# The PID Control Algorithm

## How it works and how to tune it

John A. Shaw
Process Control Solutions
November 7, 2001

John A. Shaw is a process control engineer and president of Process Control Solutions. An engineering graduate of N. C. State University, he previously worked for Duke Power Company in Charlotte, N. C. and for Taylor Instrument Company (now part of ABB, Inc.) in Rochester, N. Y. He is the author of over 20 articles and papers and continues to live in Rochester.

## Table of Contents

## Table of Figures

# 1 INTRODUCTION

Process control is the measurement of a process variable, the comparison of that variables with its respective set point, and the manipulation of the process in a way that will hold the variable at its set point when the set point changes or when a disturbance changes the process.

An example is shown in Figure 1. In this case, the temperature of the heated water leaving the heat exchanger is to be held at its set point by manipulating the flow of steam to the exchanger using the steam flow valve. In this example, the temperature is known as the *measured* or *controlled variable* and the steam flow (or the position of the steam valve) is the *manipulated variable*.

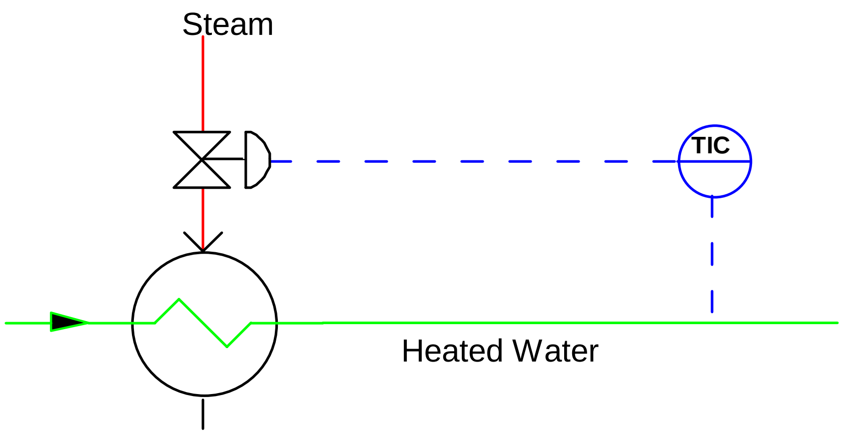



**Figure 1 Typical process control loop – temperature of heated water.**

Most processes contain many variables that need to be held at a set point and many variables that can be manipulated. Usually, each measured variable may be affected by more than one manipulated variable and each manipulated variable may affect more than one controlled variable. However, in most cases manipulated variables and measured variables are paired together so that one manipulated variable is used to control one measured variable. Each pair of measured variable, manipulated variable, the control algorithm is referred to as a *control loop*. The decision of which variables to pair is beyond the scope of this publication but is discussed elsewhere.

In some cases control loops may involve multiple inputs from the process and multiple outputs to the processes. However, in this publication, we will examine control loops with a single input and single output, by far the most common in industry.

There are a number of algorithms that can be used to control the process. The most common is the simplest: an on/off switch. For example, most appliances use a thermostat to turn the heat on when the temperature falls below the set point and then turn it on when the temperature reaches the set point. This results in a cycling of the temperature above and below the set point but is sufficient for most common home appliances and some industrial equipment.

To obtain better control there are a number of mathematical algorithms that compute a change in the output based on the controlled variable. Of these, by far the most common is known as the PID (Proportional, Integral, and Derivative) algorithm, on which this publication will focus.

First we will look at the PID algorithm and its components. We will then look at the dynamics of the process being controlled. Then we will review several methods of tuning (or adjusting the parameters of) the PID control algorithm.

## THE CONTROL LOOP

The process control loop contains the following elements:

- The measurement of a process variable. A sensor, more commonly known as a transmitter, measures some variable in the process such as temperature, liquid level, pressure, or flow rate, and converts that measurement to a signal (typically 4 to 20 ma.) for transmission to the controller or control system.

- The control algorithm. A mathematical algorithm inside the control system is executed at some time period (typically every second or faster) to calculate the output signal to be transmitted to the final control element.

- A final control element. A valve, air flow damper, motor speed controller, or other device receives a signal from the controller and manipulates the process, typically by changing the flow rate of some material.

- The process. The process responds to the change in the manipulated variable with a resulting change in the measured variable. The dynamics of the process response are a major factor in choosing the parameters used in the control algorithm and are covered in detail in this publication.

The interconnection of these elements is illustrated in Figure 2.

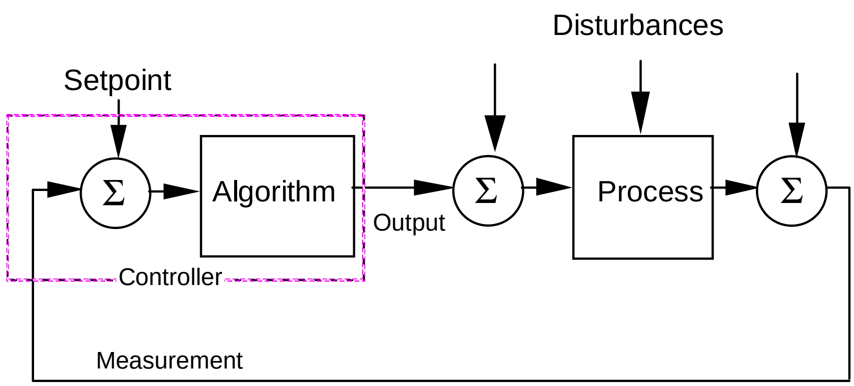



**Figure 2 Interconnection of elements of a control loop.**

The following signals are involved in the loop:

- The *process measurement*, or *controlled* variable. In the water heater example, the controlled variable for that loop is the temperature of the water leaving the heater.

- The *set point*, or the value to which the process variable will be controlled.

- *One or more load variables*, not manipulated by this control loop, but perhaps manipulated by other control loops. In the steam water heater example, there are several load variables. The flow of water through the heater is one that is likely controlled by some other loop. The temperature of the cold water being heated is a load variable. If the process is outside, the ambient temperature and weather (rain, wind, sun, etc.) are load variables outside of our control. A change in a load variable is a *disturbance*.

Other measured variables may be displayed to the operator and may be of importance, but are not a part of the loop.

## ROLE OF THE CONTROL ALGORITHM

The basic purpose of process control systems such as is two-fold: To manipulate the final control element in order to bring the process measurement to the set point whenever the set point is changed, and to hold the process measurement at the set point by manipulating the final control element. The control algorithm must be designed to quickly respond to changes in the set point (usually caused by operator action) and to changes in the loads (disturbances). The design of the control algorithm must also prevent the loop from becoming unstable, that is, from oscillating.

## AUTO/MANUAL

Most control systems allow the operator to place individual loops into either manual or automatic mode.

In manual mode the operator adjusts the output to bring the measured variable to the desired value. In automatic mode the control loop manipulates the output to hold the process measurements at their set points.

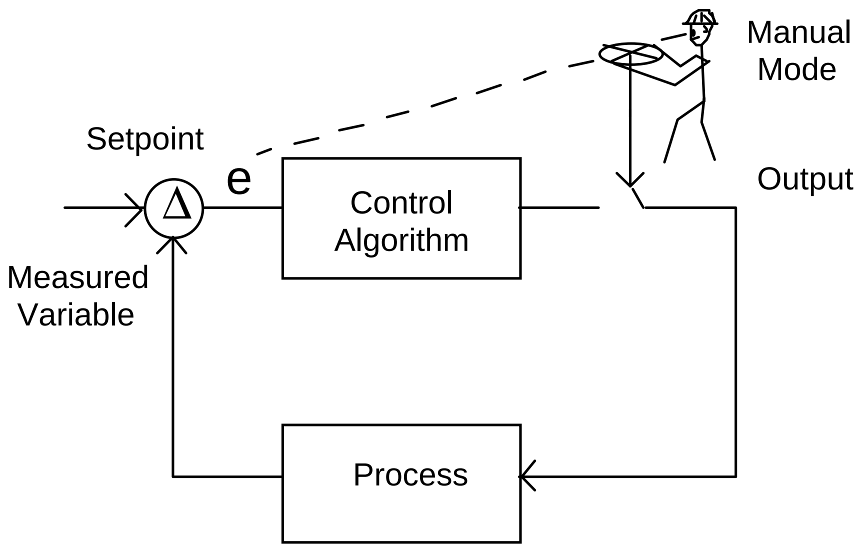



**Figure 3 A control loop in manual.**

In most plants the process is started up with all loops in manual. During the process startup loops are individually transferred to automatic. Sometimes during the operation of the process certain individual loops may be transferred to manual for periods of time.

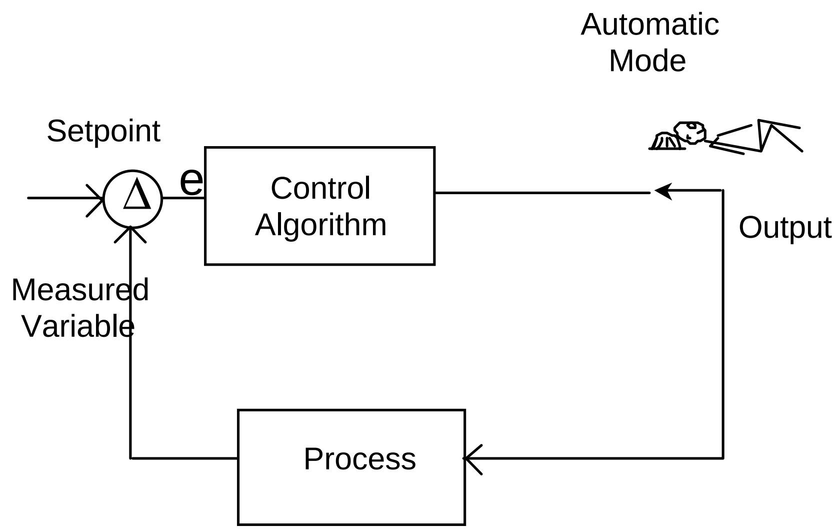



**Figure 4 A control loop in automatic**

# 2 THE PID ALGORITHM

In industrial process control, the most common algorithm used (almost the only algorithm used) is the time-proven PID**—P**roportional, **I**ntegral, **D**erivative**—** algorithm.

In this section we will look at how the PID algorithm works from both a mathematical and an implementation point of view.

## KEY CONCEPTS

• **The PID control algorithm does not "know" the correct output that will bring the process to the set point.**

The PID algorithm merely continues to move the output in the direction that should move the process toward the set point until the process reaches the set point. The algorithm must have feedback (process measurement) to perform. If the loop is not closed, that is, the loop is in manual or the path between the output to the input is broken or limited, the algorithm has no way to "know" what the output should be. Under these (open loop) conditions, the output is meaningless.

• **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 and is will be discussed in a future publication.

## ACTION

The most important configuration parameter of the PID algorithm is the *action*. Action determines the relationship between the direction of a change in the input and the resulting change in the output. If a controller is *direct acting* an increase in its input will result in an increase in its output. With *reverse action* an increase in its input will result in a decrease in its output.

The controller action is always the opposite of the process action.

## THE PID RESPONSES

The PID control algorithm is made of three basic responses, Proportional (or gain), integral (or reset), and derivative. In the next several sections we will discuss the individual responses that make up the PID controller.

In the following discussion we will use the term called "error" that is the difference between the process and the set point. If the controller is direct acting, the set point is subtracted from the measurement; if reverse acting the measurement is subtracted from the set point. Error is always in percent.

Error = Measurement-Set point (Direct action)

Error = Set point-Measurement (Reverse action)

## PROPORTIONAL

The most basic response is proportional, or gain, response. In its pure form, the output of the controller is the error times the gain added to a constant known as "manual reset".

Output = E x G + k

where:

Output = the signal to the process
E = error (difference between the measurement and the set point.
G = Gain
k = manual reset

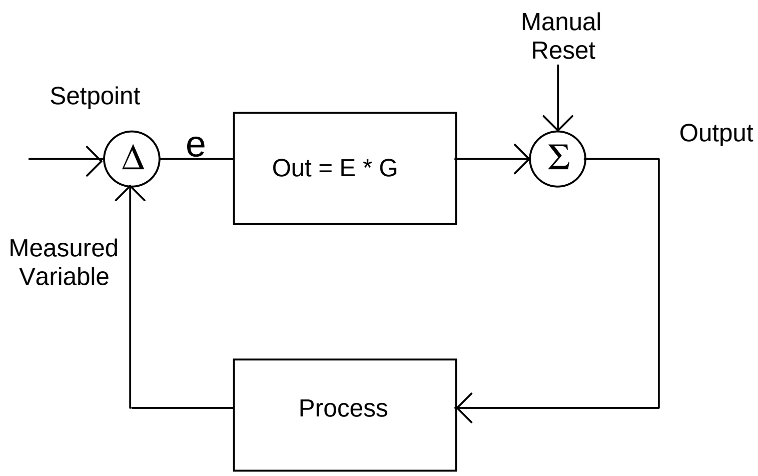



**Figure 5 A control loop using a proportional only algorithm.**

The output is equal to the error time the gain plus manual reset. A change in the process measurement, the set point, or the manual reset will cause a change in the output. If the process measurement, set point, and manual reset are held constant the output will be constant

Proportional control can be thought of as a lever with an adjustable fulcrum. The process measurement pushes on one end of the lever with the valve connected to the other end. The position of the fulcrum determines the gain. Moving the fulcrum to the left increases the gain because it increases the movement of the valve for a given change in the process measurement.

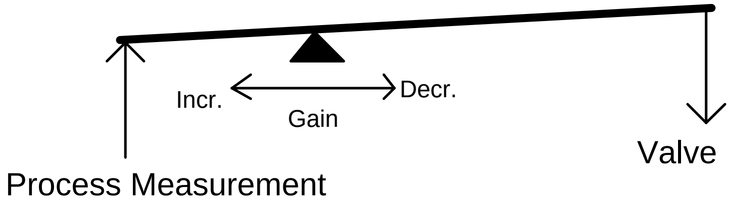



**Figure 6 A lever used as a proportional only reverse acting controller.**

## PROPORTIONAL—OUTPUT VS. MEASUREMENT

One way to examine the response of a control algorithm is the open loop test. To perform this test we use an adjustable signal source as the process input and record the error (or process measurement) and the output.

As shown below, if the manual reset remains constant, there is a fixed relationship between the set point, the measurement, and the output.

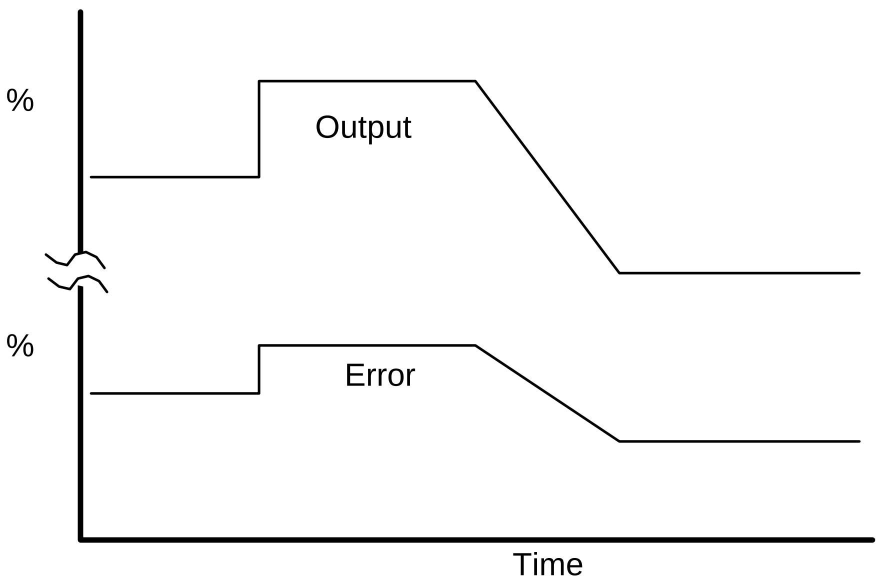



**Figure 7 Proportional only controller error vs. output over time.**

## PROPORTIONAL—OFFSET

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

Take, for example, this tank with liquid flowing in and flowing out under control of the level controller. The flow in is independent and can be considered a load by the level control.

The flow out is driven by a pump and is proportional to the output of the controller.

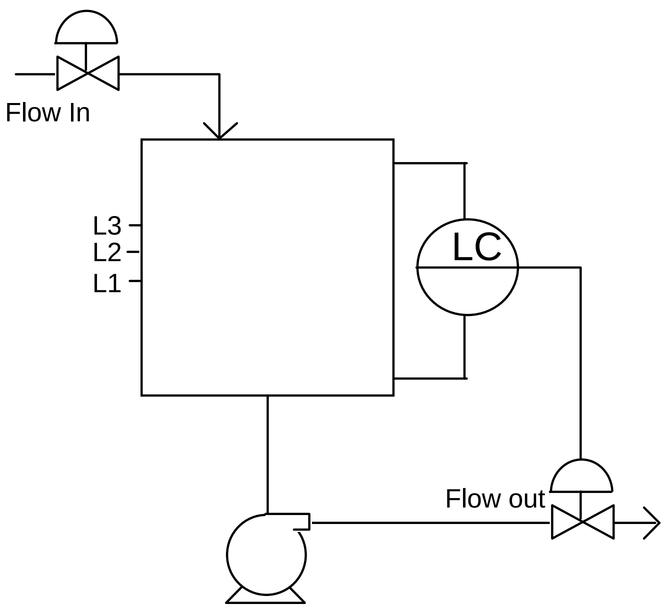



**Figure 8 Proportional only level control.**

Assume first that the level is at its set point of 50%, the output is 50%, and both the flow in and the flow out are 500 gpm. Then let's assume the flow in increases to 600 gpm. The level will rise because more liquid is coming in than going out. As the level increases, the valve will open and more flow will leave. If the gain is 2, each one percent increase in level will open the valve 2% and will increase the flow out by 20 gpm. Therefore by the time the level reaches 55% (5% error) the output will be at 60% and the flow out will be 600 gpm, the same as the flow in. The level will then be constant. This 5% error is known as the offset.

Offset can be reduced by increasing gain. Let's repeat the above "experiment" but with a gain of 5. For each 1% increase in level will increase the output by 5% and the flow out by 50 gpm. The level will only have to increase to 52% to result in a flow out of 600 gpm and cause the level to be constant. Increasing the gain from 2 to 5 decreases the offset from 5% to 2%. However, only an infinite gain will totally eliminate offset.

Gain, however, cannot be made infinite. In most loops there is a limit to the amount of gain that can be used. If this limit is exceeded the loop will oscillate.

## PROPORTIONAL—ELIMINATING OFFSET WITH MANUAL RESET

Offset can also be eliminated by adjusting manual reset. In the above example (with a gain of two) if the operator increased the manual reset the valve would open further, increasing the flow out. This would cause the level to drop. As the level dropped, the controller would bring the valve closed. This would stabilize the level but at a level lower than before. By gradually increasing the manual reset the operator would be able to bring the process to the set point.

## ADDING AUTOMATIC RESET

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

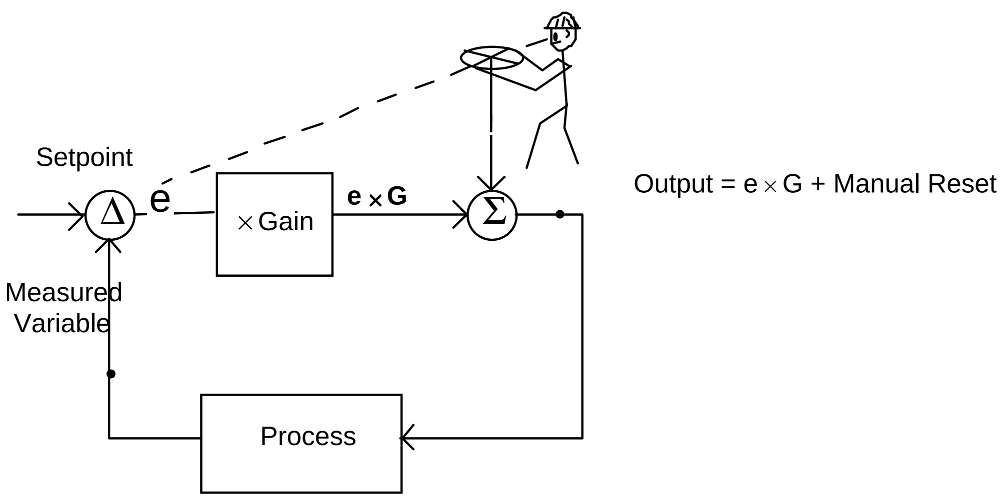



**Figure 9 The operator may adjust the manual reset to bring the measurement to the set point, eliminating the offset.**

If the process is to be held at the set point the manual reset must be changed every time there is a load change or a set point change. With a large number of loops the operator would be kept busy resetting each of the loops in response to changes in operating conditions.

This manual reset may be replaced by **automatic reset** that will continue to move the output whenever there is any error:

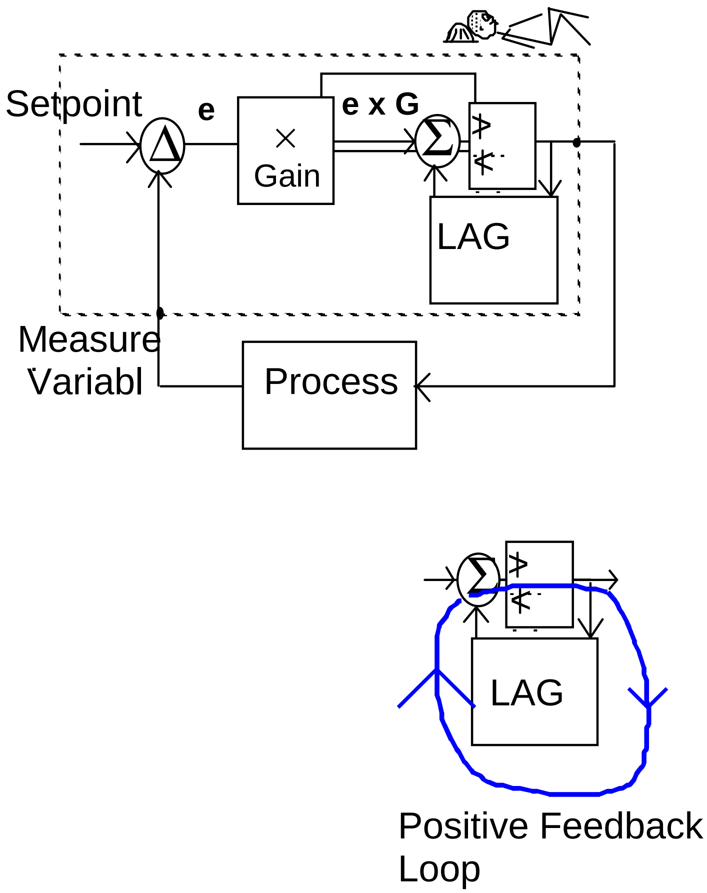



**Figure 10 Addition of automatic reset to a proportional controller. The positive feedback loop will cause the output to ramp when ever the error is not zero.**

This is called "**Reset**" or **Integral Action**. Note the use of the positive feedback loop to perform integration. As long as the error is zero, the output will be held constant. However, if the error is non-zero the output will continue to change until it has reached a limit.

## INTEGRAL MODE (RESET)

If we look only at the reset (or integral) contribution from a more mathematical point of view, the reset contribution is:

$$\mathbf{Out = g \times K_r \times \int e\, dt}$$

where $\mathbf{g}$ = gain

$\mathbf{K_r}$ = reset setting in repeats per minute.

At any time the rate of change of the output is the gain time the reset rate times the error. When the error is zero the output does not change; if the error is positive the output increases.

Shown below is an open loop trend of the error and output. We would obtain this trend if we recorded the output of a controller that was not connected to a process while we manipulated the error.

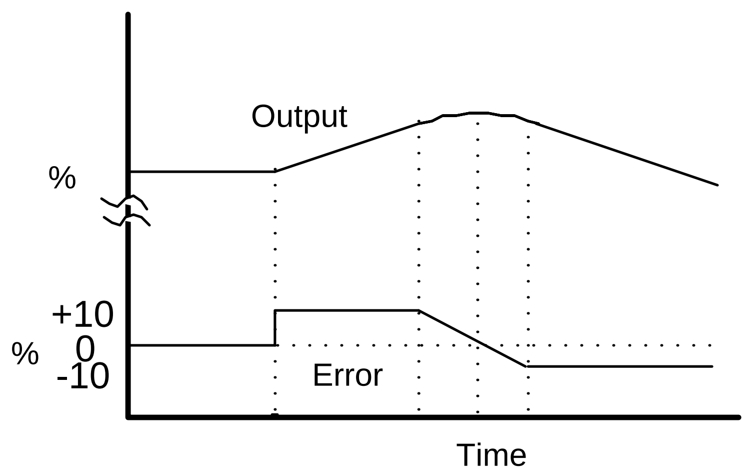



**Figure 11 Output vs. error over time.**

## CALCULATION OF REPEAT TIME

Most controllers use both proportional action (gain) and reset action (integral) together. The equation for the controller is:

$$\mathbf{Out = g\,(\,e + K_r \int e\,dt\,)}$$

where $\mathbf{g}$ = gain

$\mathbf{K_r}$ = reset setting in repeats per minute.

If we look an open loop trend of a PI controller after forcing the error from zero to some other value and then holding it constant, we will have:

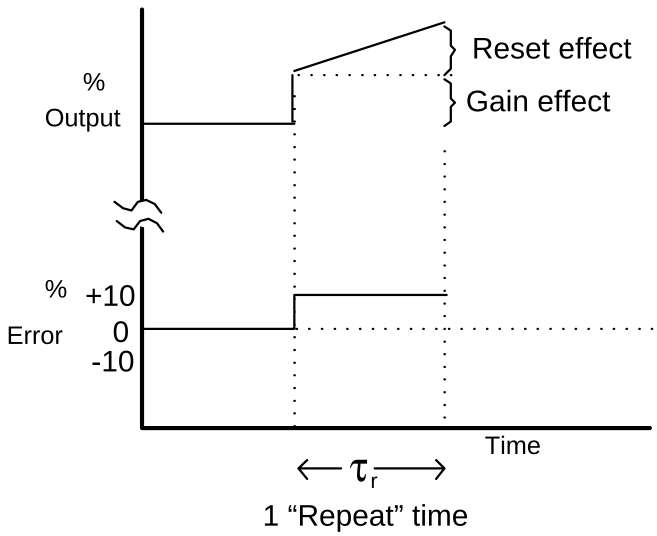



**Figure 12 Calculation of repeat time**

We can see two distinct effects of the change in the error. At the time the error was changed the output also changed. This is the "gain effect" and is equal to the product of the gain and the change in the error. The second effect (the "reset effect" is the ramp of the output due to the error. If we measure the time from when the error is changed to when the reset effect is equal to the gain effect we will have the "repeat time." Some control vendors measure reset by repeat time in minutes. Others measure reset by "**repeats per minute**." Repeats per minute is the inverse of minutes of repeat.

## DERIVATIVE

Derivative is the third and final element of PID control. Derivative responds to the rate of change of the process (or error). Derivative is normally applied to the process only). It has also been used as a part of a temperature transmitter ("Speed-Act™" - Taylor Instrument Companies) to overcome lag in transmitter measurement. Derivative is also known as Preact™ (Taylor) and Rate.

The derivative contribution can be expressed mathematically:

$$\mathbf{Out = g \times K_d \times \frac{dp}{dt}}$$

where **g** is gain,

**$K_d$** is the derivative setting in minutes, and

**p** is the process value

The open loop response of controller with proportional and derivative is shown graphically:

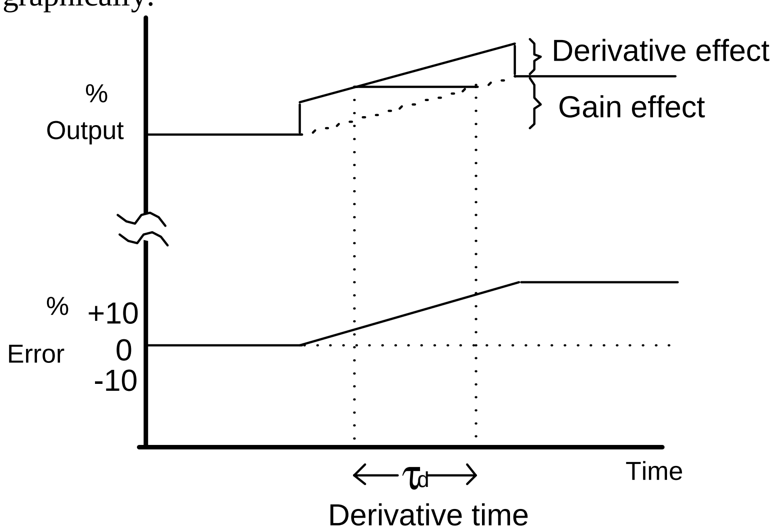



**Figure 13 Output vs. error of derivative over time**

This diagram compares the output of a controller with gain only (dashed line) with the output of a controller with gain and derivative (solid line). The solid line is higher than the dashed line for the time that the process is increasing due the addition of the rate of change to the gain effect. We can also look at the solid line as being "leading" the dashed line by some amount of time ($\tau_d$)

The amount of time that the derivative action advances the output is known as the "derivative time" (or Preact time or rate time) and is measured in minutes. All major vendors measure derivative the same: in minutes.

## COMPLETE PID RESPONSE

If we combine the three terms (Proportional gain, Integral, and Derivative) we obtain the complete PID equation.

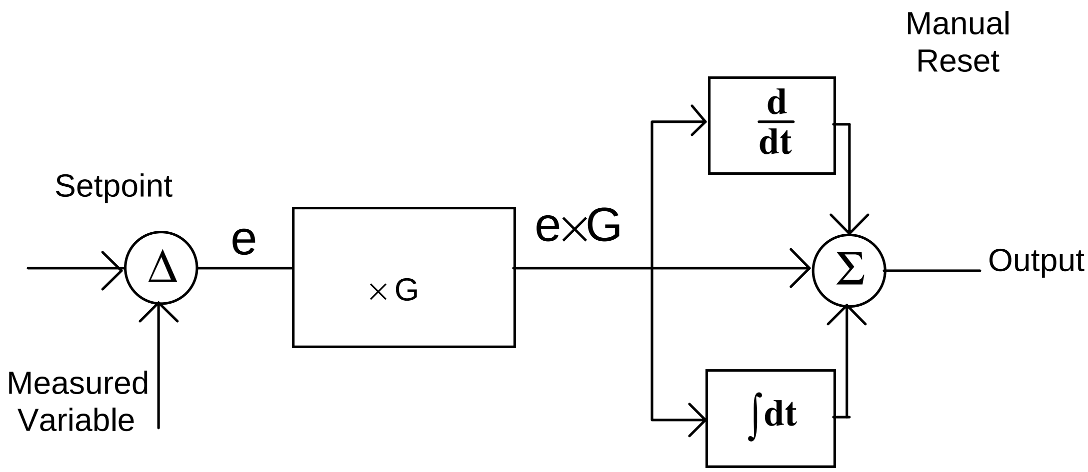



**Figure 14 Combined gain, integral, and derivative elements.**

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

Where

G = Gain
R = Reset (repeats per minute)
D = Derivative (minutes)

This is a general form of the PID algorithm and is close to, but not identical to, the forms actually implemented in industrial controllers. Modifications of this algorithm are described in the next section.

## RESPONSE COMBINATIONS

Most commercial controllers allow the user to specify Proportional only controllers, proportional-reset (PI) controllers, and PID controllers that have all three modes. The majority of loops employ PI controllers.

Most control systems also allow all other combinations of the responses: integral, integral-derivative, derivative, and proportional-derivative. When proportional response is not present the integral and derivative is calculated as if the gain were one.

## 3 IMPLEMENTATION DETAILS OF THE PID EQUATION

The description of the PID algorithm shown on the previous page is a "text book" form of the algorithm. The actual form of the algorithm used in most industrial controllers differs somewhat from the equation and diagram of shown on the previous page.

### SERIES VS. PARALLEL INTEGRAL AND DERIVATIVE

The form of the PID equation shown above, which is the way the PID is often represented in text books, differs from most industrial implementations in the basic structure. Most implementations place the derivative section in series with the integral or reset section.

We can modify the diagram shown above to reflect the series algorithm:

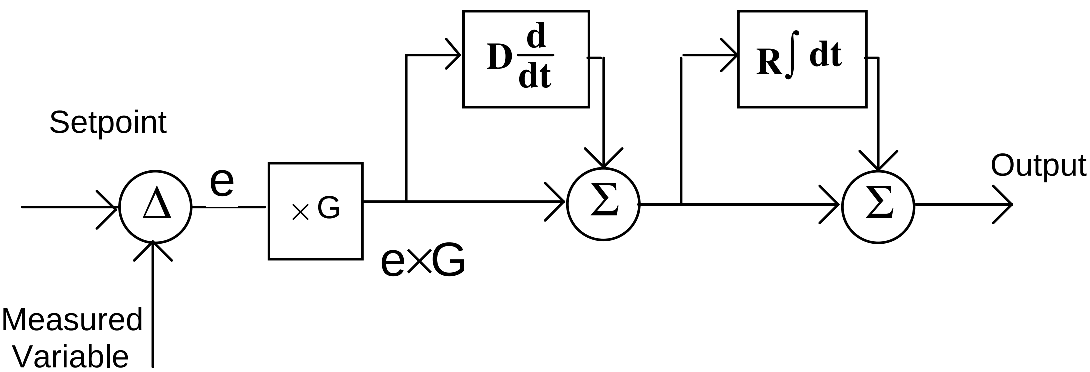



**Figure 15 A more correct representation of the complete PID response.**

The difference between this implementation and the parallel one is that the derivative has an effect on the integration. The equation becomes:

$$Out = (RD+1)G\left(e + \frac{R}{(RD+1)}\int edt + \frac{D}{(RD+1)}\frac{de}{dt}\right)$$

where R = the reset rate in repeats per minute,

D = the derivative in minutes,

and G = the gain.

The effect is to increase the gain by a factor of RD + 1, while reducing the reset rate and derivative time by the same factor.

Almost all analog controllers and most commercial digital control systems use the series form. Such tuning methods as the Ziegler-Nichols methods were developed using series form controllers.

Unless derivative is used there is no difference between the parallel (non-interactive) and series (interactive) forms.

## GAIN ON PROCESS RATHER THAN ERROR

The gain causes the output to change by an amount proportional to the change in the error. Because the error is affected by the set point, the gain will cause any change in the set point to change the output.

This can become a problem in situations where a high gain is used where the set point may be suddenly changed by the operator, particularly where the operator enters a new set point into a CRT. This will cause the set point, and therefore the output, to make a step change.

In order to avoid the sudden output change when the operator changes the set point of a loop, the gain is often applied only to the process. Set point changes affect the output due to the loop gain and due to the reset, but not due to the derivative.

## DERIVATIVE ON PROCESS RATHER THAN ERROR

The derivative acts on the output by an amount proportional to the rate of change of the error. Because the error is affected by the set point, the derivative action will be applied to the change in the set point.

This can become a problem in situations where the set point may be suddenly changed by the operator, particularly in situations where the operator enters a new set point into a CRT. This causes the set point to have a step change. Applying derivative to a step change, even a small step change, will result in a “spike” on the output.

In order to avoid the output spike when the operator changes the set point of a loop, the derivative is often applied only to the process. Set point changes affect the output due to the loop gain and due to the reset, but not due to the derivative.

Most industrial controllers offer the option of derivative on process or derivative on error.

## DERIVATIVE FILTER

The form of derivative implemented in controllers also includes filtering. The filter differs among the various manufactures. A typical filter comprises two first order filters that follow the derivative. The time constant of the filters depends upon the derivative time and the scan rate of the loop.

## EQUIVALENT CODE

The PID algorithm can be understood by reference to a small basic program that is the equivalent to the PID algorithm in its most common application. This is not the same implementation as used on any particular controller, but will provide the same response based on the following assumptions:

1. “Standard” integral algorithm is used.
2. Derivative is on process only.
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 calculatio
InputLast=Input
InputDF=InputDF+(InputD-InputDF)*DFilter/60          1st derivative filter
OutputTemp=(InputDF-SetPoint)*Gain+Feedback          Basic gain calculatio
IF OutputTemp >100  THEN OutputTemp= 100             Output Limits
IF OutputTemp <0  THEN OutputTemp= 0
Output=OutputTemp                                    The final output
Feedback=Feedback+(Feedback-Output)*ResetRate/60     Filter for reset
                                                     feedback
```

# 4 ADVANCED FEATURES OF THE PID ALGORITHM

## RESET WINDUP

One problem with the reset function is that it may ‘‘wind up’’. Because of the integration of the positive feedback loop, the output will continue to increase or decrease as long as there is an error (difference between set point and measurement) until the output reaches its upper or lower limit.

This normally is not a problem and is a normal feature of the loop. For example, a temperature control loop may require that the steam valve be held fully open until the measurement reaches the set point. At that point, the error will be cross zero and change signs, and the output will start decreasing, “throttling back” the steam valve.

Sometime, however, reset windup may cause problems. Actually, the problem is not usually the windup but the “wind down” that is then be required.

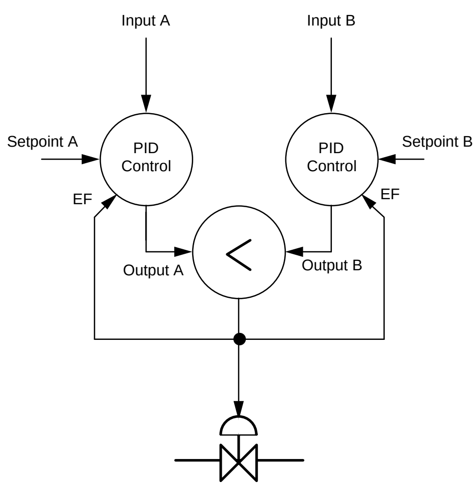



**Figure 16 Two PID controllers that share one valve.**

Suppose the output of a controller is broken by a selector, with the output of another controller taking control of the valve. In the diagram the lower of the two controller outputs is sent to the valve. Which ever controller has the lower output will control the valve. The other controller is, in effect, open loop. If its error would make its output increase, the reset term of the controller will cause the output to increase until it reaches its limit.

The problem is that when conditions change and the override controller no longer needs to hold the valve closed the primary controller’s output will be very far above the override signal. Before the primary controller can have any effect on the valve, it will have to “wind down” until its output equals the override signal.

## EXTERNAL FEEDBACK

The positive feedback loop that is used to provide integration can be brought out of the controller. Then it is known as external feedback:

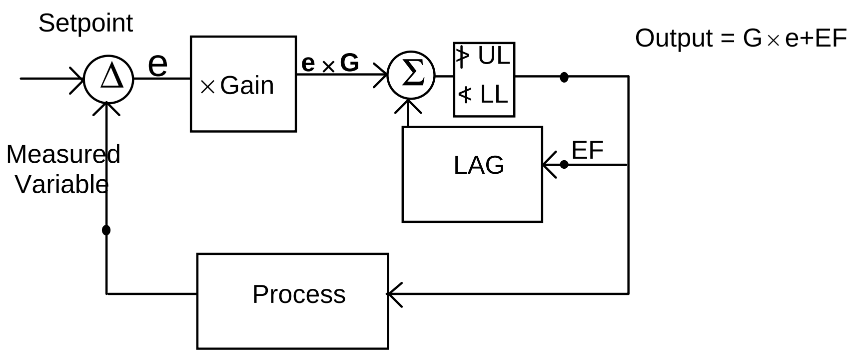



**Figure 17 A proportional-reset loop with the positive feedback loop used for integration.**

If there is a selector between the output of the controller and the valve (used for override control) the output of the selector is connected to the external feedback of the controller. This puts the selector in the positive feedback loop.

If the output of the controller is overridden by another signal, the overriding signal is brought into the external feedback. After the lag, the output of the controller is equal to the override signal plus the error times gain. Therefore, when the error is zero, the controller output is equal to the override signal. If the error becomes negative, the controller output is less than the override signal, so the controller regains control of the valve.

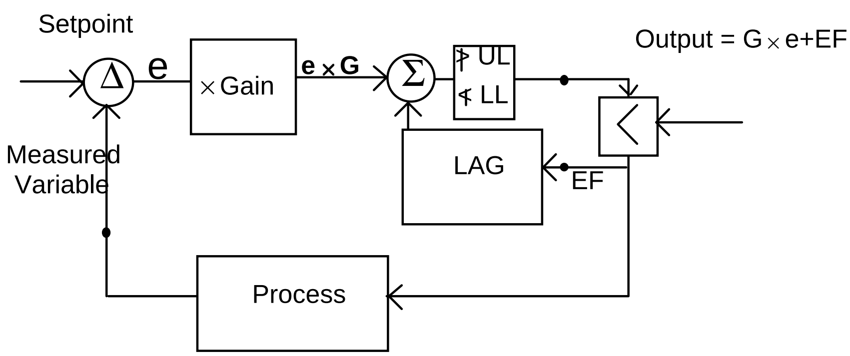



**Figure 18 The external feedback is taken from the output of the low selector.**

## SET POINT TRACKING

If a loop is in manual and the set point is different from the process value, when the loop is switched to auto the output will start moving, attempting to move the process to the set point, at a rate dependent upon the gain and reset rate. Take for example a typical flow loop, with a gain of 0.6 and a reset rate of 20. If difference between the set point and process is 50% at the time the loop is switched to automatic the output will ramp at a rate of 10%/second.

Often when a loop has been in manual for a period of time the value of the set point is meaningless. It may have been the correct value before a process upset or emergency shutdown caused the operator to place the loop into manual and change the process operation. On a return to automatic the previous set point value may have no meaning. However, to present a process upset the operator must change the set point to the current process measurement before switching the loop from manual to automatic.

Some industrial controllers offer a feature called “set point tracking” that causes the set point to track the process measurement when the loop is not in automatic control. With this feature when the operator switches from manual to automatic the set point is already equal to the process, eliminating any bump in the process.

# 5 PROCESS RESPONSES

Loops are tuned to match the response of the process. In this section we will discuss the responses of the process to the control system.

The dynamic and steady state response of the process signal to changes in the controller output. These responses are used to determine the gain, reset, and derivative of the loop.

## Self Regulation

For most processes, as a variable increases it will tend to reduce its rate of increase and eventually level off even without any change in the manipulated variable. This is referred to as *self regulation*.

Self regulation does not usually eliminate the need for a controller, because usually the value at which the variable will settle will be unacceptable. The control system will need to act to bring the controlled variable back to its set point.

An example of self regulation is a tank with flow in and out. The manipulated variable is the liquid flow into a tank. The controlled variable is the flow out of the tank. The load is the valve position of the discharge flow. With the valve position constant, the flow out of the tank is determined by the valve position and the level (actually the square of the level). As the level in the tank falls, the pressure (or liquid head) decreases, decreasing the flow rate. Eventually the discharge flow will decrease to the point that it equals the inlet flow, and the level will maintain a constant value. Likewise, if the flow into the tank increases, the level will begin to increase until the discharge flow equaled the inlet flow (unless the tank became full and overflowed first).

This also occurs in temperature loops. For example, if in a room with have an electric space heater with no thermostat. The room is too cool, so we turn on the space heater. As more heat enters the room, the room temperature increases. However, the flow of heat out through the walls is proportional to the difference between the inside and the outside temperatures. As the room temperature increases, that difference increases, and eventually equals the amount of heat produced by the space heater. As the temperature is increasing the rate of change decreases until the temperature levels off and a higher temperature.

Sometimes the self regulation is sufficient to eliminate any need for feedback control. However, more often the self regulation is not sufficient (the tank overflows or the room becomes too hot), therefore control is still needed. Because of self regulations, for at least some range of controller outputs there will be a corresponding process value.

The self regulation is responsible for the curve shown in the dynamic response of a controlled variable to a change in the measured variable.

## STEADY STATE RESPONSE

The steady state response of the process to the controller output is characterized primarily by process action, gain, and linearity.

### Process Action

Action describes the direction the process variable changes following a particular change in the controller output. A direct acting process increases when the final control element increases (typically, when the valve opens); a reverse acting process decreases when the final control element increases.

For example, if we manipulate the inlet valve on a tank to control level, an increase in the valve position will cause the level to rise. This is a direct acting process. On the other hand, if we manipulate the discharge valve to control the level, opening the valve will cause the level to fall. This is a reverse acting process.

### Process Gain

Next to action, process gain is the most important process characteristic. The *process gain* (not to be confused with *controller gain*) is the sensitivity of the process variable to changes in a controller output. Gain is expressed as the ratio of change in the process to the change in the controller output that caused the process change.

From the standpoint of the controller, gain includes the gain of the valve itself, of the process, and of the measurement transmitter. Therefore the size of the valve and the span of the transmitter will affect the process gain.

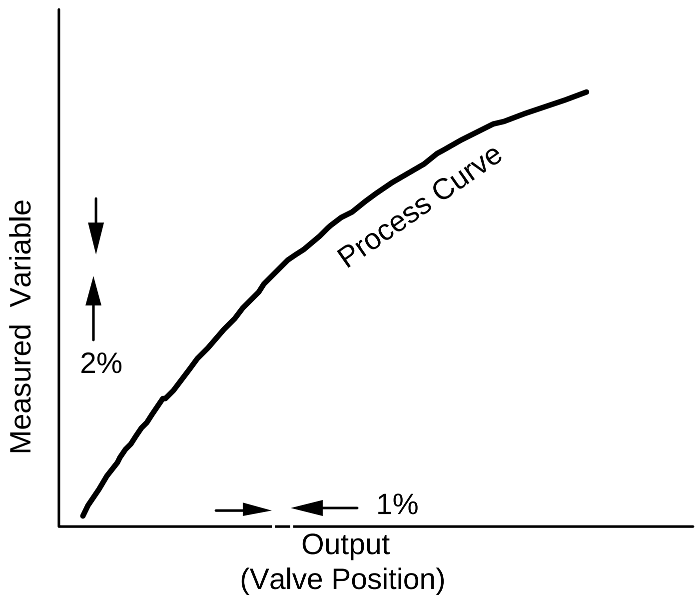



**Figure 19 The direct acting process with a gain of 2.**

In Figure 19 a 1% increase in the controller output causes the measured variable to increase by 2% . Therefore the process is direct acting and has a process gain of two.

## Process Linearity

The gain of the process often changes based on the value of the controller output. That is, with the output at one valve, a small change in the output will result in a larger change in the process measurement than the same output change at some other output value.

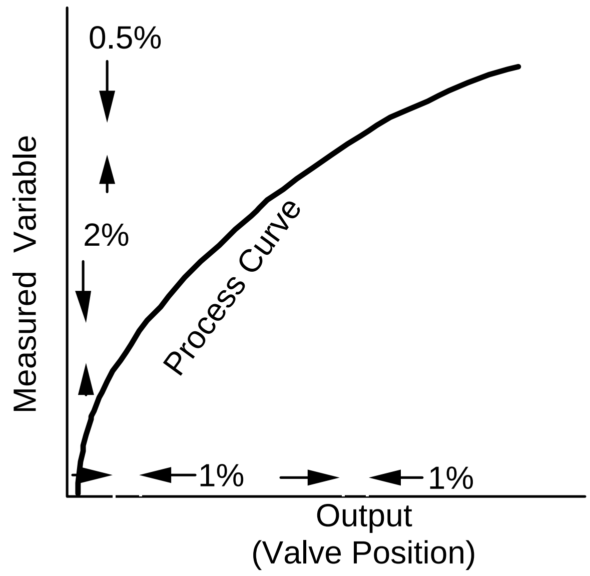



**Figure 20 A non-linear process.**

The process shown in Figure 20 is non linear. With controller output very low, a 1% increase in the output causes the measured variable to increase by 2%. When the output is very high, the same 1% output increase causes the process to increase by only 0.5%. The process gain decreases when the output increases.

From the standpoint of controller tuning, the process linearity includes the linearity of the process, the final control element, and the measurement. It also includes any control functions between the PID algorithm and the output to the valve.

## Valve Linearity

Valves may be linear or non-linear. A linear valve is one in which the flow through the valve is exactly proportional to the position of the valve (or the signal from the control system). Based on their linearity, valves may fall into three classes (illustrated in Figure 21): linear, equal percentage, and quick opening.

Linear valves have the same gain regardless of the valve position. That is, at any point a given increase in the valve position will cause the same increase in the flow as at any other point.

Equal percentage valves have a low gain when the valve is nearly closed, and a higher gain when the valve is nearly open.

Quick opening valves have a high gain when the valve is nearly closed and a lower gain when the valve is nearly open.

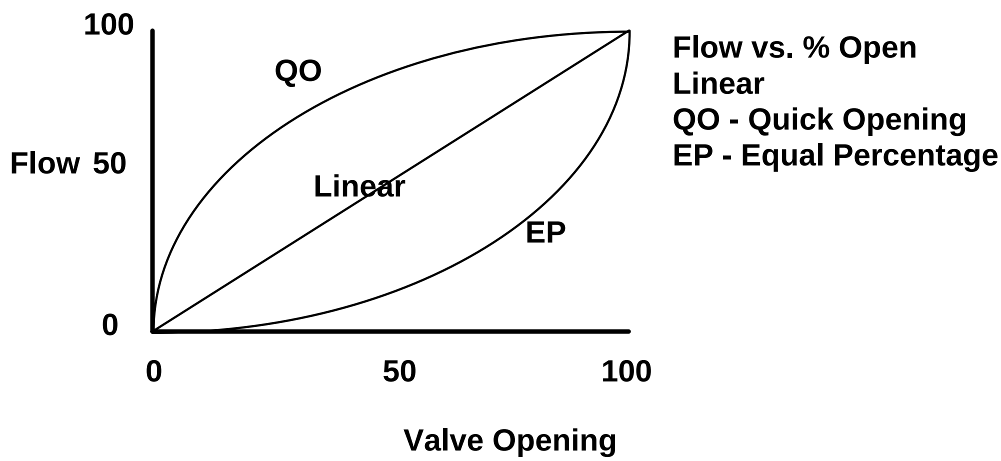



**Figure 21 Types of valve linearity.**

## Valve Linearity: Installed characteristics

Even a linear valve does not necessarily exhibit linear characteristics when actually installed in a process. The characteristics described in the previous section are based on a constant pressure difference across the flanges of the valve. However, the pressure difference is not necessarily constant. When the pressure is a function of valve position, the actual characteristics of the valve are changed.

Take for example the flow through a pipe and valve combination shown in Figure 22. Liquid flows from a pump with constant discharge pressure to the open air. There is a pressure drop through the valve that is proportional to the square of the flow. Assume that with a valve position of 10% the flow is 100 gpm. Also assume that with the particular size and length of the pipe 100 gpm causes a 10 psi pressure drop across each section of pipe. This leaves a net pressure from of 80 psi across the valve.

Assume now that we wish to double the flow rate to 200 gpm. By doubling the flow, we will increase the pipe pressure drop by a factor of four. With a pressure drop of 40 psi across each section of pipe, we will only have a valve differential pressure of only 20 psi. To make up for the loss of pressure, will have to increase the valve opening by a factor of four to make up for the pressure loss and by a factor of two to double the flow. Therefore, to double the flow rate we will have to open the valve from 10% to 80%. The so called linear valve now has the characteristics of a quick opening valve.

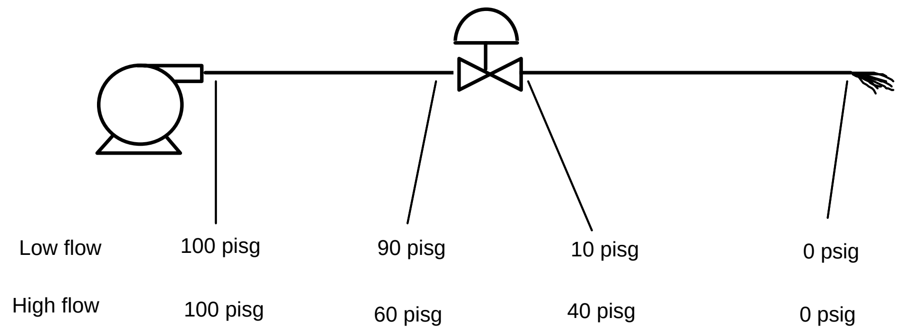



**Figure 22 A valve in a process. At *high* flow, the head loss through the pipe is more, leaving a smaller differential pressure across the valve.**

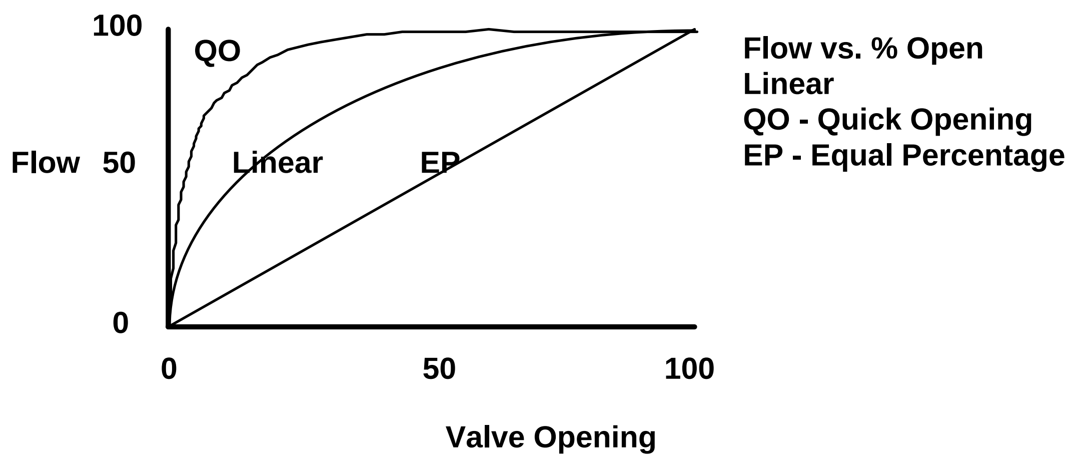



**Figure 23 Installed valve characteristics.**

## PROCESS DYNAMICS

The measured variable does not change instantly with the controller output changes. Instead, there is usually some delay or lag between the controller output change and the measured variable change. Understanding the dynamics of the loop is required in order to know how to properly control a process.

There are two basic types of dynamics: simple lag and dead time. Most processes are a combination of several individual lags, each of which can be classed as simple lag or dead time.

### Simple lag

The most common dynamic element is the simple lag. If a step change is made in the controller output , the process variable will change as shown in Figure 24.

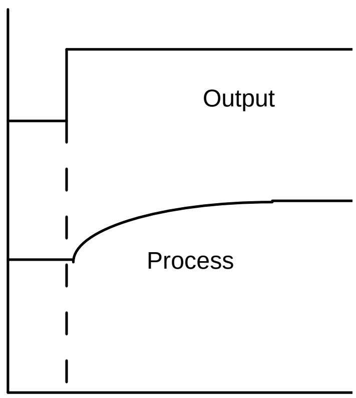



**Figure 24 Process with a single lag.**

An example of a process dominated by one loop is shown in Figure 25. The flow of the liquid out of the vessel is proportional to the level. If the inlet valve is opened, increasing the flow into the vessel, the level will rise. As the level rises, the flow output will rise, slowing the rate of increase in the level. Eventually, the level will be at the point where the flow out will be equal to the flow in.

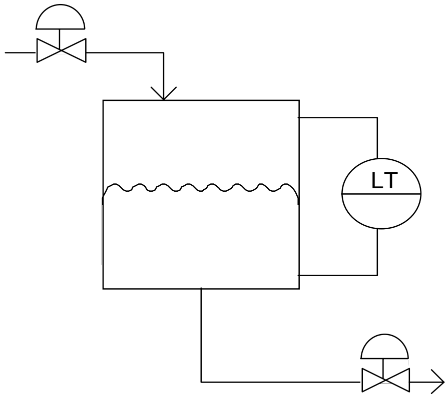



**Figure 25 Level is a typical one lag process.**

## Multiple Lags

Most processes have more than one lag, although some of the lags may be insignificant. Lags are not additive. A response of a multiple lag is illustrated in Figure 26.

The process measured variable begins to change very slowly, and the rate of change increases up to a point, known as the point of inflection, where the rate of change decreases as the measurement approaches its asymptote.

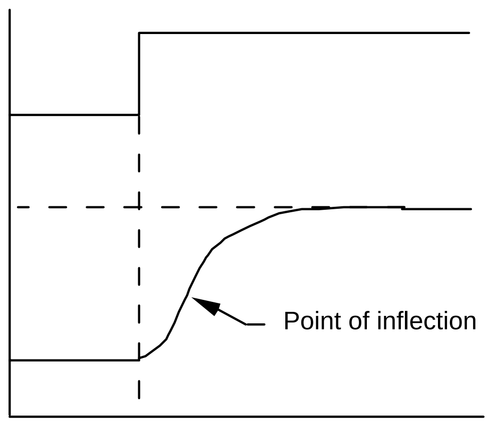



**Figure 26 Process with multiple lags.**

The first part of the curve, where the rate of change is increasing, is governed primarily by the second largest lag. The second part of the curve, beyond the point of inflection, is governed primarily by the largest lag.

## Dead time

*Dead Time* is the delay in the loop due to the time it takes material to flow from one point to another. For example, in the temperature control loop shown below, it takes some amount of time for the liquid to travel from the heat exchanger to the point where the temperature is measured. If the temperature at the exchanger outlet has been constant and then changes, there will be some period of time before any change can be observed by the temperature measurement element. Dead time is also called distance velocity lag and transportation lag.

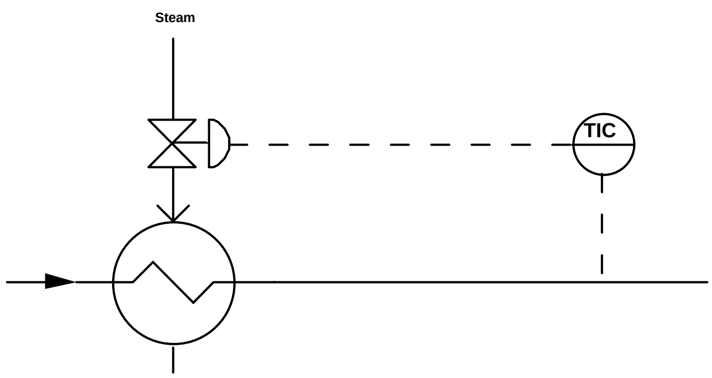
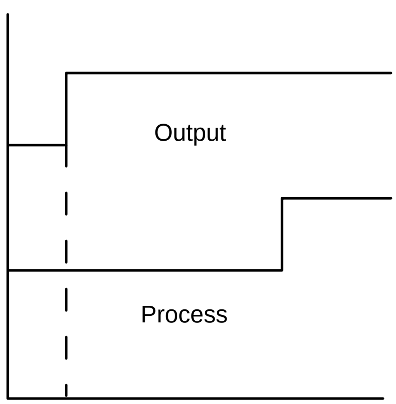



**Figure 27 Heat exchanger. The distance between the heat exchanger and the temperature measurement creates a dead time.**



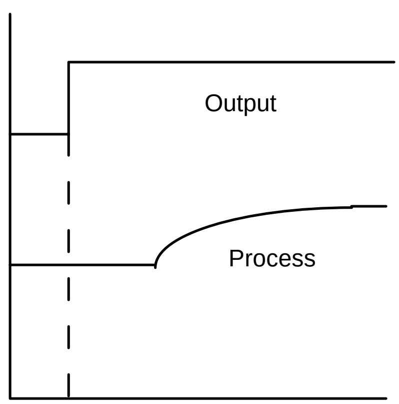




**Figure 28 Pure dead time.** **Figure 29 Dead time and lag.**

## Process Order

Often processes have been described as first order, second order, etc., based on the number of first order linear lags included in the process dynamics. It can be argued that all processes are of a higher order, with a minimum of three lags and a dead time. These lags, which are present in all processes, include the lag inherent in the sensing device, the primary lag of the process, and the time that the valve (or other final control element) takes to move. However, in many processes the smaller lags are so much smaller than the largest lag that their contributions to the process dynamics are negligible.

Dead time is also present in all processes. With pneumatic control, there is some dead time due to the transmission of the pressure signal from the process to the controller, and from the controller to the valve. This is eliminated by electronic controls (unless one considers the transmission of the electric signal, usually a few microseconds or less). With digital controls, there is an effective dead time equal to one half the loop scan rate [2]. In most cases, the loop will be scanned fast enough so that this dead time is insignificant. In some cases, such as liquid flow loops, this dead time is significant and affects the amount of gain that can be used.

Rather than consider a process to be first order, second order, etc., it may be better to consider all loops to be higher order to a degree. As an alternative to process order, we will characterize processes by the degree to which one first order lag dominates the other lags in the process (not considering any true dead time). *Dominant-lag* processes are those that consist of a dead time plus a single significant lag, with all other lags small compared to the major lag. *Multiple-lag* or *non-dominant-lag* processes are those in which the longest lag is not significantly longer than the next longest lag. One measure of the dominance of a single lag is the value of the process measurement at which the point of inflection (POI) occurs.(see Figure 30) In the most extreme case (only a single lag) the POI occurs at the initial process value. With about three equal, non-interacting lags the POI occurs at about 33% of the difference between the initial and the final process value.[7]

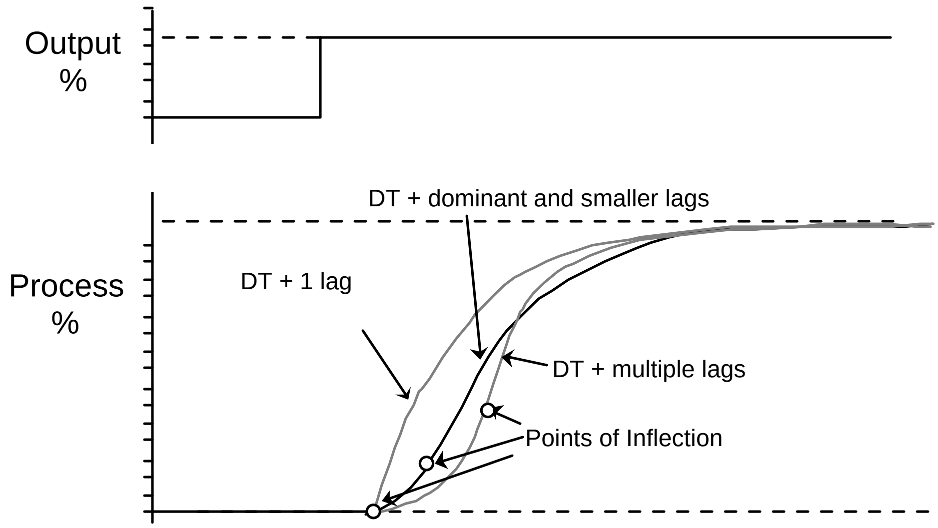



**Figure 30 The response of multiple lag processes to a step change in the controller's output.**

## MEASUREMENT OF PROCESS DYNAMICS

Process dynamics usually consist of several lags and dead time. The dynamics differ from one loop to another. The dynamics can be expressed by a detailed list of all of the lags and the dead time of the loop, or they can be approximated using a simpler model.

One such model is a dead time and a first order lag. Graphically, the process response of such a model is:

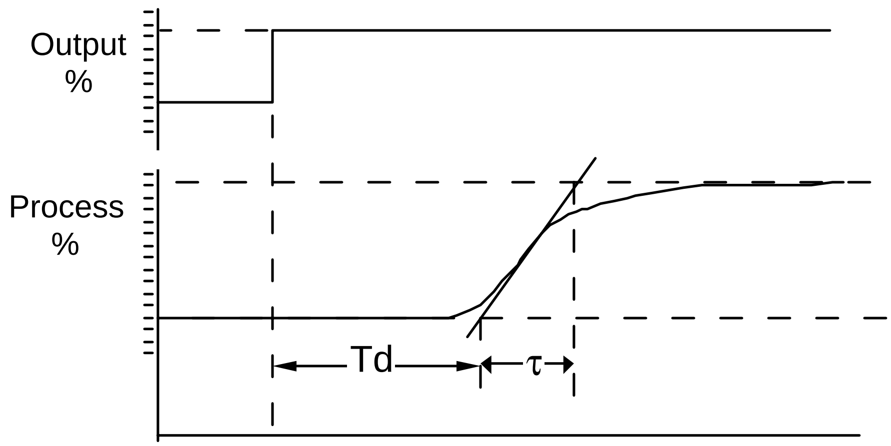



**Figure 31 Pseudo dead time and process time constant.**

The dynamics can be approximated by two numbers: $\tau$ is the process time constant. It is approximately equal to the largest lag in the process. Td is the pseudo dead time and approximates the sum of the dead time plus all lags other than the largest lag.

### First Order Plus Dead Time Approximation

Several tuning methods (such as the Ziegler-Nichols open loop method) are based on an approximation of the process as a combination of a single first order lag and a dead time, known as the First Order Plus Dead Time (FOPDT) model. These methods identify the process by making a step change in the controller output. The process trend is recorded and graphical or mathematical methods are used to determine the process gain, dead time, and first order lag.

*Process gain* is the ratio of the change in the process to the change in the controller output signal. It depends upon the range of the process measurement and includes effects of the final control element.

*Pseudo dead time* (Td) is the time between the controller output change and the point at which the tangent line crosses the original process value. The pseudo dead time is influenced by the dead time and all of the lags smaller than the longest lag in the process.

*Process time constant* ($\tau$) is the rate of change of the process measurement at the point at which the rate of change is the highest. The time constant is strongly influenced by the longest lag in a multiple lag process.

The ratio of the pseudo dead time to the process time constant is often referred to as an “uncontrollability” factor (Fc) that is an indication of the quality of control that can be expected. The gain (for a P, PI, and PID controller) at which oscillation will become unstable is inversely proportional to this factor. Smith, Murrill, and Moore, [5] proposed that the factor be modified by adding one half of the sample time to the dead time for digital controllers.

## LOADS AND DISTURBANCES

The process measurement is affected not only by the output of the control loop but by other factors called loads. These can include such factors as the weather, the position of other valves, and many other factors.

An example is shown in Figure 32. The level of the tank is controlled by manipulating the valve on the discharge line. However, the level is also affected by the flow into the tank. In fact, the flow into the tank has just as much effect on the level as the flow out of the tank. The inlet flow is therefore a load.

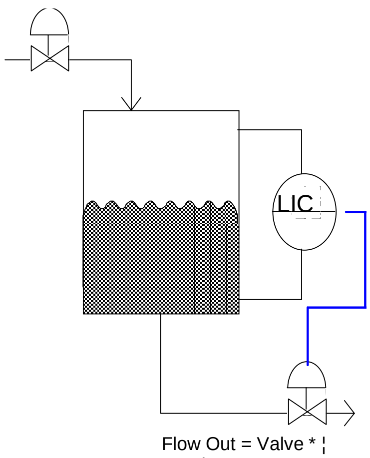



**Figure 32 Level control**

In a steam heater, the temperature is controlled by the valve on the steam line. However, the temperature is also affected by the temperature of the air around the vessel, although not nearly as much as by the steam valve. This temperature outside of the vessel is also a load.

The changes in a load are called *disturbances*. Almost all processes contain disturbances. They can be as major as the effect of the inlet flow on the vessel or as minor as the effect of weather on the temperature loop.

# 6 LOOP TUNING

Once a loop is configured and started up, in order for it to work correctly someone has to put the correct gain, reset, and derivative values into the PID control algorithm.

## TUNING CRITERIA OR “HOW DO WE KNOW WHEN ITS TUNED”

One of the most important, and most ignored, facets of loop tuning is the determination of the proper tuning of a loop.

### The extremes: instability and no response

The loop performance must fall between two extremes. First, the loop must respond to a change in set point and to disturbances. That is, an error, or difference in the process and the set point, must eventually result in the manipulation of the output so that the error is eliminated. If the gain, reset, and derivative of the loop are turned to zero there will be no response.

The other extreme is instability. An unstable loop will oscillate without bound. A set point change will cause the loop to start oscillating, and the oscillations will continue. At worst, the oscillations will grow (or diverge).

Proper tuning of a loop will allow the loop to respond to set point changes and disturbances without causing instability.

### Informal methods

There are several rules of thumb for determining how the quality of the tuning of a control loop.

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

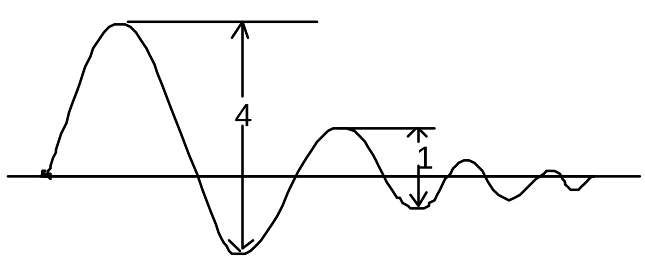



**Figure 33 Quarter wave decay.**

Traditionally, quarter wave decay has been considered to be the optimum decay ratio. This criteria is used by the Ziegler Nichols tuning method, among others. There is no specific combination of tuning parameters that will provide quarter wave decay.

Quarter wave decay is not necessarily the best tuning for either disturbance rejection or set point response. However, it is a good compromise between instability and lack of response.

### *Minimum overshoot.*

For some loops the objective of the tuning is to minimize the overshoot following a set point change.

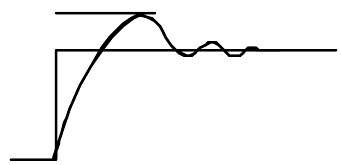


**Figure 34 Overshoot following a set point change.**

### *Maximum disturbance rejection.*

For other loops the primary concern is the reduction of the effect of disturbances.

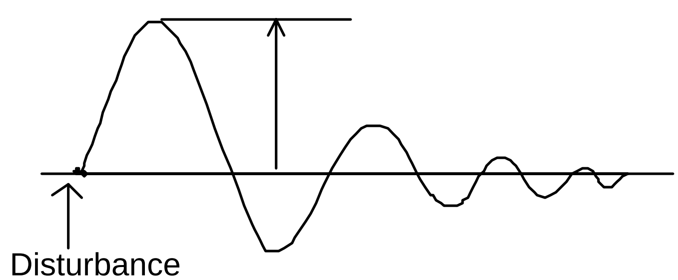



**Figure 35 Disturbance Rejection.**

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

## MATHEMATICAL CRITERIA—MINIMIZATION OF INDEX

**Figure 36 Integration of error.**

There are several criteria for evaluating tuning that are based on integrating the error following a disturbance or set point change. These methods are not used to test control loops in actual plant operation because the usual process noise and random disturbances will affect the outcome. There are used in control theory education and research using simulated processes. The indices provide a good method of comparing different methods of controller tuning and different control algorithm.

IAE - Integral of absolute value of error

$$\int |e|\ dt$$

ISE - Integral of error squared

$$\int e^2\ dt$$

ITAE - Integral of time times absolute value of error

$$\int t\,|e|\ dt$$

ITSE - Integral of time times error squared:

$$\int t\,e^2\ dt$$

## ZIEGLER NICHOLS TUNING METHODS

In 1942 J. G. Ziegler and N. B. Nichols, both of the Taylor Instrument Companies (Rochester, NY) published a paper [1] that described two methods of controller tuning that allowed the user to test the process to determine the dynamics of the process. Both methods assume that the process can be represented by the model (described above) comprising the process gain, a ‘‘pseudo dead time’’, and a lag. The methods provide a test to determine process gain and dynamics and equations to calculate the correct tuning.

The Ziegler Nichols methods provide quarter wave decay tuning for most types of process loops. This tuning does not necessarily provide the best ISE or IAE tuning but does provide stable tuning that is a reasonable compromise among the various objectives. If the process does actually consist of a true dead time plus a single first order lag, the Z-N methods will provide quarter wave decay. If the process has no true dead time but has more than two lags (resulting in a “pseudo dead time”) the Z-N methods will usually provide stable tuning but the tuning will require on-line modification to achieve quarter wave decay.

Because of their simplicity and because they provides adequate tuning for most loops, the Ziegler Nichols methods are still widely used.

### Determining the First Order Plus Dead Time model

The Ziegler Nichols method, as well as several other methods for controller tuning, rely on a model of the process that comprises one first order lag plus dead time. The FOPDT model parameters can be determined from the actual process using a simple process reaction test. The output from the controller is increased (or decreased) in a step change, and the reaction of the process is recorded. The process gain is the ratio of the change in the process (in percent) to the change that had been made in the controller output.

Several methods have been proposed to calculate the pseudo dead time, and time constant from the reaction curve.

Ziegler and Nichols proposed a graphical method (Figure 37) using a tangent line drawn through the steepest part of the curve (the point of inflection). The line continues below the original process value. The time between the output change and the point at which the tangent line crosses the original process line is called the lag (the term Pseudo Dead Time will be used in this paper). The slope of the line is then calculated. The original Ziegler-Nichols formulas used the slope or the rate of change rather than the time.

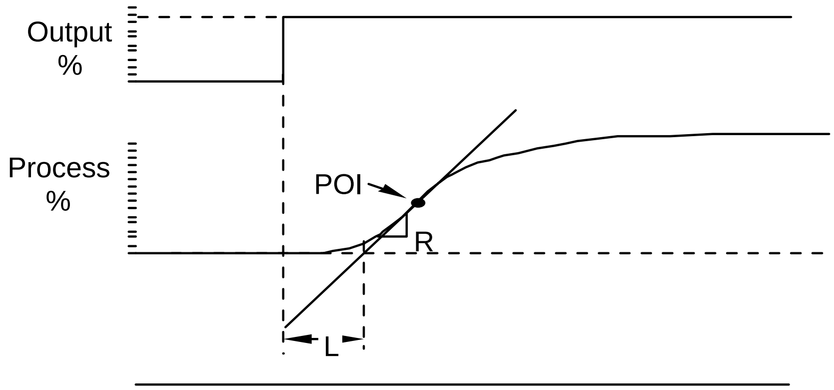



**Figure 37 The Ziegler-NIchols Reaction Rate method.**

Another graphical method (Figure 38), which is the mathematical equivalent of the original Ziegler-Nichols method, is known as the “tangent method”. In this method the same tangent line is drawn, but the process time constant is the time between the interception of the tangent and the original process value line and the eventual process value line. The formulas that are commonly provided for the Ziegler-Nichols open loop method use the process time constant.

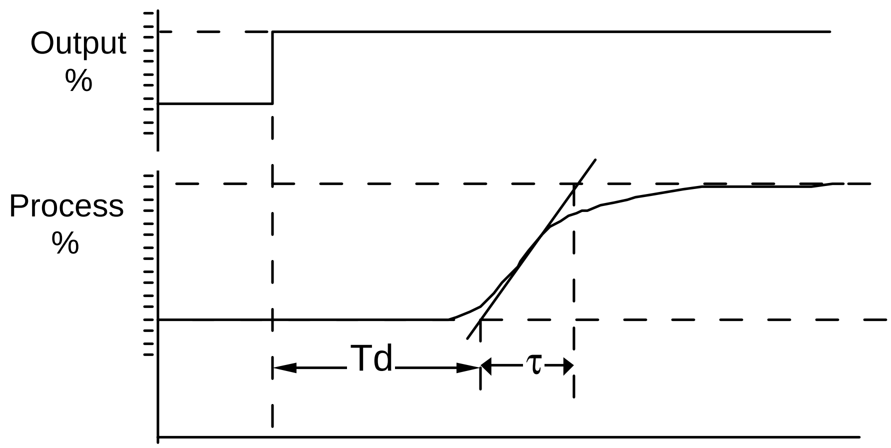



**Figure 38 Tangent method.**

These two methods will provide identical results when applied perfectly, that is, no error in the drawing of the line and no noise in the process signal.

Two additional methods are the mathematical equivalent of the previous two only if the process dynamics really did comprise only a single order lag and a dead time. When the process differs from this model, the following two methods will provide different results that may actually provide better tuning.

The first of these is sometimes known at the “tangent-and-point method” (T+P) Figure 39 [6]. In this method the same tangent line is drawn as before and used to calculate the pseudo dead time as before. However, a point equal to 63.2% of the value between the original and the ultimate process measurement is made on the tangent line. The time between the end of the pseudo dead time and the time at which the tangent line goes through the 63.2% point is the process time constant. This method will give the same results as the first two when the process is truly a dead time plus first order lag. As the values of the smaller lags increase, the tangent and point method gives a smaller time constant than the two graphical methods.

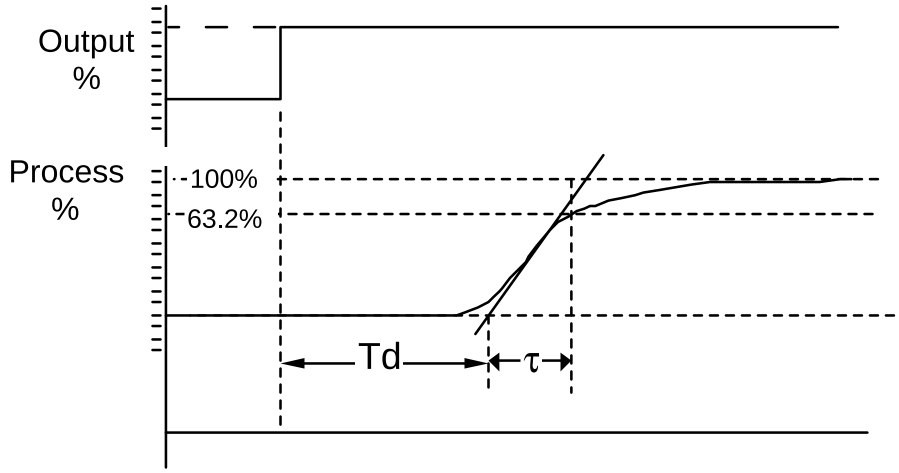



**Figure 39 The tangent plus one point method.**

Another variation is the “two point method”, proposed by C. Smith [3, p141], illustrated in Figure 40. This method does not require the drawing of a tangent line but measures the times at which the process changes by 28.3% ($t_1$) and 63.2% ($t_2$)of the total process change. Then two formulas are used to calculate the pseudo dead time and the process time constant.

*process time constant* $\tau = 1.5(t_1 - t_2)$

*pseudo dead time* $T_d = t_1 - \tau$

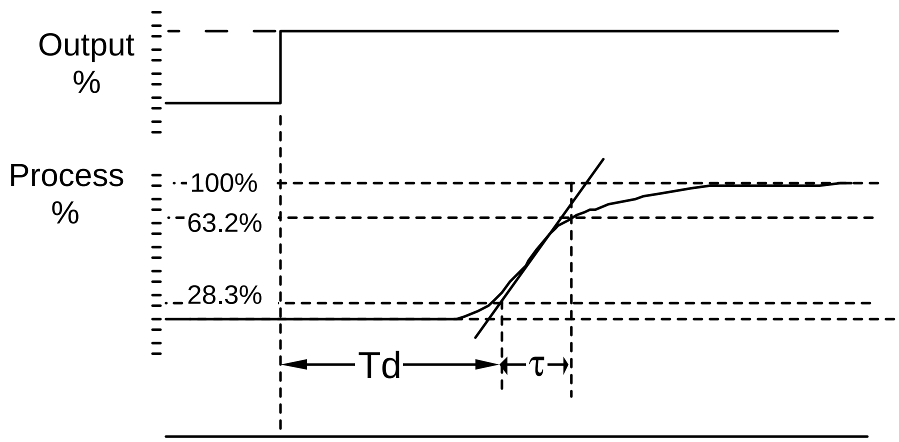



**Figure 40 The two point method.**

One advantage of the two point method is that it does not require drawing the tangent line. This improves the accuracy, particularly for processes with noise. This method will provide the same pseudo dead time and time constant when the process dynamics are dominated by a true dead time and single lag. Like the tangent and point method, the two point method provides different values for the pseudo dead time and the process time constant when the process dynamics comprise multiple lags. When used with the multiple lag processes, the one point and two point methods also provide better tuning. In table 3, tests within each set all result in the same pseudo dead time and time constant using the tangent line method. However, using the two point method the pseudo dead time and time constant are different. For multiple lag processes the two point method results in a longer dead time and shorter time constant than tangent methods. This provides more conservative tuning using any of the FOPDT tuning methods.

## Ziegler Nichols open loop tuning method

The first method is the open loop method, also known as the ‘‘reaction curve’’ method. This method calculates the actual values of the of the assumed process model (the gain, pseudo dead time, and lag).

For this method to work the process must be “lined out”—that is, not changing. With the controller in manual, the output is changed by a small amount. The process is then monitored.

The following values are calculated using one of the methods described above:

$K_p$ Process Gain

$\tau$ Process Time Constant

$T_d$ Pseudo Dead Time

The gain, reset, and Preact are calculated using:

| | Gain | Reset | Derivative |
|---|---|---|---|
| P | $\frac{\tau}{T_d\,Kp}$ | | |
| PI | $0.9\,\frac{\tau}{T_d\,Kp}$ | $\frac{.3}{T_d}$ | |
| PID | $1.2\,\frac{\tau}{T_d\,Kp}$ | $\frac{.5}{T_d}$ | .5Td |

## Ziegler Nichols closed loop tuning method

The closed loop (or ultimate gain method) determines the gain that will cause the loop to oscillate at a constant amplitude. Most loops will oscillate if the gain is increased sufficiently.

The following steps are used:

1. Place controller into automatic with low gain, no reset or derivative.
2. Gradually increase gain, making small changes in the set point, until oscillations start.
3. Adjust gain to make the oscillations continue with a constant amplitude.
4. Note the gain (Ultimate Gain, $G_{u,}$) and Period (Ultimate Period, $P_{u.}$)

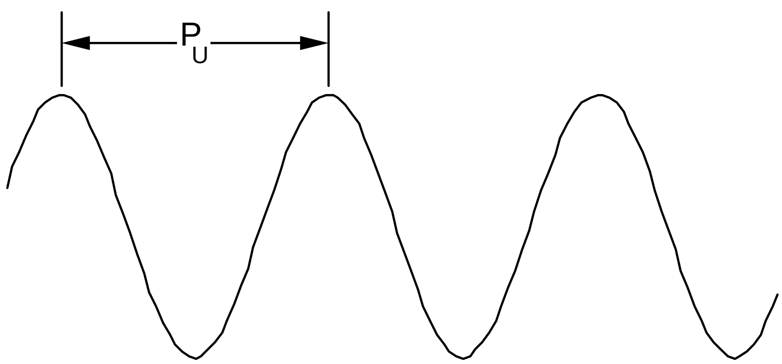



**Figure 41 Constant amplitude oscillation.**

The gain, reset, and Preact are calculated using:

| | Gain | Reset | Preact |
|---|---|---|---|
| P | $0.5\ G_U$ | — | — |
| PI | $0.45\ G_U$ | $\frac{1.2}{P_U}$ | — |
| PID | $0.6\ G_U$ | $\frac{2}{P_U}$ | $\frac{P_U}{8}$ |

## COHEN-COON

The Cohen-Coon method is similar to the Ziegler-Nichols reaction rate method in that it makes use of the FOPDT model to develop the tuning parameters. The parameters (shown below) are more complex, involving more arithmetic operations. As can be seen from the tables, the Cohen-Coon method will result in a slightly higher gain than the Ziegler-Nichols method. For most loops it will provide tuning closer to quarter wave decay and with a lower ISE index than the Ziegler-Nichols open loop method [4].

| | Gain x Kp | Reset Rate/$\tau$ | Derivative/$\tau$ |
|---|---|---|---|
| P | $\frac{\tau}{T_d} + .333$ | | |
| PI | $.9\frac{\tau}{T_d} + .082$ | $\frac{3.33(\tau/Td)[1+(\tau/Td)/11]}{1+2.2(\tau/Td)}$ | |
| PID | $1.35\frac{\tau}{T_d} + .27$ | $\frac{2.5(\tau/Td)[1+(\tau/Td)/5]}{1+.6(\tau/Td)}$ | $\frac{0.37(\tau/Td)}{1+0.2(\tau/Td)}$ |

## LOPEZ IAE-ISE

Integral Absolute Error and Integral Squared Error are two methods of judging the tuning of a control loop. (see below). A method of selecting tuning coefficients to minimize the IAE or ISE criteria for disturbances was developed by Lopez, et. al. [5].

This method uses the FOPDT model parameters and a set of equations (Table 9) to calculate the tuning parameters. Tests show that the parameters provide results close to the minimum IAE or ISE, particularly when the actual process dynamics are similar to the FOPDT model (the process contains a true dead time and one lag). When the process has multiple lags the equations do not provide the best possible tuning, but they still provide better tuning (lower IAE and ISE indices) than the other methods.

| | Gain x Kp | Reset Rate | Derivative |
|---|---|---|---|
| P | $1.411 Td/\tau^{-.917}$ | | |
| PI | $1.305 Td/\tau^{-.959}$ | $(\tau/.492)(Td/\tau)^{.739}$ | |
| PID | $1.495 Td/\tau^{-.945}$ | $(\tau/1.101)(Td/\tau)^{.771}$ | $.56\tau(Td/\tau)^{1.006}$ |

## CONTROLLABILITY OF PROCESSES

The gain at which a loop will oscillate depends upon the dynamics of the loop. In general, a loop that has no dynamic elements other than one first order lag will not oscillate at any gain. A loop with dead time or with multiple lags will oscillate at some gain.

If we refer to the model used on the Ziegler Nichols open loop test, the gain at which a loop will exhibit undamped, sustained oscillations (the ultimate gain in the Ziegler Nichols closed loop test) will depend upon the ratio of the process time constant ($\tau$) to the pseudo dead time (Td).

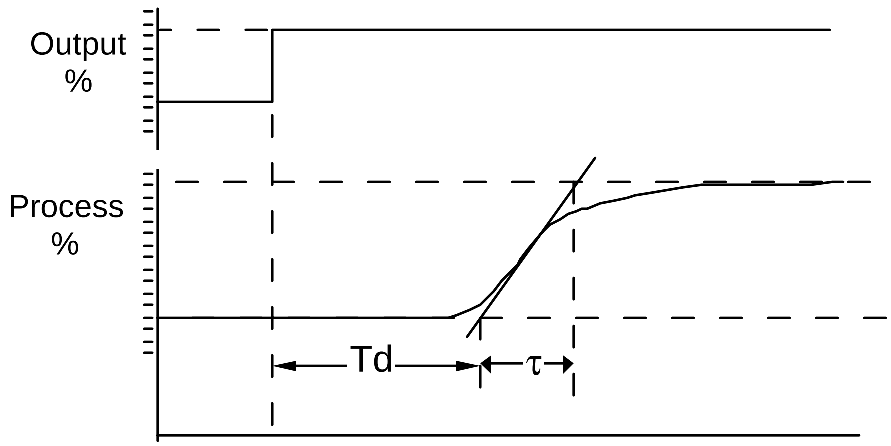



**Figure 42 Pseudo dead time and lag.**

The importance of this fact goes beyond finding the best tuning parameters. There is an advantage to a loop that can have a higher gain. If a loop can have a higher gain it will have greater rejection of disturbances and will respond more rapidly to setpoint changes. Therefore, it is advantageous to be able to increase the gain if doing so will not cause the loop to become unstable.

Remember that the time constant is proportional to the largest lag in the system. The pseudo dead time is based on the dead time and all other lags. The allowable gain (and the gain required for quarter wave decay) can be increased by increasing the ratio $\frac{t}{Td}$ . An increase in either dead time or in any lag other than the longest lag will decrease the ratio and therefor decrease the allowable gain. The loop scan period has the effect of process dead time. Increasing the scan period will decrease the ratio and the allowable gain. Also, adding any lag smaller than the longest lag, (for example, adding a large well to a thermocouple or a filter to a noisy loop) will decrease the allowable gain.

## FLOW LOOPS

Flow loops are too fast to use the standard methods of analysis and tuning. The speed of the loop compared to the update rate of the display prevents collection of the data for either the open loop or closed loop methods. However, there are some rules of thumb regarding the tuning of flow loops.

Typically, the tuning of a flow loop using digital control is:

Gain = 0.5 to 0.7

Reset = 15 to 20 repeats/min.

No Derivative

### Analog vs. Digital control:

The guidelines above are based on digital controller such as that used in Distributed Control Systems. Some flow loops using analog controllers are tuned with higher gain. However, the same loop may go unstable if a digital controller is used.

While normally a digital control system will provide response very similar to that of an analog control system, the performance can be quite different with fast loops such as flow. With an analog controller, the flow loop has a time constant ($\tau$) of a few seconds and pseudo dead time much smaller than one second. However, 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 $\tau$ to cause instability when the gain is higher than one.

# 7 BIBLIOGRAPHY


1. J. G. Ziegler, N. B. Nichols, Optimum Settings for Automatic Controllers, ASME Transactions, Vol. 64, 1942, pp. 759-768.

2. A. B. Corripio, *Tuning of Industrial Control Systems*, Instrument Society of America, 1990.

3. C. L. Smith, *Digital Computer Process Control,* International Textbook Company, 1972.

4. J. A. Miller, A. M. Lopez, C.L. Smith, P. W. Murrill, A *Comparison of Controller Tuning Techniques*, Control Engineering, Dec. 1967, pp. 72-75.

5. A. M. Lopez, P. W. Murrill, C. L. Smith, “Controller Tuning Relationships Based on Integral Performance Criteria,” *Instrumentation Technology*, V. 14 [Nov. 1967], p. 57.

6. H. Wade, Course notes for *Principles of Applied Automatic Control* (an ISA short course), Instrument Society of America, 1992.

7. J. Shaw, *Analysis of Traditional PID Tuning Methods*, presented at the Instrument Society of America conference, Chicago, IL, September, 1993.