticality until someone familiar with the process area specifies criticality as Low, Average, High, or Vital. In this case, this loop would be considered to have vital criticality and can be changed by the user for future reports. The criticality can be used for reference in scanning or can be used as search criteria for interactive database queries.

To get a better handle on what “fair” means, a novice user could click on the column heading above “fair” and get a pop-up help window in their web browser. A portion of the help window is shown in Fig. 5. According to the help, the score of 35 is on the low end of fair and is almost considered poor (40+). The user concludes that this loop definitely needs attention.

To investigate further, the user clicks on the TCGH606 tag name to view the detailed loop information web page. The performance history table and performance analysis from the page are shown in Fig. 6. Most of the abbreviations present in Fig. 6 have already been defined for Figs. 3 and 4. There is one new abbreviation used in the performance diagnostics table in Fig. 6, Spec. Hdw Diag (%). This is the percentage of assessments where the spectral analysis results in a diagnosis of hardware problems.

The performance history table shows a fairly consistent preliminary diagnosis of hardware problems (in

DR1 LEVEL CONTROL (LCDC801)

Area: TED_B270

Tuning History

To reorder the table, click on the desired table category.

Date	Gain %/%	Integral rpt/sec	Derivative sec	Filter smooth	By Whom	Notes
11-Oct-2001	0.50	0.00100	0.00	0.00	Unknown-Claim it!	none
09-May-1999	0.80	0.00100	0.00	0.00	Unknown-Claim it!	none
14-Jan-1999	0.80	0.00060	0.00	0.00	Unknown-Claim it!	none

Tuning Analysis

The current tuning parameter values fall within typical bounds for this type of controller. This does not necessarily mean that the tuning is optimal, just that the tuning meets some generic tuning criteria.

Performance History

Click on a table category to see a data plot. Any values in red below were flagged as an indication of potentially poor performance. Note: the most recent 10 sets of performance assessment data have been returned. You can also request the last 10, 25, 50, 75, 100 sets of performance data or view the complete performance history for this loop

Date	Mean PV	Mean OP	Mean Err (%)	OP Std Dev (%)	CLPA	Osc Diag	SP Cross
04-Oct-2001	60.01	33.96	0.98	1.64	**0.80**	hdw	103
27-Sep-2001	60.01	35.39	1.18	1.92	**0.76**	hdw	**67**
20-Sep-2001	60.01	37.23	0.76	1.24	**0.68**	hdw	**90**
26-Aug-2001	56.00	33.42	1.29	1.92	**0.83**	hdw	**63**
19-Aug-2001	55.00	32.71	0.64	1.11	0.43	hdw	281
25-May-2001	55.00	32.40	0.86	1.30	0.17	none	828
18-May-2001	54.98	34.27	0.88	1.44	0.15	dstb	823
11-May-2001	55.99	30.93	0.86	1.29	0.17	dstb	859
03-May-2001	55.99	30.98	0.81	1.42	0.12	none	923
26-Apr-2001	50.98	32.14	0.98	1.43	0.17	dstb	841

Fig. 3. Excerpt from a "complete loop information" report.

Tagname	Description	Perf. Class	Score	Crit	Mean Err	OP Std Dev	Mean OP	Mean CLPA	Pct Osc Hdw Diag	Mean SP Cross	Num Assess
FCJE747	JR3T EG SPRAY DILUTION	fair	36.3	Avg	42.01	2.32	37	0.03	0.0	2108	5
TCTE600B	T-NORTH HOT OIL LOOP	fair	35.9	Avg	0.20	9.75	25	0.88	100.0	144	3
TCQ60418B	QN04 TOP REACTOR BED TEMP	fair	35.8	Avg	5.63	8.72	200	0.68	0.0	26	2
TCGH606	GR1 DOWTHERM TEMP CONTROL	fair	35.6	Avg	0.14	9.19	65	0.84	71.4	117	7

Fig. 4. Excerpt from a performance detail report.

Fair - Loops in this category are not performing up to potential and should be improved. Control is probably being maintained in a broad sense, but deviation from setpoint is likely to be degrading process performance. The measurement may be cycling. These loops should be investigated further. Many times, the problem is not difficult to find and improvement can be obtained without a lot of effort. Occasionally, level loops with a flow smoothing objective that are performing adequately end up in this category.
(Score range 20-40)

Fig. 5. Pop-up help for performance classification.

the "Osc Diag" column). The text analysis of the performance history helps explain the tabular results for a novice user. The contents of the detailed loop information are suggesting a hardware problem.

3.4. Detailed performance assessment

The user can choose to view a detailed performance assessment of closed-loop operating data for TCGH606 by clicking on the link shown at the bottom of Fig. 6. This produces a detailed performance assessment (shown in Fig. 7) using the captured dataset from the latest automated assessment on 9 October. The main part of the detailed assessment includes: (1) a summary text section containing overall performance and problem diagnosis information, (2) time-series plots of the closed-loop setpoint, measurement, and output data, (3) numerical statistics, and (4) graphical analyses. To assist novice users in understanding the different graphs and statistics, pop-up help is available by clicking on portions of the detailed assessment. The pop-up help includes descriptions of the item and, for graphical analyses, shows example patterns and their interpretation. Graphical patterns often provide additional diagnosis information, although the user is currently responsible for matching the pattern to a diagnosis.

For the assessment shown in Fig. 7, several inferences can be made:

- The text summary section at the top pretty much confirms the conclusions formed from the loop information page (Fig. 6). The performance could probably be greatly improved and the problem seems to be hardware in the loop (only one of the tests suggests a problem other than hardware).
- The top right graph shows the joint probability distribution between the controller error and the output. The graph is shown in color for users and color is important for detecting patterns. The grayscale representation in Fig. 7 makes the pattern difficult to see. Users would see warm colors (yellow, orange, red) in a ring around the center and cool colors (blue, green) in the center and outside. This ring pattern is a strong indicator of hardware problems in self-regulating loops such as the temperature loop in question.
- The y–y plot below the large time-series plots is a zoomed section of the time-series. The computer chooses an "interesting" section of the data. The pattern in the zoomed plot suggests a hysteresis problem in the hardware. The controller output needs to move quite a bit (after changing direction) before the measurement (PV) starts to move in the same direction. Note that in the grayscale representation of the plot, the PV is the darkest line, the controller output (OP) is the lightest line, and the setpoint (SP) is the remaining line.
- The diagnosis of a disturbance problem by the spectral analysis is a result of the analysis not finding harmonic peaks in the spectrum. Typically, hardware problems will result in process nonlinearities that produce spectral harmonics. In some sense, there is a disturbance in this loop in the form of cyclic setpoint changes, but this is not the dominant feature.

The prevalence of hardware problem diagnoses from the performance history and the detailed assessment are convincing evidence that TCGH606 does indeed have a hardware problem. The fact that the spectral analysis

Date	Mean PV	Mean OP	Mean Err (%)	OP Std Dev (%)	CLPA	Osc Diag	Spec	SP Cross
09-Oct-2001	221.51	57.25	0.14	8.01	**0.77**	hdw	dstb	138
04-Oct-2001	224.44	65.25	0.12	4.53	**0.82**	hdw	tune	224
16-Sep-2001	221.01	57.80	0.14	4.72	**0.82**	hdw	hdw	123
07-Sep-2001	220.69	55.57	0.15	4.93	**0.83**	hdw	hdw	145
31-Aug-2001	221.52	57.55	0.13	4.58	**0.82**	hdw	hdw	165
23-Aug-2001	223.50	62.78	0.16	5.07	**0.76**	hdw	hdw	140
16-Aug-2001	231.82	78.39	0.13	3.98	**0.84**	hdw	hdw	143
08-Aug-2001	222.09	74.75	1.11	31.87	**0.86**	dstb	dstb	**86**
15-Jul-2001	233.30	4.02	1.54	306.89	**0.98**	dstb	dstb	**16**
01-Jul-2001	229.95	87.46	0.16	4.53	**0.84**	hdw	dstb	124

Plot performance data

Performance Analysis from 01-Jul-2001 to 09-Oct-2001

Keep in mind that the analysis below is based on an average of multiple performance assessments taken over a given time window. Current performance may be significantly different if process or other conditions have changed. Also, the analyses are based on general criteria and are not always accurate. Use this information as one of many tools to help you find and fix significant loop problems.

Basic Performance Info for TCGH606		
Parameter	**Value**	**Analysis**
No. Assess.	10	The relative number of assessments can impact the performance analysis results (e.g., performance results from just a few assessments may not accurately represent overall loop performance).
Lo/Avg./Hi PV	220.69 224.98 233.30	PV average and range give some idea of region of operation included in performance assessments.
Lo/Avg./Hi OP	4.02 60.08 87.46	OP average and range give some idea of region of operation included in performance assessments.

Performance Metrics: How is TCGH606 performing?		
Parameter	**Value**	**Analysis**
CLPA	0.83	High average CLPA value suggests that the measurement deviation from setpoint has patterns that the controller should be removing. Controller performance could be greatly improved.
Mean Error (%)	0.38	Low average error between the setpoint and measurement suggests performance is good.
Overall Score	39	This overall score suggests FAIR performance. Some aspect of performance could be greatly improved. Optimizing this loop is likely to lead to some measurable improvement in related process performance.

Performance Diagnostics: If performance is poor, what may be causing the performance problem?		
Parameter	**Value**	**Analysis**
Avg. SP Cross	131	Somewhat low average SP crossings suggests a problem such as process disturbances, sluggish tuning, or a poorly performing valve.
Osc. Hdw Diag. (%)	80.0	High value suggests potential valve/hardware problems.
Spec. Hdw Diag. (%)	50.0	High value suggests potential valve/hardware problems.
Avg. Idle Index	-0.03	An idle index in this range produces no diagnosis.

Detailed Performance Assessment

Run a detailed assessment using automated assessment data from 09-Oct-2001

Fig. 6. Excerpt from a complete loop information report for loop TCGH606.

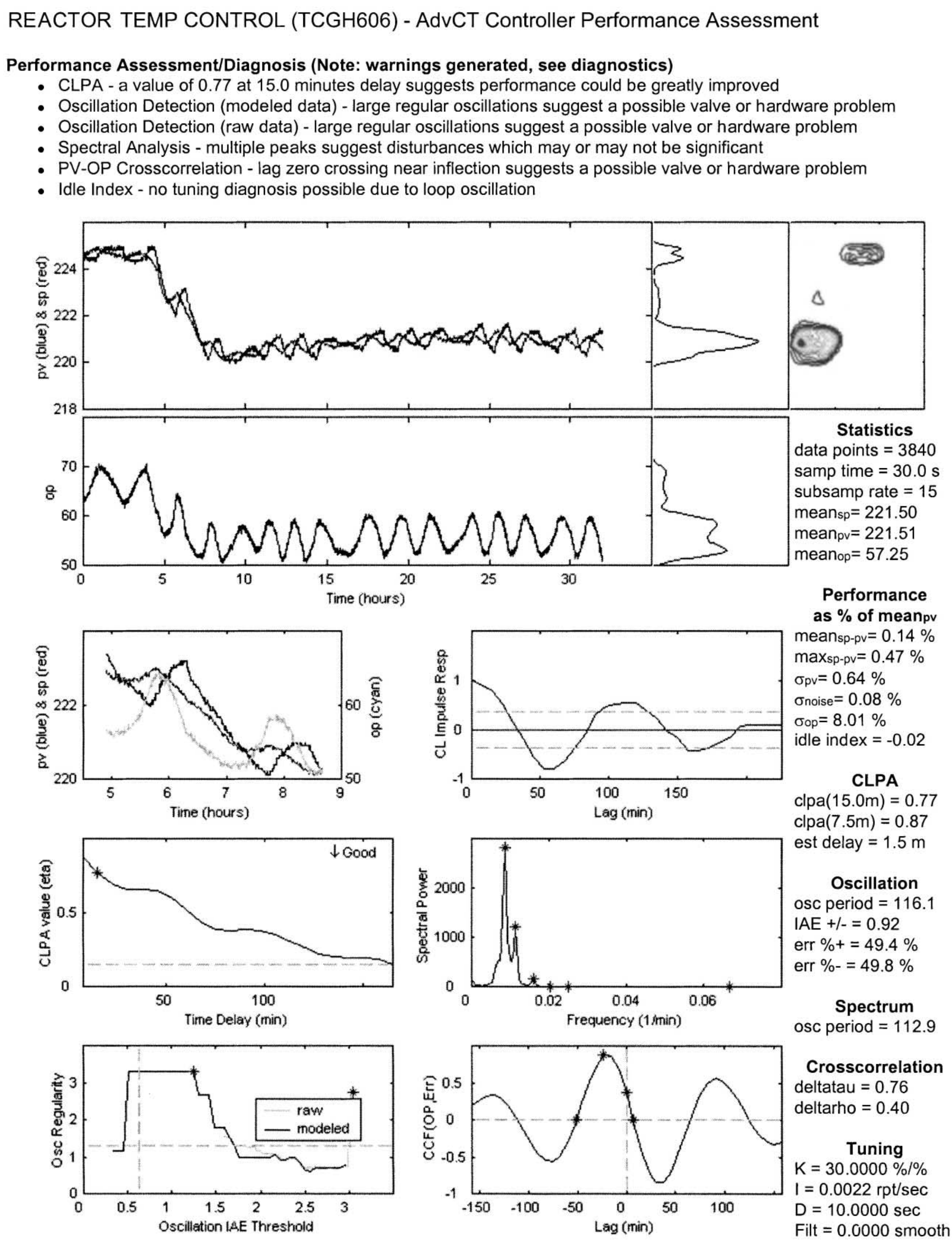


Fig. 7. Excerpt from a detailed performance assessment for loop TCGH606.

diagnosis in the detailed assessment does not agree with the majority of the other results does not cast significant doubt on the potential for a hardware problem. Rather, it is an example of how the user should review all the available information and make conclusions based on a preponderance of the evidence.

Given the importance of the loop, the logical next step is to attempt to verify the suspected hardware problem. It is worth noting that an experienced user of the system may stop at this point and conclude that there is a hardware problem and request that the valve be fixed. Such a conclusion could be aided by knowing, for example, that similar valves in temperature service in the plant have shown hysteresis problems.

3.5. Verifying the preliminary diagnosis

Assuming that the user is not completely convinced of the problem (or if a mistake in diagnosis would be very costly), they might request a preliminary field check of the control valve hardware. A quick check of the TCGH606 valve by an instrument mechanic reveals that the transducer has adequate supply pressure and reasonable output pressure. The valve is a typical sliding stem globe valve that "looks OK". The valve has no positioner. These preliminary findings are typical when trying to verify hardware problems; even loops that have significant problems often do not appear to have problems when checked in a cursory way. Occasionally, a bypassed positioner, leaking diaphragm, or other obvious problem is found, but this is relatively rare. At this point, a more rigorous check of the valve is necessary to confirm the problem. Options at this stage of the investigation include:

- Check and refine the preliminary diagnosis by recording the stem position and transducer pressure output during normal operation using appropriate instrumentation.
- Check and refine the preliminary diagnosis by performing a full field test of the valve and transducer when the process is down or the valve is bypassed.
- Perform step tests of the valve during operation in manual mode to generate more (circumstantial) evidence of a valve problem.
- Accept the preliminary diagnosis as correct and final given the overwhelming evidence in the assessment results.

Given process considerations, the user selects the full field test option, which is performed during the next process shutdown. The key result from the field test is shown in Fig. 8. The transducer shows some slight miscalibration, but the valve shows about 8% hysteresis, poor calibration, and some nonlinearity. This hysteresis measurement is right in line with the assessment results and is clearly the source of the poor performance. The valve needs to be pulled and repaired during the shutdown. In this case, a positioner should also be added for this critical loop to achieve maximum performance.

This example illustrates how easy it is to find an important problem using the performance detail report, as well as the troubleshooting time that can be saved by utilizing the historical performance and detailed controller information. The computer-generated preliminary diagnoses are not always correct. However, the chance that the user will draw good conclusions is improved by having a number of diagnoses and a number of assessments over time.

3.6. Performance summary report

The features described above are geared toward detecting and diagnosing problems with individual control loops. The performance summary report is intended to serve a complementary purpose—help users measure and track the collective performance of loops in their process area. With this monthly report, users can view trends in overall ratings of loop performance by type (flow, pressure, etc.), as well as benchmark performance of loops in their process area with similar loops in other Eastman process areas. The strengths of this report are in monitoring performance on an area-wide scale, mea-

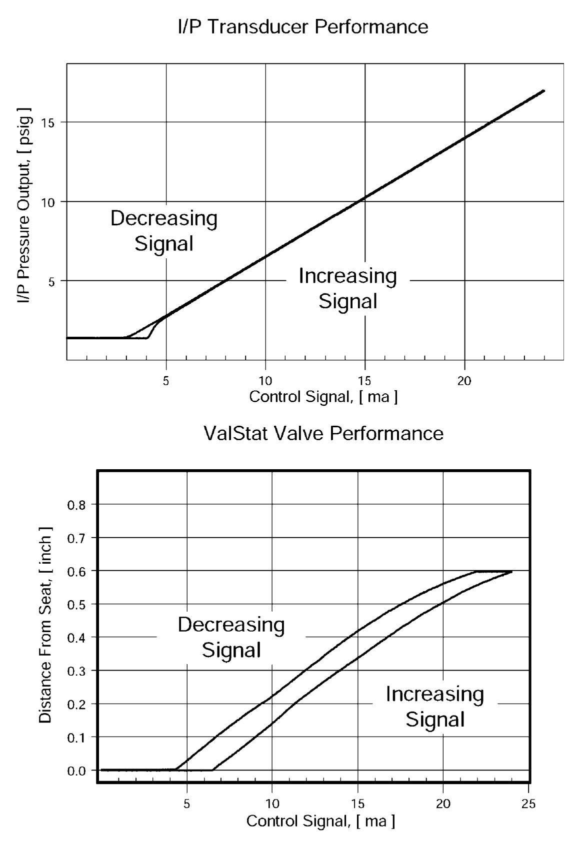


Fig. 8. Results from a field test on a valve with suspected problems.

suring the success of loop maintenance/improvement efforts, and identifying loop types needing the most improvement.

Consider the following example to illustrate the use of an actual performance summary report. Upon receiving the report, the user quickly peruses the entire report, finding a variety of graphs and statistics. She finds that over the past 3 months, about 3000 assessments have been carried out on almost 350 loops in her area. This seems interesting, but her attention is quickly drawn to the graph shown in Fig. 9, the overall performance of controllers by type over the 3-month period. A significant percentage of controllers have poor or fair rankings. The user is not sure of what the number in parentheses after the loop type means, but she reads the explanation above the graph in the report and finds that it is sort of a "grade" for the controller type as a whole, where higher numbers are better grades. It appears that many loops of each type are getting failing grades.

Another graph in the report, shown in Fig. 10, provides some encouragement. The grades for each of the loop types are plotted by month for the last year. The current performance may look bad, but the graph shows that there has been steady improvement.

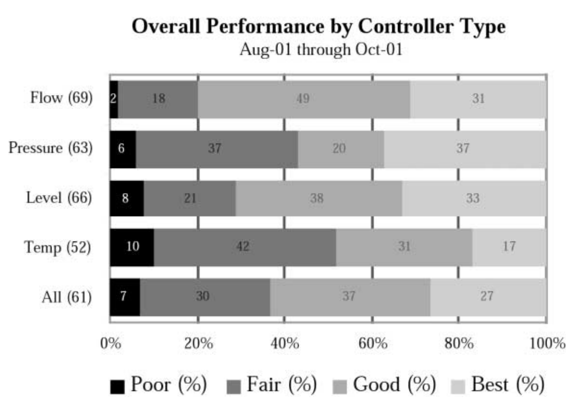


Fig. 9. Excerpt from the performance summary report showing overall performance of 350 loops of various types over a three-month period.

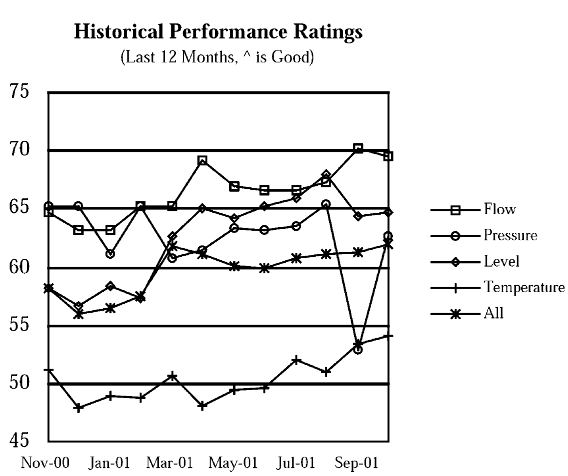


Fig. 10. Excerpt from the performance summary report showing a single-number performance statistic over time for various categories (types) of controllers.

The final section of the report contains a series of graphs like the one shown in Fig. 11, indicating how controllers of a given type from the user's process area rank against those across Eastman Chemical Company facilities. Of particular interest is the graph for pressure controllers, given that the historical performance for pressure controllers shown in Fig. 10 did not show the same improvement over time that the other types did. The user is somewhat encouraged upon seeing that her pressure controllers with a collective score of 63 are not too far away from the best-performing group of pressure controllers within the company, which have a score of 70.

This example shows the utility of the performance summary report for users concerned about area-wide controller performance. In a short time, users can gauge how the loops in their area are doing, as well as identify major improvement opportunities. Built-in to the report are measures that can be directly used to initiate and monitor loop maintenance and improvement efforts.

4. Eastman controller demographics

As of January 2002, the Eastman performance assessment system contains daily tuning history on over 14,000 controllers, and weekly performance history is available on almost 9000 controllers. These controllers originate from several brands of control systems and are operating in 40 distinct plant/process areas located at 9 different Eastman sites throughout the world. The bulk of this large-scale system has been in place since 1999.

4.1. Controller types

Fig. 12 shows the percentage of the 14,000 controllers by controller category. Not surprisingly, flow con-

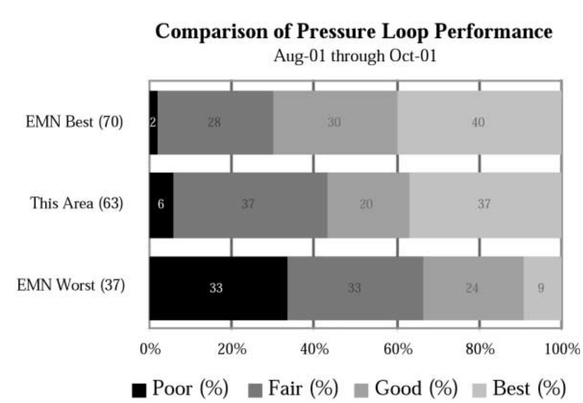


Fig. 11. Excerpt from the performance summary report showing the overall performance of pressure controllers as compared to the best and worst ranked areas within the company.

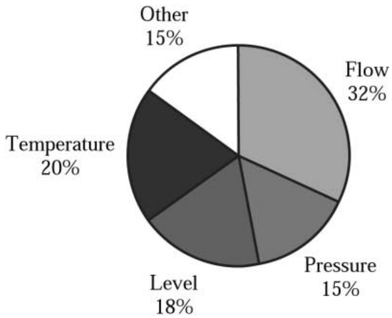


Fig. 12. Percentage of 14,000 Eastman controllers by type.

trollers make up the largest percentage of controllers. The "other" category in Fig. 12 is significant and includes analyzer controllers, power controllers, weight controllers, speed controllers, and other miscellaneous types of controllers.

4.2. Controller performance ratings

In the Eastman performance assessment system, a controller's overall performance is rated with a continuous numeric score and the scores are divided into one of four classifications: poor, fair, good, or best. The numeric score is derived from a weighted sum of the most important performance statistics and diagnostic results. For the 9000 controllers having weekly performance history, Table 1 shows the percentage of each controller type rated in each of the 4 performance classifications, along with a description of typical performance for each classification. The percentages of controllers in the poor and fair categories show the potential for controller improvement and optimization. Our experience has shown that many of these under performing controllers, especially those with a poor rating, have hardware problems (valve/positioner/transducer). Eastman has traditionally placed a low priority on loop hardware maintenance, but results from the assessment system are changing that philosophy.

Note that the performance classification percentages in Table 1 are for controllers operating in automatic. Related to this, we have found that approximately 30% of all the PID controllers at Eastman are in manual at any given time. This percentage is similar to published statistics for the process industries, which imply that a serious problem exists. At Eastman, the vast majority of controllers in manual mode are not in that mode as a result of performance problems; some are due to processes not running, some are due to a process operating mode that does not require certain controls, some are "dummy" controllers that just hold data, and others are abandoned in place due to process changes over time. This situation may not be representative for all companies, but conclusions should be drawn carefully regarding the significance of the percentage of controllers in manual. In our case, the statistic suggests more about better utilizing our supply of controllers than it does about controller performance.

Table 1
Percentage of 9000 Eastman controllers by performance classification

Class	Controller type					Class description
	Flow	Pressure	Level	Temperature	All	
Best	39	24	27	13	28	These loops are performing well and do not need attention. They are typically tracking the setpoint well, with very few or no significant deviations.
Good	35	29	30	28	31	These loops are performing adequately, but probably have some component of performance that could be improved. Benefit to cost ratio for making improvements to these controllers is likely to be small unless the tolerance for deviation from setpoint for the loop is unusually low.
Fair	23	30	34	43	31	These loops are not performing up to potential and should be improved. Control is probably being maintained in a broad sense, but deviation from setpoint is likely to be degrading process performance. These loops should be investigated further.
Poor	3	17	9	16	10	These loops typically have a serious performance problem. The loop may be cycling strongly, may have large and frequent deviations from setpoint, or may not be tracking the setpoint at all. Many of these loops may have low criticality. Improvements to critical loops in this category will often lead to significant process performance improvements.

5. Summary of system benefits and issues

Development and use of the performance assessment system has generated many benefits and revealed several key issues. These are outlined below.

5.1. Benefits summary

Before development of the performance assessment system at Eastman, poorly performing controllers were found one-by-one in areas where a process control engineer was working. Control loop maintenance (much less optimization) was reduced to reactively troubleshooting controllers that caused enough problems to make themselves obvious. The performance assessment system has ushered in a new era for us. With the performance assessment system:

- The controllers in a process area can be ranked in order of performance. Plant personnel can easily (almost effortlessly) obtain these rankings.
- A wealth of controller tuning, performance, and configuration information is instantly accessible for troubleshooting controller problems.
- The overall performance of controllers in an entire plant or process area can be easily monitored over time.
- Users can be automatically alerted about changes (e.g., controller tuning changes) that are indicative of potential problems.
- Comparison and benchmarking of analogous controllers at different Eastman plant sites is possible because of the universal user access and the large-scale assessment system.

As a result of this functionality, significant productivity gains and other benefits have been realized. The most significant benefits are:

- Huge amounts of process data are transformed into concise information valuable to a variety of plant personnel (area managers, staff engineers, etc.).
- Controllers needing attention can be identified quickly from the hundreds of controllers in process areas, making ongoing controller improvement programs feasible. More problem controllers are fixed and optimized, reducing process variability and operator intervention requirements.
- The process for troubleshooting problem controllers is streamlined, cutting troubleshooting time typically by two-thirds.
- Monitoring and benchmarking of area-wide controller performance fosters large-scale controller improvement efforts, resulting in positive economic impact on processes.
- Use of the system has strengthened communication between control engineers, staff engineers, area managers, and maintenance forces. Input from these personnel is essential to the various components of controller improvement.

The economic benefits resulting from performance assessment are difficult to quantify on a loop-by-loop basis. More often, each problem loop is contributing in a complicated way to poor overall process performance. After finding and fixing problem loops throughout a plant, 6 months to a year worth of data shows reduced off-class production, reduced product property variability, and occasionally lower operating costs or improved production rate. Controller alarms and operator interventions are also typically reduced, which enhances safety and frees some operator time for other value-adding tasks.

As an example, one Eastman process area has been using performance assessment for almost two years. The area has approximately 400 active PID controllers. Assessments identified many loops with hardware problems and approximately 40 loops have had repair or replacement of the valve, positioner, transducer, air supply, or DCS output board. Over the last year, off-class production due to process-control-related causes has been reduced by 53% (540 klbs/year). The standard deviation of the primary specification for material produced in this area has been reduced by 38%. The area has advanced from the 40th percentile to the 75th percentile of all Eastman process areas worldwide in overall controller performance. The areas above it typically are new plants with new equipment and instrumentation. Other problem loops have been identified in this process area, but have not yet been fixed pending a shutdown or availability of funds. We expect that further significant performance gains will accrue when these remaining loops are repaired.

5.2. Issues summary

A primary goal in developing the assessment system was to enable efficient application to a large number of controllers (20,000–30,000 envisioned). Any manual effort required to incorporate loops quickly becomes burdensome and costly with such a large number of loops. We have been able to design the system so it operates effectively with only the specification of a loop type. Initial configuration of the loop type is handled by the system administrators and can often be determined directly from the ISA standard tag name of the loop. The list of loops can be extracted automatically from the DCS or imported from a DCS report in text format.

Users, or process personnel, do not have to do any initial configuration. However, we have provided the ability for users to later customize individual loop configuration, as this can provide additional value. Some examples of this include the ability to specify controller criticality and to adjust weightings on how the performance statistics affect individual loop performance classifications. It is usually not difficult to get users to specify criticality or weighting since they do it naturally if presented a list of problem controllers sorted with only the default criticality and weight. Their process experience tells them which variables can (or should) float and which must be tightly controlled. If they specify their preferences once, they save themselves time in the long run as the list will be sorted more appropriately the next time they look at it.

Due in part to it being a highly automated large-scale application, the assessment system is not intended to be a complete substitute for human oversight, especially with regard to problem diagnosis. We have chosen to offer diagnosis information in a probabilistic framework. We have a number of diagnosis tests, and the results of each are presented. In some cases, the results are completely consistent, but in many cases there is some disagreement. Furthermore, we perform assessments weekly and retain the results so we have a matrix of diagnoses over time and over tests. Each of the diagnoses can be viewed as a "vote" intended to convince the user of the existence of a specific problem. We have found this approach to result in good user conclusions, especially for controllers with hardware problems. Our experience also indicates that reasonable diagnosis inaccuracy can be tolerated if there is a good relationship between the users and the performance assessment experts. In cases where the automated diagnosis does not provide a clear indication of the problem, at least the existence of a problem has been identified and field tests ranging from manual output steps to valve stem position measurement can be carried out to determine the problem.

It is worth noting that when computing an overall performance score for a loop, we also use the "matrix" approach of various statistics over a number of assessments. Since we capture the assessment data on a schedule, we will occasionally get data that include atypical operation such as startups, shutdowns, grade transitions, large upsets, etc. Assessments on this type of data will produce results that are also atypical when compared with the majority of assessments. Therefore, we use outlier detection and removal techniques to get a more robust composite score that is more representative of typical operation.

The performance assessment system is just one part of a process needed to optimize controller performance. Little gain will be realized by introducing the system without instituting an Observe-Orient-Decide-Act type of work process as well. Performance assessment brings huge efficiency increases to the "observe" and "orient" tasks, making decisions easier. However, action is critical. Where action involves repair or replacement of control loop hardware, good feedback about the benefit of this investment is needed to keep the decisions being made in favor of action. At Eastman, we are beginning to establish a culture of loop hardware maintenance as the benefits of this action are reflected in controller performance improvements that now can be measured and communicated.

6. Future enhancements

Work is ongoing to enhance the Eastman performance assessment system in the areas of problem diagnosis, closed-loop identification, and detection of plant-wide distributed disturbances.

While it seems unrealistic to completely automate controller problem diagnosis, we plan to improve problem diagnosis capability by employing nonlinear time-series signatures, automated pattern recognition, and an improved diagnostic rule base.

The number one request of users is specific guidance or recommendations for tuning parameters. We plan to improve our closed-loop model identification such that the model confidence is suitable for making tuning recommendations. Improvements in this area will also add accuracy to problem diagnoses related to poor tuning.

There is also significant interest in improving disturbance diagnosis by identifying sets of loops that appear to share a common disturbance and even identifying a loop that may be the root cause of the distributed disturbance. This capability is somewhat outside the core of performance assessment, but it integrates nicely with the spectral analysis that we already do.

One area of current research interest that is not on our short-term enhancement list is assessment of multivariable controller performance. While this would be of value, we believe that our limited resources would be better utilized in making improvements that apply to the much more common single-loop controller. We expect to be adding multivariable functionality in the longer term as multivariable controllers become more numerous and as the multivariable assessment algorithm development matures.

Acknowledgements

The authors gratefully acknowledge the support and technical input of the Eastman Advanced Controls Technology group, in particular the invaluable help of

Jim Downs, Ernie Vogel, and Joey Watson. Additionally, we acknowledge Nina Thornhill whose research and practical applications in the controller performance assessment field have been of great benefit to us.

References

[1] T.J. Harris, Assessment of control loop performance, Can. J. Chem. Eng. 67 (1989) 856–861.
[2] T.J. Desborough, L.D. Harris, Performance assessment measures for univariate feedback control, Can. J. Chem. Eng. 70 (1992) 1186–1197.
[3] Kozub, D.J. Garcia, C.E. Monitoring and diagnosis of automated controllers in the chemical process industries, Proc. AIChE Annual Meeting, St. Louis, MO, USA, 1993.
[4] N. Stanfelj, T.E. Marlin, J.F. MacGregor, Monitoring and diagnosing process control performance: the single-loop case, Ind. Eng. Chem. Res. 32 (1993) 301–314.
[5] S.J. Qin, Control performance monitoring—a review and assessment, Computers and Chem. Eng. 23 (1998) 173–186.
[6] B. Huang, S.L. Shah, Performance Assessment of Control Loops, Springer-Verlag, London, 1999.
[7] P. Jofriet, C. Seppala, M. Harvey, B. Surgenor, T. Harris, An expert system for control loop performance, Pulp and Paper Canada 97 (6) (1996) T207–T211.
[8] Kozub, D.J. Controller performance monitoring and diagnosis: experiences and challenges, in: Proc. of the Fifth International Conference on Chemical Process Control, Tahoe City, CA, USA, 1996, pp. 83–96.
[9] N.F. Thornhill, M. Oettinger, P. Fedenczuk, Performance assessment and diagnosis of refinery control loops, AIChE Symposium Series 94 (320) (1998) 373–379.
[10] N.F. Thornhill, M. Oettinger, P. Fedenczuk, Refinery-wide control loop performance assessment, J. Proc. Control 9 (1999) 109–124.
[11] N.F. Thornhill, T. Hagglund, Detection and diagnosis of oscillation in control loops, Control Eng. Practice 5 (10) (1997) 1343–1354.
[12] Miller, R.M., Timmons, C.F., Desborough, L.D. CITGO's experience with controller performance assessment, Proc. NPRA 1998 Computer Conference, San Antonio, TX, USA, 1998.
[13] T.J. Harris, C.T. Seppala, L.D. Desborough, A review of performance monitoring and assessment techniques for univariate and multivariate control systems, J. Proc. Control 9 (1999) 1–17.
[14] T. Hagglund, Automatic detection of sluggish control loops, Control Eng. Practice 7 (12) (1999) 1505–1511.

PROCESS CONTROL TUTORIALS

Batching Tutorial

What is a Flow Batch Controller? A special purpose instrument which is intended to be used in conjunction with a flow sensor and a control valve to dispense a desired amount of a fluid into a container, tank, or vehicle. In some cases the temperature may also be used to estimate the fluid density from stored fluid properties.

How does a Batcher Work? The basic batcher is illustrated in the figures below. The operator begins by entering the desired amount of fluid to be dispensed into a batch quantity set-point on the instrument. The Start button is pushed. The valve opens and the vessel begins filling. The flow sensor sends the flow signal to the batcher. The batcher compares the amount delivered and shuts the valve when the desired amount has been dispensed.

What is batch overrun and how do I prevent it? Batch over run is the term given for the amount of fluid dispensed which is greater than the setpoint which was entered. Batch overrun results from the delay in the valve closing. Two techniques are used to minimize batch overrun. See Batch Overrun Compensation and Two Stage Batching.

Batch Overrun Compensation- This technique uses a feature in some batchers which "learn" the amount of batch overrun and then seek to turn the batch off "early" by the average amount of the batch overrun. This feature may be enabled or disabled in some models.

Two Stage Batching- This technique for reducing Batch Overrun uses two valves, one slow fill and one fast fill, to reduce the flow rate just before the batch ends to reduce the amount of overrun. The user can enter the prewarn value for the slow fill at the end of the batch.

Slow Fill- This is a technique used in conjunction two stage batching where a vessel is initially filled at a slow rate to prevent splashing before the fast fill begins. The user can enter the amount of fluid to be filled during the slow fill.

Count Mode- In general, a batcher may be configured to either count from 0 up to the batch quantity or to count down from the batch quantity to 0.

Maximum Batch Preset- This is a safety feature which places a limit on the maximum batch size the operator may enter. It is intended to eliminate large operator entry errors.

Batch Auto Restart- This is a capability which may be used in some applications where the same size container will be filled repeatedly. A programmable time is allowed for the removal of the previously filled container and the repositioning of the new empty container between batches.

Flow Time Out or Security- This is a safety feature which automatically stops a batch when a loss of flow signal is encountered for longer than a user programmed time while a batch is in progress. It is intended to prevent a spill in the event of a failed flow sensor.

Drain Time- This is a feature in some batchers which delays the print of batch record for a user programmable time to permit draining of a fluid into the receiving vessel.

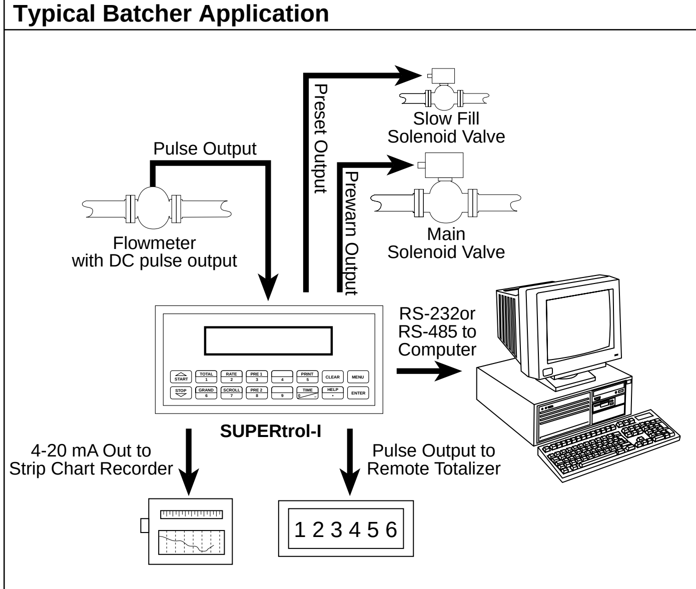
Prin**ting Capability**- Many batchers support the generation of a transaction printout. Usually a RS-232 port is provided which may be connected to a printer. A transaction print may be generated manually by pressing a PRINT key, or automatically. The format of the printout and the information which it contains are usually selectable by the user.

Print on End of Batch- This is a capability to automatically create a print out when a batch has been completed by sending out a report on a RS-232 port to a local printer.

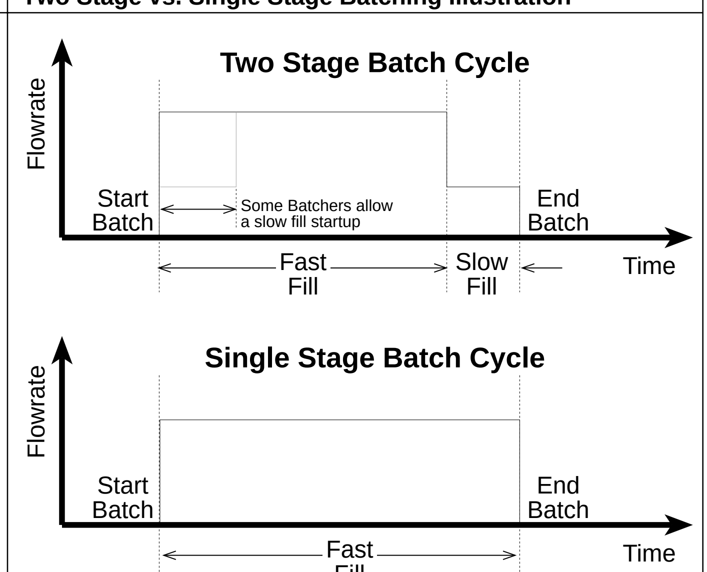
Overrun Alarm Detection- This is a safety feature which generates an alarm if the batch quantity has exceeded the desired batch size by more than the allowed limit. It is intended as a safety measure to notify the user of a malfunctioning vale which has failed to close on command..

Remote Start/Stop/Clear Capability- Many batchers have provisions for wiring remote switches or contact closure such that a remote operator or system can control the starting and stopping of a batcher.

Typical Batcher Application



Two Stage vs. Single Stage Batching Illustration



Communications Solutions Tutorial

In recent years there has been a virtual explosion of new technologies and methods which greatly simplify the exchange of information between systems. This virtual explosion in new technologies complement many of the traditional direct wiring approaches of interconnecting instruments around a plant, complex, city, or region.

KEP seeks to assist our customers in “getting connected” by using the serial communication ports provided on many of our models such that they may be used for communications with computers, for modems, for printing, for data-logging, and in wire-line and wireless communications.

KEP offers a variety of compatible hardware and software system building blocks which many users find helpful in interconnecting their instruments to their computer over their preferred communication channel.

How can I get a printed report? Many instruments may be supplied with a standard or optional RS-232 serial port which may be connected to a printer with a RS-232 serial interface. Printers are purchased separately as an accessory**.**

What information can I get on my printed report? The printing capabilities of instruments vary widely. Instruments with more advanced printing capabilities permit the user to decide on the form length, include a custom print header, time and date, sequential print number, and all the desired information. Some models include a more limited print list. Basic models support only the printing of a single number.

How can I initiate the report to print? Depending on the model being used there may be one or more ways to initiate a print. These include: Remote Print Switch, Local Print Key, End of Batch, Interval, and Time of Day.

How can I get information into my PC? There are several issues involved with getting information into a PC from an instrument. The first is the decision for the communication channel to be used. The second is the data gathering software (server). The final is the selection of the software that will display or store the information for the operator (client software).

What is a Server or DDE Server? A server is a communication utility program that you purchase which enables you to easily communicate with an instrument or PLC. Most programs offer a wizard which guides you through naming and selecting the communication channel with its com port and setting, the instruments which will be on that cable and the various measurements, or tags, being made by each instrument. Other programs will reference instrument name and tag.

How can I get information into my spread sheet? One of the simplest ways involves using a “DDE or OPC Server” which has been configured to constantly gather information from your instrument to make it available for other programs to access. (See using a dde server.) The information is accessed in the desired cell of your spread sheet by entering the following: “=KEPDDE|UNIT_NAME!DATA_ITEM_NAME”. One of the nice aspects of this approach is there is no need to write a program in many applications. A DDE server and the above command is all you need.

I want to write my own program. How do I go about it? You will need to consider using an off the shelf server or writing your own custom program in the language of your choice. Each instrument with serial communications has a special user manual which describes the format of a request for information and a list of the information. These will act as an aid while you are writing and debugging your program.

What is an HMI Software? HMI software is a software toolbox that enables a user to create custom screens for displaying information and controlling his plant. Capabilities include: controls and displays on touch screen, graphics symbols or object libraries, real time trending, data logging, and alarming. The software toolbox also includes a powerful programming or scripting language.

What do I need to get information into my Human Machine Interface Software? DDE and OPC Servers are routinely used. Alternately, custom scripting may be used in some cases.

What are the some of the common communications possibilities on the market? The choice of communications solutions available on the market is quite large. These include direct connect, wireless, fiber optic, and those which utilize the phone system. There are many others. Each technique offers advantages based on the needs of the system.

What is a hardware interface? The sender and receiver of information must match. This includes at the electrical signal level and at the low level communication settings. Industry standards exist for defining the hardware interface for signaling of information. These include RS-232, RS-422 and RS-485. There are a range of related communication settings which include baud rate, parity, start and stop bits which further clarify the interface.

What is RS-232 and how far can I send it? RS-232 is an industry standard for electrical signal levels. It is commonly used with many serial devices where the information will be send over distances not to exceed 50-200’. RS-232 ports are provided on all personal computers with a connector style known as mini-D or D-Sub.

What is RS-485 and how far can I send these signals? RS-485 is an industry standard for electrical signal levels. It is commonly used with many serial devices where the information will be send over distances not to exceed 4000’. Information is carried of 3 wires including a ground reference. RS-485 to RS-232 adapters are required to provide connections to the RS-232 ports on all personal computers.

What is a protocol? A protocol is an agreed upon method for exchanging information. It is used to decide on the method of formatting information that will be carried along a communication cable. An example would be the MODBUS-RTU protocol used on many instruments. However, there are a vast number on the market place.

What is remote metering? This may be described as any approach that is used to access information from a remote instrument to a centralized PC by connecting to and then polling an instrument for information. Telephone (modem) and wireless systems are commonly used.

What is Wireless Communications? Wireless if a term that includes a variety of technologies which do not require the sender and receiver of the system be directly connected by a wire cable. Instead a wireless transceiver is used. In common usage it may be divided into subclassifications. Common ones include wireless telephone, wireless one and two way personal messaging or paging, and radio telemetry.

What is the cost of a wireless solution? The costs of initial equipment, and installation cost vary. There is usually a monthly service charge associated with each transceiver that is based on the amount of air time, or amount of information to be transferred. There are often reduced charges for off peak hour usage.

Factory Automation Solutions Tutorial

The selection of factory automation hardware and software is a topic still quite new to many users of conventional flow instrumentation. They are presented with a increased range of possible solutions to their plant wide automation needs.

Broadly speaking there are three basic approaches to solve instrumentation and control needs. These are networks of instrumentation, or PLC based designs, or PC based designs. Each has its own merits based on the size of the plant and the need for local control.

Industrial PC's are finding their way into more and more monitoring and control applications each year. In most cases the PC is used as an operator station or data gathering station which collects information from a number of instruments or PLC's.

Many users are trying to grow their own system by looking at their need for information and tackling small portions of their plant one step at a time and slowly adding these to their existing PC network within their plant.

How will information be displayed on my PC? Generally speaking there are two broad mechanisms which are involved in the display of factory information on a PC. One program is gathering and sharing data with the display, or “client” program. The data gathering program is called a “Server/Driver”. “Client” programs include “HMI” or Human Machine Interface programs and common PC Spread Sheet and/or Database report programs. Many are available on the market.

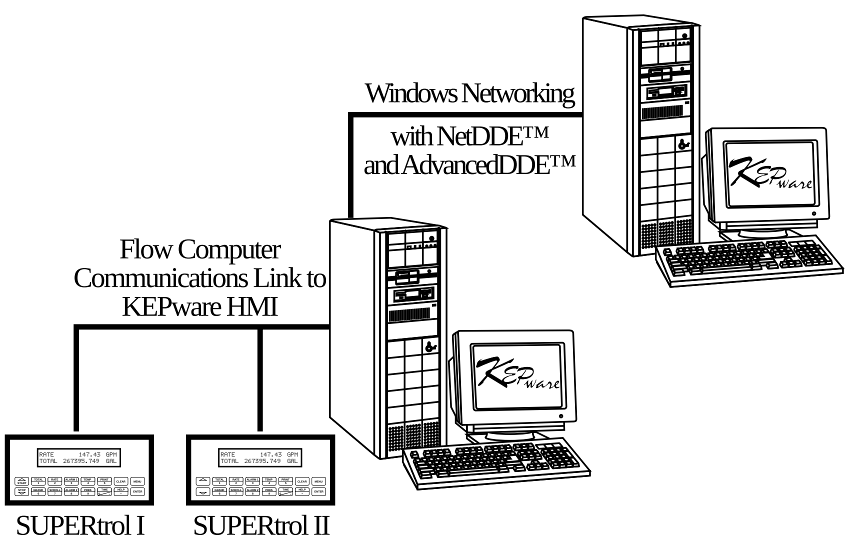
How do I select an Industrial PC? In most cases the hardware selection is done after you have decided on the software, on what you want this to do, and how it will be connected to the rest of the plant. Many experts agree that you should purchase a PC which is compatible with your software and with the best capabilities you can afford. Industrial PC offerings change frequently.

What are some of the selection criteria for Industrial PC’s? Most customers begin by reviewing the processor, memory and hardware requirements for the software they plan on using since this lists the minimum requirements for any PC they might use. Next the desired display type/size, operator input, environmental ratings, and materials of construction are reviewed. The number and type of required field and/or instrument communication channel and the desired network connection is also considered. Supplier quotations are then solicited.

What are the common field or instrument communication channels? There has been a lack of standardization in instruments and PLC’s. There are many on the market and in most plants. As a result it is not uncommon to find that several communication ports are required on your PC. Industrial PC's are usually provided with 2 or 4 RS-232 serial com ports. Instruments and PLC’s are arranged into groups that share a communication channel hardware and protocol type. Each com port is then associated with a “Server” software that knows how to gather information over that channel and how to share that information with the “client” software which is running on that or remote PC's. In some cases a “signal adapter or converter” is required to convert the COM Ports RS-232 into the signal type required by that channel. An example might be a RS-485 communication channel with several instruments which uses the MODBUS-RTU protocol would connect to COM PORT1 using a RS-232 to RS-485 adapter.

What are the common office LAN connections used in business? It is important to note that an industrial PC is after all a PC. Your system administrator will add a network card and software in the same manner as other PC’s in your office. Many Industrial PC's come with an Ethernet connection as standard or as a option.

Typical Application:



Field Indicators Tutorial

Field indicators are signal conditioner/converter devices with a display. Field Indicators are intended for mounting on or near the flow sensor. They perform many of the same roles of signal conditioner/converters plus that of providing a convenient local display.

Many “smart” Field Indicators provide additional, advanced functionality such as sensor linearization.

Field Indicators are ancillary display devices also intended to amplify, filter, condition, scale, and convert the low level “raw” signals produced by many transducers and convert it into the desired, industry standard high level signal before transmitting it across a potentially noisy environment. In some cases, a secondary function is providing signal isolation.

Generally, the output signals may be in the form of either a pulse and/or analog current/voltage that is proportional to the span of the signal being measured. Open collector transistors are common as pulse output signals. The most common analog signal is a 4-20mA current signal.

In many flowmeter types the frequency of the raw input signal carries the flow information. The frequency is related to flow rate. Each pulse or cycle is related to a small equivalent quantity of flow. The quantity represented by each pulse varies with each individual meter and must be scaled to obtain engineering units.

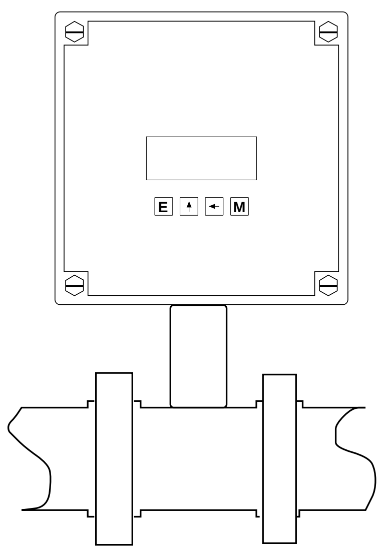
The input signal to a pulse signal conditioner may be a contact closure, a magnetic pickup, or a low level pulse. Some conditioner/converters scale the pulse signal such that each pulse represents a engineering quantity of flow, for example 1 pulse per gallon). Some converters convert the variable frequency signal into a current proportional to flow rate.

In many cases, the field indicator is intended to be powered either by an internal battery, or by the 4-20mA output current loop, or by a DC supply voltage normally available in most instruments with 24 VDC being the most common.

Enclosures are available for outdoor weatherproof and also hazardous locations. Most have provisions for mounting on the flowmeter and/or near the flowmeter.

Field Rate/Total Indicators are applied in most PLC and PC based control systems to adapt the process signals into the standardized levels provides on I/O Cards while at the same time providing a display of information in the field.

Typical Application



BAT R/T
mounted with
flowmeter

Flow Computer Tutorial

What is a flow computer? A special purpose device which computes a corrected flow based on information derived from raw input signals and stored sensor and fluid properties information

What are the typical applications requiring a flow computer? Computation of Heat Flow, Mass Flow, Corrected Volume Flow typically require a flow computer. In addition, many flow sensors require linearization to improve accuracy. The flow computer is also used for data logging, communication, remote metering, alarming and control functions. In many cases a flow computer may replace some of the functionality of a small PLC in your application.

What are typical uses of flow computers? The figures and equations below illustrate a number of the common applications for flow computers.

Where do the equations come from which are solved by the flow computer? All flow measurement sensors have basic mathematical expressions which describe how they relate the measured input signal to a flow measurement. Often there are a number of such expressions for each flowmeter type which range from the simple to those which include additional second order effects. In addition, there are basic equations from thermodynamics and industry standard equations which are utilized in liquid, gas, steam, and heat.

How can you enhance the accuracy of flow meters? A flow computer often offers a variety of performance enhancement functions. These range from simple square root functions, to more elaborate linearization tables applicable to that flowmeter type. In addition, the flow computer can correct for changes in physical dimensions of the flowmeter with temperature and for the effects of changes in fluid properties of the material being measured in some cases.

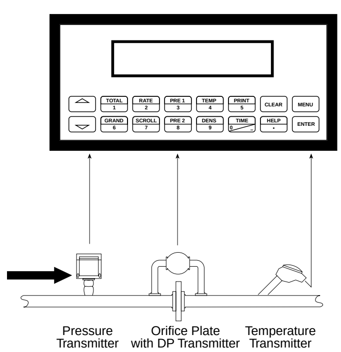
How are fluid properties determined? Fluid properties are stored within the flow computer. Properties are then computed as a function of measured fluid temperature and/or pressure. Density and viscosity are among the most commonly computed fluid properties.

What types of flowmeters typically use flow computers? The most common types used in conjunction with flow computers are turbine, vortex, positive displacement, orifice and similar types, magnetic flowmeters, and a variety of special flowmeter types. Flow computers are often used with other types when the application calls for local information display, data communications, control, alarm, and data logging functions.

What other factors should be considered? Flexibility in use of flow computation and use of inputs and outputs, signal input resolution and accuracy, isolation, 24VDC to power transmitters, networking, communications software and accessories, printing, data logging and remote metering support. Approvals may also be required. Instrument setup software is also of value. Application support from the manufacturer is also important.

Applications & Equations

Steam Mass & Steam Heat Illustration



Calculations

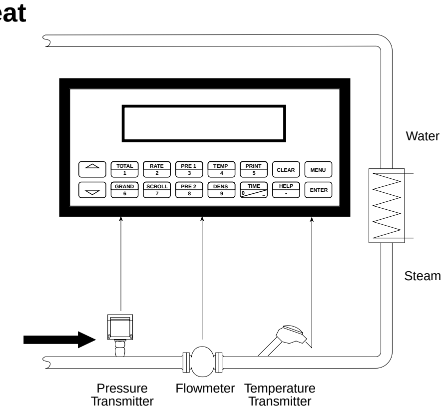
Mass Flow

Mass Flow = volume flow • density (T, p)

Heat Flow

Heat Flow = Volume flow • density (T, p) • Sp. Enthalpy of steam (T, p)

Steam Net Heat Illustration



Calculations

Net Heat Flow

Net Heat Flow =
Volume flow • density (T, p) • [E_D (T, p)– E_W ($T_{S(p)}$)]

E_D = Specific enthalpy of steam
E_w = Specific enthalpy of water
$T_{S(p)}$ = Calculated condensation temperature (= saturated steam temperature for supply pressure)

Ratemeter / Totalizer Tutorial

What is a Rate/Totalizer Indicator? This is a general purpose instrument which conditions the electrical signal generated by the flowmeter and scales the resulting flow information into a flow rate and flow total display in the units of measure desired by the end user. Additional functionality such as alarms, analog output, pulse output, and serial communications may also be provided. Also see the section on flow computers. See the figure below for a typical system configuration.

What capabilities should I look for to ensure compatibility with my type of flowmeter? Rate/Totalizers are available to work with most flowmeter types and most common electrical signals produced by flowmeters. Begin by selecting an instrument(s) that will accept the signal provided by the flowmeter. In some cases an amplifier or signal conditioner may be necessary. Next decide whether linearization will be required within the Rate/Totalizer and how the calibration will be represented within the instrument. Also determine if the Rate/Totalizer can provide the correct power required to operate the flow sensor (if needed).

What are basic areas of concern? Most customers begin a selection by looking for the instrument that has the type of information display they prefer, that will work with the available power, and is available in a package which can be mounted in the desired location.

What is an analog output and why is it used? Flow rate information is usually sent from one system to another as a 4-20mA signal. Some instruments permit the user to select what item of information is to be sent on the analog output. The corresponding span is user programmable. Additional features include programmable damping and user selectable ranges.

What is a pulse output and why is it used? Flow total information is usually sent from one system to another as a pulse which represents a quantity of flow. The remote system may sum these pulses to compute the flow total. Attributes of a pulse output include provisions for user scaling of the amount of flow each pulse represents, and the width of the pulse. Specifications will usually list the electrical drive ratings for the pulse output.

What is an alarm output and why is it used? Relays are often used as controls to activate alarms. A alarm will usually include a provision for setting the alarm point. Additional features may include a programmable delay before the alarm will activate, an programmable alarm duration, and/or a programmable alarm hysteresis.

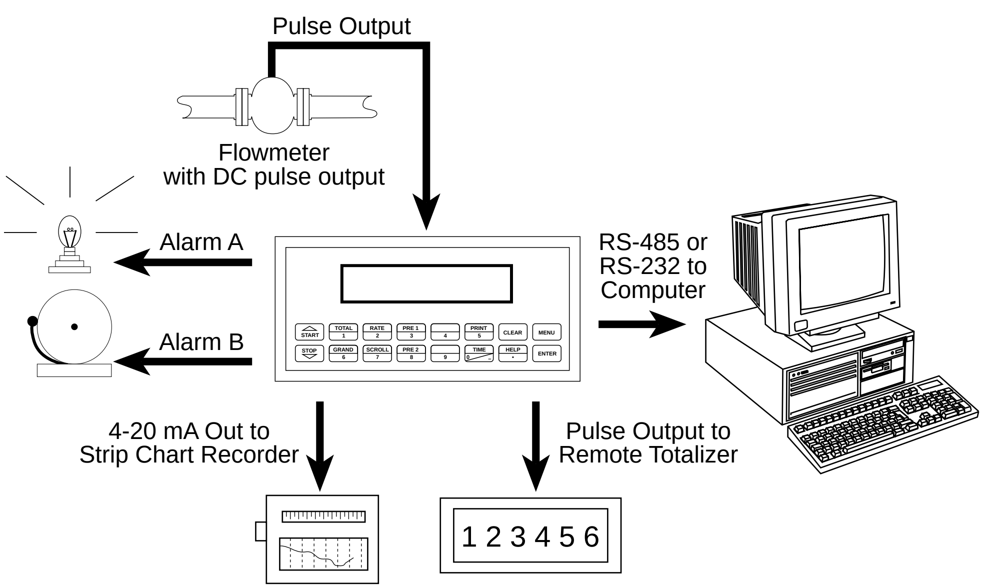
What are remote inputs and how are they used? Often there is a need to connect a remote switch near the operator for such purposes as remote reset, or remote print. Many Rate/Totalizers offer a variety of capabilities as remote inputs.

What is serial communications and why is it used? Serial communications is used to transmit information between two computers, or between a computer and a printer. There are several commonly used standard hardware interfaces. These Include RS-232, RS-422, and RS-485. There are also a variety of communication protocols, or message formats, which are used. Some of these are unique to the equipment manufacturer, others are industry standards. See also the section on communication solutions.

What is temperature compensation? In some cases the temperature may also be used to estimate the fluid density from stored fluid properties. Many customers prefer to correct their flow readings to the equivalent mass or corrected volume at a desired reference temperature.

What are other areas of concern? Many areas where rate/totalizer indicators are installed are out of doors or are located in hazardous areas. Special purpose enclosures are available for many instruments to ensure that the equipment will be protected in these environments. A NEMA-4 rating is weather proof. A NEMA-7 rating is explosion proof.

Typical Ratemeter/Totalizer Application



Signal Conditioners and Converters Tutorial

Signal conditioners, signal converters, transmitters and amplifiers are devices which represent the majority of the instrumentation requirement for transducers. They are provided with flow, temperature, pressure, as well as many other transducer sensor types.

In some cases the signal conditioner/converter is provided by the sensor manufacturer so the user will have his desired output signal.

However, in other cases, there is a need for an external signal conditioner/converter to provide the desired output signal or to provide it at a more attractive price.

Signal conditioners and converters are ancillary devices intended to amplify, filter, condition, scale, and convert the low level “raw” signals produced by many transducers and convert it into the desired, industry standard high level signal before transmitting it across a potentially noisy environment. In some cases, a secondary function is providing signal isolation.

Generally, the output signals from the sensor may be in the form of either a pulse or analog current / voltage that is proportional to the span of the signal being measured. Open collector transistors are common as pulse output signals. The most common analog signal is a 4-20mA.

In many flowmeter types the frequency of the raw input signal carries the flow information. The frequency is related to flow rate. Each pulse or cycle is related to a small equivalent quantity of flow. The quantity represented by each pulse varies with each individual meter and must be scaled to obtain engineering units.

The input signal to a pulse signal conditioner may be a contact closure, a magnetic pickup, or a low level pulse. Some conditioner/converters scale the pulse signal such that each pulse represents a engineering quantity of flow, for example 1 pulse per gallon). Some converters convert the variable frequency signal into a current proportional to flow rate.

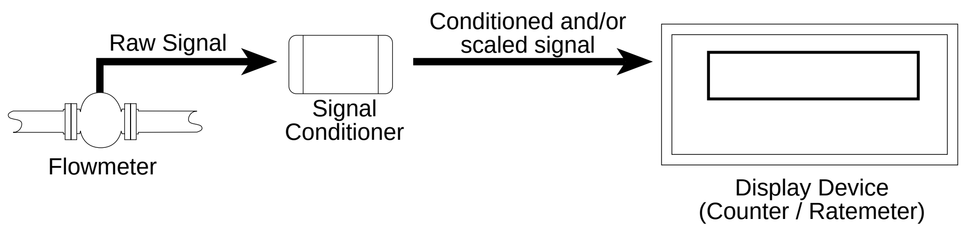
In nearly all cases the signal conditioner/converter is intended to be powered by a DC supply voltage normally available in most instruments with 24 VDC being the most common.

Enclosures are available for outdoor weatherproof and also hazardous locations.

Signal Conditioner/Converters are applied in most PLC and PC based control systems to adapt the raw process transducer signals into the standardized levels provides on I/O Cards.

Only the most common signal conditioner/converters applicable for flow metering are shown in the data sheets to follow.

Typical Application:



Steam Metering Concepts

Typical Steam Metering System
Using Differential Pressure Flowmeter System and Electronic Flow Computer

This document was created to aid in the explanation of the measurement of steam by a differential pressure (inches of water) based steam flowmeter system, and the related service operations.

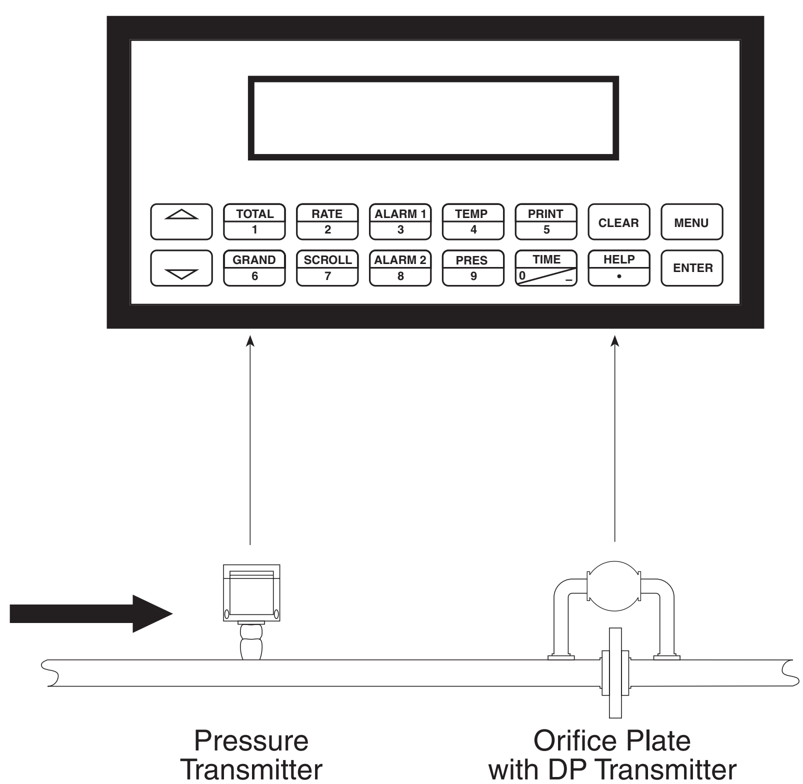
General

A typical steam-metering system consists of several components:

- Flow Meter installed in the saturated steam line
- One (or more) transmitters mounted on or near the flowmeter
- A static pressure transmitter to measure steam pressure
- A Flow Computer to compute the steam flow
- A Data Logger/Modem to monitor the customer site and provide trend information

NOTE: If remote metering is supported, a remote PC, modem, and remote metering software may be used in conjunction with this system.

HELP TIP: The piping surrounding the location where the flowmeter and related transmitters are installed is called the **"meter run".**



Equipment Selection

The flowmeter is sized by the manufacturer, based on the expected line conditions and flow rate anticipated in the application. This normally requires that the line size, steam pressure range and expected steam flow rate range be known. The differential pressure transmitter(s) is selected for steam service with a measurement range that will meet or exceed the range of differential pressure to be produced by the flowmeter. The static pressure transmitter is selected for steam service with a measurement range that will meet or exceed the range of static line pressure to be encountered in the application. The flow computer performs the necessary calculations needed to compute the steam flow from the electrical signals being fed into it. The datalogger and modem permit remote monitoring of the information fed into the data logger. The information is sent to the datalogger by either the input measurement or the computed outputs from the flow computer. A remote PC with modem can access the information either in the data logger or in the current readings of the flow computer.

Factory Calibration

The flow meter, transmitters, flow computer, and data logger are **calibrated** by their respective manufacturer's prior to being supplied to a utility company, in accordance with the instructions provided when the units are purchased.

Installation

During installation the flowmeter, differential pressure transmitter(s), and static pressure transmitters are installed in accordance with industry guidelines and manufacturers' instructions.
The individual calibration and setup documents provided by the manufacturers are reviewed.

Startup

During startup the individual components of the systems are **setup** so that they operate correctly.

For the transmitters this normally involves double-checking of each transmitter range and optional features using a hand held terminal.

The flow computer is **setup** by entering the information on the flowmeter, and the ranges for each differential and static pressure transmitter. In addition, the items to be included in the data logger are also setup. This is usually done from the front keypad, although connecting the device to a laptop and using a special program supplied by the manufacturer can also be used. Setup may also be accomplished remotely via the modem connection.

The setup of each individual item is verified. For each measurement, there is a transmitter to scale and send an electrical signal to the receivers that need this information. The scaling of each transmitter must also be set into the corresponding flow computer input channel. If a change is made to one, it must be made to all.

The basic operation of the system can be verified by checking that the respective sensors are producing the correct signals, based on the observed signal, the flow range setup in the sender and receiver of the information, and the observed process conditions in the steam line. Signal simulators and multi-meters may also be used.

There is a "low flow cutoff" that should be set to prevent the system from metering when no flow is present. This also limits the low flow measurement range, so it is usually set to the lowest practical value.

Meter Readings

Meters may be read either locally by taking a reading from the flow computer, or remotely by taking a reading from the flow computer via modem, or by reading the data logger. The operational status of the metering system is also checked periodically.

Servicing the Metering System

Often a utility will perform various inspections each year on each steam meter. Manufacturers of the components used in a steam system provide a number of service and test aids for Service personnel that permit them to interrogate a component to determine if it is operating properly. From time to time problems may occur in any system. The transmitters flow computer, and data logger usually have some **diagnostic capability** and can assist in problem detection and notification.

If it becomes necessary, for any reason, the flowmeter may be **changed out.** This sometimes occurs when the heating load changes or the actual steam range is different than the expected range as a result of inaccurate sizing information. When a change out occurs the information on the new flowmeter must also be set into corresponding transmitter and the flow computer flow input channel. If a change is made to one, the change must be made to all.

If a transmitter is changed by either **replacement, re-scaling, or re-spanning** then the new scaling of that transmitter must also be set into corresponding flow computer input channel. If a change is made to one, the change must be made to all.

Most utilities remove portions of the meter system from service after several years for recalibration. The flow computer and data logger can usually be checked in place using simulators. They can be removed from service if needed and replaced with another device that then must be setup for use as described earlier. Inputting known pressure using a special hydrostatic pressure pump may check many pressure transmitters. In other cases, the transmitters are replaced with a calibrated replacement unit

KEP Flow Computer

Kessler-Ellis Products (KEP) offers the Supertrol 2 Flow Computer for Steam Metering applications. It is available in a variety of housings to suit a wide range of application environments.

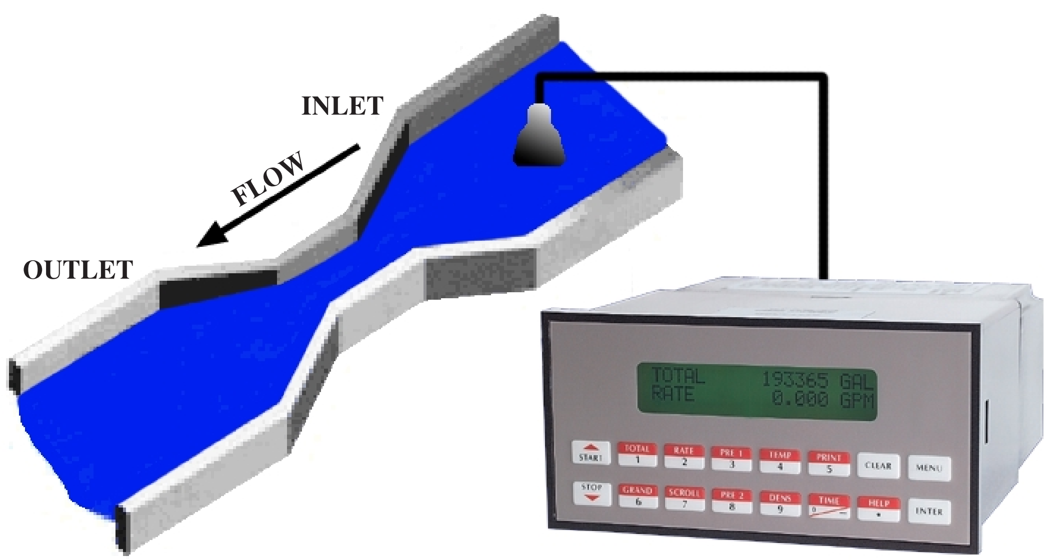
Open Channel Flow Measurement using Flumes

**Open Channel Flow Measurement Application
Using Flume System and Electronic Flow Computer**

General

The common method of measuring flow through an open channel is to measure the height or HEAD of the liquid as it passes over an obstruction (a flume or weir) in the channel. For any open channel that is free flowing through a specific controlled metering structure, there is a specific relationship between inlet height of inlet water and the flow rate. Whenever a given inlet height occurs, there will always be the corresponding flow. Therefore, if you know the flow corresponding to each inlet height, you can construct an inlet height-to-flow relationship. The water level or "Head" is accurately measured using a level sensor.

The height or level sensor outputs an electrical signal that corresponds with the height of the liquid. This signal indirectly relates to the flow rate. This relationship can be entered into the SUPERtrol-I Flow Computer using the 16 Point Linearization Table feature. The flow rate is then summed over time to display total flow.



The SUPERtrol-I can be used to measure open channel flow using flumes. We use the 16 point linearization table to convert the input level readings into their corresponding flow rates. An Excel spread sheet is available as an aid to help construct the linearization table.

REQUIRED CUSTOMER INFORMATION:
Flumes come with documentation that shows the head in inches at the flume and the corresponding flow rates. This sheet is required for each application. Details on the level sensor span and signal type that will be used are also required. See the example on the following page.

SETUP TABLE:
Below is the “flume16point.xls” application aid in the form of a spreadsheet. This will help to construct the SUPERtrol-I 16 point table. The spreadsheet is dimensionless so that it can be applied to any application regardless of the signal type, level units, or flow rate units required.

In the actual spreadsheet table there are areas in blue where you enter information and there are areas in green that represent the resulting 16 point linearization table that is created.

FLUME LINEARIZATION EXAMPLE:
Example: Signal: 4-20mA corresponding to 0-28.15”, nominal flowrate: 0 -10000 GPM (for the sample flume)

Zero Signal	4
FS Sig	20
Zero Level	0
FS Level	28.15
Zero Flow	0
FS Flow	10000

Act Head	Normalized Input	Appar. Flow	Act Flow	CF Factor	Signal In
1	0.03552	355.23979	58	0.1633	4.568
1.75	0.06217	621.66963	139	0.2236	4.995
3.5	0.12433	1243.33925	405	0.3257	5.989
5.25	0.18650	1865.00888	759	0.4070	6.984
7	0.24867	2486.67851	1180	0.4745	7.979
8.75	0.31083	3108.34813	1659	0.5337	8.973
10.5	0.37300	3730.01776	2195	0.5885	9.968
12.25	0.43517	4351.68739	2781	0.6391	10.963
14	0.49734	4973.35702	3404	0.6844	11.957
15.75	0.55950	5595.02664	4092	0.7314	12.952
17.5	0.62167	6216.69627	4812	0.7740	13.947
19.25	0.68384	6838.36590	5569	0.8144	14.941
21	0.74600	7460.03552	6373	0.8543	15.936
22.75	0.80817	8081.70515	7207	0.8918	16.931
24.5	0.87034	8703.37478	8075	0.9278	17.925
28.15	1.00000	10000.00000	10000	1.0000	20.000

TO USE THIS TABLE:
1. Begin by selecting 16 evenly spaced points from the flumes discharge table (see following page).
2. Enter the signal zero and span coming from the level transmitter (i.e.:4-20mA)
3. Enter the zero level and full scale level for the flume that correspond to the above span(i.e.:0-28.15”)
4. Enter the flow rates that correspond to the zero level and full scale level for the flume(i.e.:0-10000 GPM)
5. Enter the Actual Head (level) and Actual Flow as 16 points in ascending order of level
6. The table will be computed for you. You will be entering into the SUPERtrol-I:
   a. Setup the units of flow and time base
   b. Select the signal type coming from the level transmitter in the flow menus and also LinTbl
   c. Set in the 16 point table of Apparent Flow and Correction Factor
   d. Set in the Flow Low Scale and Flow Full Scale corresponding to the Zero Flow and FS Flow for the flume
7. Setup other features on the SUPERtrol-I as you normally would.
8. The SUPERtrol-I will now be configured for use.

EXAMPLE
Discharge From an Eighteen Inch Parshall Flume

HEAD (INCHES)	FLOW (MMGD)	FLOW (GPM)	HEAD (INCHES)	FLOW (MMGD)	FLOW (GPM)
1.00	0.08	58	14.00	4.90	3404
1.25	0.12	85	14.25	5.05	3507
1.50	0.16	112	14.50	5.19	3603
1.75	0.20	139	14.75	5.33	3700
2.00	0.25	172	15.00	5.47	3796
2.25	0.30	206	15.25	5.61	3894
2.50	0.33	232	15.50	5.75	3993
2.75	0.40	279	15.75	5.89	4092
3.00	0.45	320	16.00	6.04	4191
3.25	0.52	361	16.25	6.18	4295
3.50	0.58	405	16.50	6.33	4398
3.75	0.65	451	16.75	6.48	4501
4.00	0.72	500	17.00	6.63	4605
4.25	0.79	546	17.25	6.76	4708
4.50	0.86	596	17.50	6.93	4812
4.75	0.93	648	17.75	7.08	4915
5.00	1.01	701	18.00	7.24	5026
5.25	1.09	759	18.25	7.39	5135
5.50	1.18	817	18.50	7.55	5243
5.75	1.26	875	18.75	7.71	5352
6.00	1.34	933	19.00	7.86	5450
6.25	1.43	991	19.25	8.02	5569
6.50	1.51	1050	19.50	8.18	5683
6.75	1.61	1115	19.75	8.35	5798
7.00	1.70	1180	20.00	8.51	5912
7.25	1.79	1245	20.25	8.68	6026
7.50	1.89	1311	20.50	8.84	6140
7.75	1.95	1354	20.75	9.01	6255
8.00	2.06	1432	21.00	9.18	6373
8.25	2.17	1510	21.25	9.35	6491
8.50	2.29	1588	21.50	9.52	6609
8.75	2.39	1659	21.75	9.69	6727
9.00	2.49	1730	22.00	9.86	6845
9.25	2.60	1806	22.25	10.03	6964
9.50	2.71	1882	22.50	10.20	7085
9.75	2.82	1958	22.75	9.69	6727
10.00	2.93	2035	23.00	10.55	7328
10.25	3.05	2115	23.25	10.73	7449
10.50	3.16	2195	23.50	10.90	7571
10.75	3.28	2275	23.75	11.08	7695
11.00	3.39	2356	24.00	11.26	7821
11.25	3.51	2437	24.25	11.46	7948
11.50	3.63	2523	24.50	11.63	8075
11.75	3.76	2609	24.75	11.81	8208
12.00	3.88	2695	25.00	11.99	8328
12.25	4.01	2781	28.15	14.49	10000
12.50	4.13	2870			
12.75	4.26	2958			
13.00	4.39	3047			
13.25	4.52	3135			
13.50	4.63	3217			
13.75	4.75	3298			

Glossary of Process Control Terms

By John Gerry, P.E., ExperTune Inc.

"A to D" or A/D Converter: A to D means Analog to Digital. This electronic hardware converts an *analog* signal like voltage, electric current, temperature, or pressure into a *digital* number that a computer can process and interpret.

Auto Mode: In auto mode the controller calculates the output based its calculation using the error signal (difference between setpoint and PV). See Mode.

Anti-Reset Windup: Same as reset windup.

Closed Loop: Controller in automatic mode. See Mode.

Cascade: With 2 or more controllers. The output of the "Master" controller is the setpoint for the "Slave" controller. A classic example is the control of a reactor (a large vessel with a steel jacket around it). The product temperature (master) controller's output is the setpoint of the jacket temperature (slave) controller.

Composition: A process variable. Represents the amount of one material in a solution, or gas.

CO or Controller Output: Same as output.

Corner Frequency: For first order time constants, the "corner frequency" is the frequency where the amplitude ratio starts to turn and the phase lag equals 45 degrees. Also:

corner frequency = 1/(time constant) radians/time

DDE Windows Dynamic Data Exchange. A standard software method for communicating between applications under Microsoft Windows. Created by Microsoft starting with Windows 3.1. DDE is being replaced by OLE for process control, OPC.

Dead Time: Dead time is the amount of time that it takes for your process variable to *start* changing after your valve changes. If you were taking a shower, the dead time is the amount of time it would take for you (the controller) to feel a change in temperature after you have adjusted the hot or cold water.

Pure dead time processes are usually found in plug flow or solids transportation loops. Examples are paper machine and conveyor belt loops. Dead time is also called delay. A controller cannot make the process variable respond before the process dead time.

To a controller, a process may appear to have more dead time than what it actually has. That is, the controller cannot be tuned tight enough (without going unstable) to make the process variable respond appreciably before an **equivalent dead time**. More accurately, the characteristic time of the loop is determined by equivalent dead time. Equivalent dead time consists of pure dead time plus process components contributing more than 180 degrees of phase lag.

The phase of dead time increases proportionally with frequency. Any process having more than 180 degrees phase lag has equivalent dead time.

Derivative: The "D" part of PID controllers. With derivative action, the controller output is proportional to the rate of change of the process variable or error. Some manufacturers use the term rate or pre-act instead of derivative. Derivative, rate, and pre-act are the same thing. Derivative action can compensate for a changing process variable. Derivative is the "icing on the cake" in PID control, and most people don't use it. It can make the controller output jittery on a noisy loop and most people don't use derivative on noisy loops for this reason. See presentation on Derivative Action, the Good, the Bad, and the Ugly.

Delay: This term is often used in place of dead time. See dead time.

DCS: Digital Control System. DCS refers to larger analog control systems like Fisher, Foxboro, Honeywell, and Bailey systems. DCSs were traditionally used for PID control in the process industries, whereas PLCs were used for discrete or logic processing. However, PLCs are gaining capability and acceptance in doing PID control. Most utilities, refineries and larger chemical plants use DCSs. These systems cost from 20 thousand to millions of dollars.

Discrete Logic: Refers to digital or "on or off" logic. For example, if the car door is open and the key is in the ignition, then the bell rings.

Discrete I/O: Senses or sends either "on or off" signals to the field. For example a discrete input would sense the position of a switch. A discrete output would turn on a pump or light.

Dominant Dead Time Process: If the dead time is larger than the lag time the process is a dominant dead time process.

Dominant Lag Process: Most processes consist of both dead time and lag. If the lag time is larger than the dead time, the process is a dominant lag process. Most process plant loops are dominant lag types. This includes most temperature, level, flow and pressure loops.

Error: Error = setpoint - PV. In auto mode, the controller uses the error in its calculation to find the output that will get you to the setpoint.

Equivalent Dead Time: To a controller, a process may appear to have more dead time than what it actually has. That is, the controller cannot be tuned tight enough (without going unstable) to make the process variable respond appreciably before an **equivalent dead time**. More accurately, the characteristic time of the loop is determined by equivalent dead time consisting of pure dead time plus process components contributing more than 180 degrees of phase lag.

The phase of dead time increases proportionally with frequency. Any process having more than 180 degrees phase lag has equivalent dead time.

Gain (of the controller): This is another way of expressing the "P" part of the PID controller. GAIN = 100/(Proportional Band). The more gain a controller has the faster the loop response and more oscillatory the process.

Gain (of the process): Gain is defined as the change in input divided by the change in output. A process with high gain will react more to the controller output changing. For example, picture yourself taking a shower. You are the controller. If you turned the hot water valve up by half a turn and the temperature changed by 10 degrees this would be a higher gain process than if the temperature changed only 3 degrees.

Gain Margin: The difference in the logarithms of the amplitude ratios at the frequency where the combined phase angle is 180 degrees lag is the GAIN MARGIN.

Hysteresis: In a valve with loose linkages, the air signal to the valve will have to change by an amount equal to the hysteresis before the valve stem will move. Once the valve has begun to move in one direction it will continue to move if the air signal keeps moving in the same direction. When the air signal reverses direction, the valve will not move until the air signal has changed in the new direction by an amount equal to the hysteresis.

I/O: Input/Output. Refers to the electronic hardware where the field devices are wired. Discrete I/O would have switches for inputs and, solenoid valves and pumps for outputs. Analog I/O would have process variable inputs, and controller outputs.

Integral Action: The "I" part of the PID controller. With integral action, the controller output is proportional to the amount **and** duration of the error signal. If there is more integral action, the controller output will change more when error is present. If your units on integral are in "time/rep" or "time" then decreasing your integral setting will increase integral action. If your units on integral are in "rep/time or "1/time" then increasing your integral setting increases integral action.

Load Upset: An upset to the process (that is not from changing the set-point). A simple example: you are taking a shower and someone flushes the toilet. The temperature suddenly changes on you, the controller. Another example: you are injecting steam into flowing cold water to get lukewarm water, and the inlet cold water changes temperature.

Lag Time: Lag time is the amount of time *after the dead time* that the process variable takes to move 63.3% of its final value after a step change in valve position. Lag time is also called a capacity element or a first order process. Very few real processes are pure lag. Almost all real processes contain some dead time.

Measurement: Same as "process variable."

Manual Mode: In manual mode, the user sets the output. See Mode.

Mode: Auto, manual, or remote. In auto mode the controller calculates the output based its calculation using the error signal (difference between setpoint and PV). In manual mode, the user sets the output. In remote, the controller is actually in auto but gets its setpoint from another controller.

MMI: Man Machine Interface. Refers to the software that the process operator "sees" the process with. An example MMI screen may show you a tank with levels and temperatures displayed with bar graphs and values. Valves and pumps are often shown and the operator can "click" on a device to turn it on, off or make a setpoint change. Examples are Intellution's FIX DMACS, Wonderware's Intouch, Genesis's ICONICS, TA Engineering's AIMACS, and Intec's Paragon.

Open Loop: Controller in manual mode. See Mode.

OPC or OLE for Process Control is a standard set by the OPC Foundation for fast and easy connections to controllers. ExperTune Inc., is an OPC Foundation Member.

Output: Output of the PID controller. In auto mode the controller calculates the output based its calculation using the error signal (difference between setpoint and PV). In manual mode, the user sets the output.

Phase Margin: The difference in phase at the frequency where the combined process and controller amplitude ratio is 0 is the PHASE MARGIN.

PID Controller: Controllers are designed to eliminate the need for continuous operator attention. Cruise control in a car and a house thermostat are common examples of how controllers are used to automatically adjust some variable to hold the process variable (or process variable) at the set-point. The set-point is where you would like the process variable to be. Error is defined as the difference between set-point and process variable.

$$(\text{error}) = (\text{set-point}) - (\text{process variable})$$

The output of PID controllers will change in response to a ***change*** in process variable or set-point.

pH: A measure of how acidic or basic a solution is. pH is often a process variable to control.

PLC: Programmable Logic Controller. These computers replace relay logic and usually have PID controllers built into them. PLCs are very fast at processing discrete signals (like a switch condition). The most popular PLC manufacturer's are Allen Bradley, Modicon, GE, and Siemens (or TI).

PV or Process Variable: What you are trying to control: temperature, pressure, flow, composition, pH, etc. Also called the measurement.

Proportional Band: The "P" of PID controllers. With proportional band, the controller output is proportional to the error or a change in process variable. Proportional Band = 100/Gain.

Proportional Gain: This is the "P" part of the PID controller. See gain. (of the controller). (Proportional gain)=100/(Proportional Band).

Rate: Same as the derivative or "D" part of PID controllers.

Register: A storage location in a PLC. The ExperTune PID Tuner needs to know certain register addresses to tune loops in PLCs.

Regulator: When a controller changes a process variable to move the process variable back to the setpoint, it is called a regulator.

Reset: Same as the integral or "I" part of PID controllers.

Reset Windup: With a simple PID controller, integral action will continue to change the controller output value (in voltage, air signal or digital computer value) after the actual output reaches a physical limit. This is called reset (integral) windup. For example, if the controller is connected to a valve which is 100% open, the valve cannot open farther. However, the controller's calculation of its output can go past 100%, asking for more and more output even though the hardware cannot go past 100%. Most controllers use an "anti-reset windup" feature that disables integral action using one of a variety of methods when the controller hits a limit.

Robust: A loop that is robust is relatively insensitive to process changes. A less robust loop is more sensitive to process changes. See a presentation on Loop Stability, The Other Half of the PID Tuning Story

Sample Interval: The rate at which a controller samples the process variable, and calculates a new output. Ideally, the sample interval should be set between 4 and 10 times faster than the process dead time. See a presentation on What Sample Interval Should I Use?

Set-Point: The set-point is where you would like the process variable to be. For example, the room you are in now has a setpoint of about 70 degrees. The desired temperature you set on the thermostat is the setpoint.

Servo: When a controller changes a process variable to move the process variable in response to a setpoint change, it is called a servo.

Time Constant: Same as lag time.