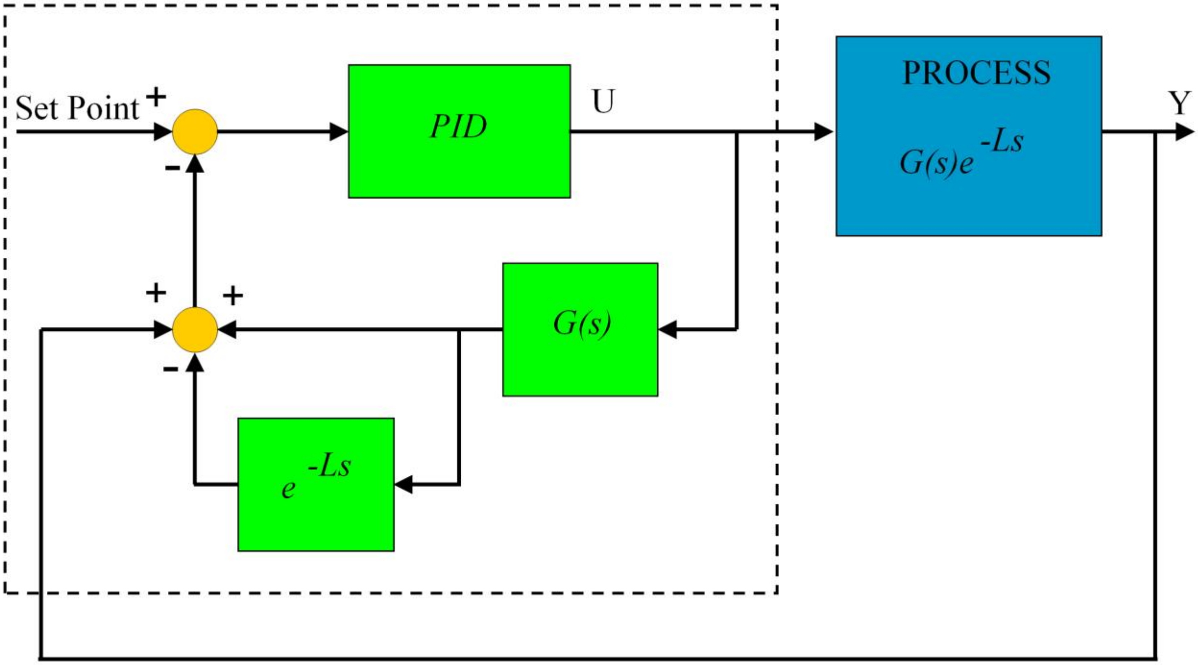


PROCESS CONTROL
TUTORIAL

A n overview of engineering mathematics,
process dynamics and PID control

Jon Monsen, Ph.D., P.E.

PROCESS CONTROL TUTORIAL

An overview of engineering mathematics, process dynamics and PID control

Jon Monsen, Ph.D., P.E.

Process Control Tutorial

An overview of engineering mathematics, process dynamics and PID Control

By Jon Monsen, Ph.D., P.E.

Dedication

This book is dedicated to my wife, Paulette, for her lifetime of support and to my many friends at Jamesbury, Neles and Valin for the help and encouragement that I received over the years.

Special thanks to Richelle Fischer of Valin Corporation for her help in reviewing my manuscript.

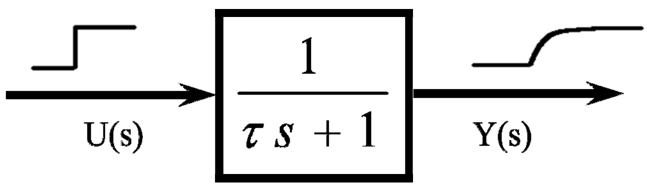
Contents

Chapter 1

Review of Mathematics for Process Dynamics

Topics

- •Functions
- •Rate of Change
- •Exponential Function
- •Exponential Decay
- •Time Constant
- •Laplace Transform
- •Transfer Functions



The purpose of this chapter that presents an overview of the mathematics that apply to process dynamics is to familiarize you with, or refresh your memory of, some of the terms that are used in the chapter on "Dynamics of Industrial Processes" and that are used by control system engineers.

The final goal of this chapter is to instill in the reader an understanding of "transfer functions," the topic at the bottom the list of topics in the figure above. To help the reader appreciate the importance of each of the topics, the discussion below starts with transfer functions and works up the list explaining how each topic depends on an understanding of the previous topic.

When we talk about how various process elements (heaters, blending tanks, etc.) respond to various types of inputs we will use a shorthand notation that involves block diagrams and transfer functions.

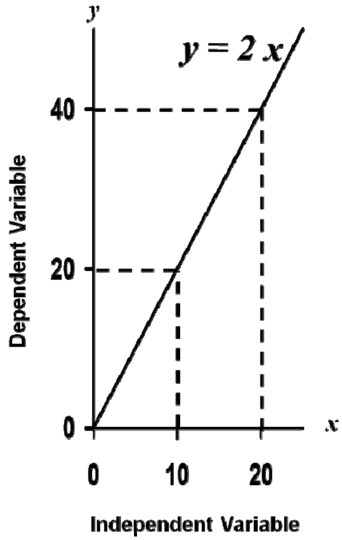
Because the transfer functions are written in terms of the Laplace transform of a process element's response to time varying inputs, we need to understand what the Laplace transform is. The block and transfer function in the figure above are particularly important. Write down the expression in the block and the words “first order lag” as we will keep coming back to them.

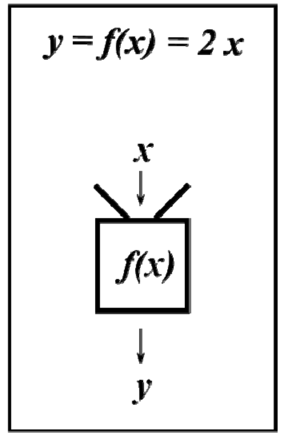
The time constant is central in describing the dynamic behavior of most process elements. (The tau, τ, in the transfer function in the figure on the previous page represents the time constant.) Since the time constant is a characteristic of the naturally occurring exponential decay we need to discuss that topic.

To understand the exponential decay we need to define the Exponential function and the number “e” which describes things whose rate of change depends on their size at that time. Since the Exponential function is based on the Derivative and rate of change, we need to define these subjects.

Since all the mathematical concepts we will talk about are functions of various types, we need to discuss what a function is.

Functions





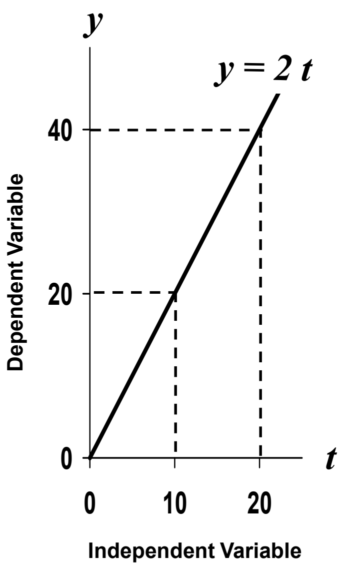
The first topic is FUNCTIONS.

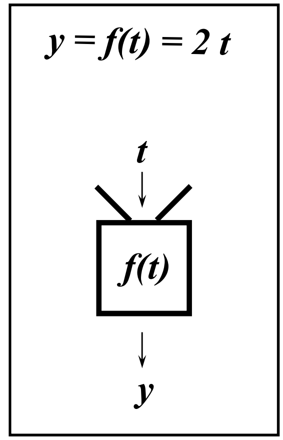
A function describes how one variable (in this case, y) depends on another variable, in this case x. In this example, the value of y depends on the value of x, so we say y is the dependent variable and x is the independent variable. We also say that "y is a function of x" and write it as shown in the box

A function can be thought of as a machine (depicted on the right side of the figure) into which you put values of the independent variable and get out the corresponding values of the dependent variable.

In this example the functional relationship between x and y is that y is 2 times x. When you put an x of 10 into the function (or machine), a y of 20 comes out and when you put in an x of 20 a y of 40 comes out.

Functions of TIME





Since I am leading up to a discussion of PROCESS DYNAMICS where we will talk about how the inputs and outputs of processes change with time, we will be talking about FUNCTIONS OF TIME.

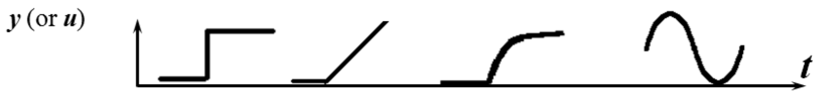
The figure on this page is the same as the one on the previous page, except that y is a function of time, that is the value of y depends on time and everywhere there was an x on the previous page, there is a t (for time) on this page.

Functional Notation

$y = f(t) = 2t$	"y is a function of t (time) and is equal to 2t" or "y equals *f* of t which equals 2t"

$y(t) = 2t$	"y is a function of t (time) and is equal to 2t" or "y of t equals 2t"

$y(t)$ $u(t)$	"y of t (or u of t)" "y (or u) is a function of time"

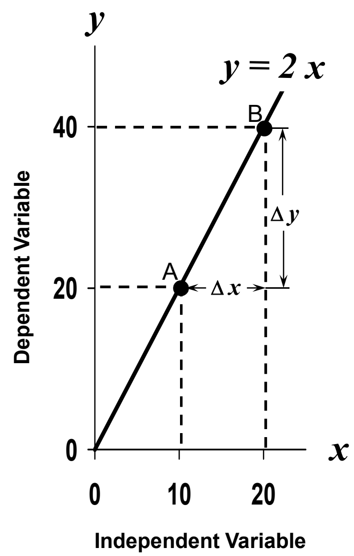


Besides seeing y written as it was on the previous page (top box on this page) it is also sometimes written like in the second box and we would read it as written in the second box.

Later I will be referring to time functions whose shape is to be determined, or could be assigned different equations and shapes. Then they will be written as shown in the third box and I will simply refer to it as y or u of t, meaning it is a yet to be determined function of time.

It is fairly common to refer to the inputs to process elements as u and the outputs from process elements as y.

Rate of Change



Rate of change of ***y*** with respect to ***x***

= **Slope of Graph**

$$= \frac{\Delta y}{\Delta x}$$

$$= \frac{40 - 20}{20 - 10}$$

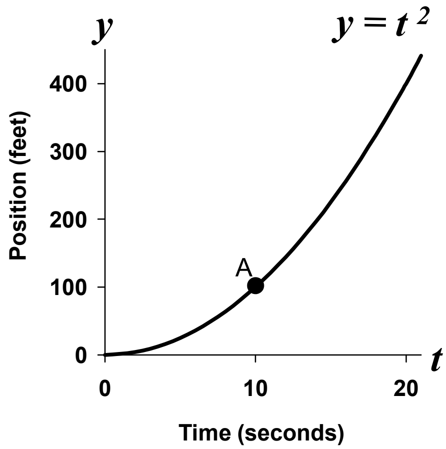
$$= \frac{20}{10}$$

$$= 2$$

The rate of change of the dependent variable with respect to changes in the independent variable is of considerable interest in the study of dynamic systems. In the case of a linear function (one whose graph is a straight line) the rate of change is constant and is equal to the slope of the graph. The rate of change can be determined very simply by choosing an arbitrary change in y (which we call “delta y) and dividing it by the corresponding change in x (which we call “delta x.”

In this example I have arbitrarily chosen to evaluate the rate of change of y with respect to x using the two points A and B. Between points A and B the change in y is 20 and the corresponding change in x is 10. 20 divided by 10 is 2, so the rate of change of y with respect to x is 2.

Rate of Change



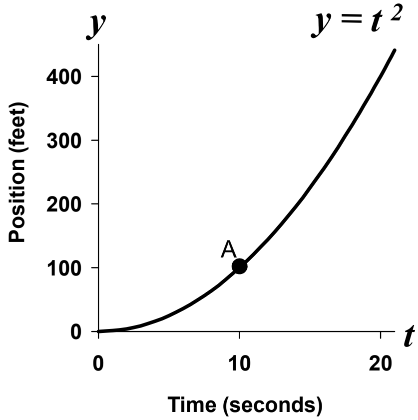
Rate of change of y with respect to t

= **Slope of Graph**

With a non-linear graph, determining the rate of change of the dependent variable (in this example, y) with respect to the independent variable (in this example, time or t) is not so simple, because the rate of change is constantly changing.

This graph describes how far some object has traveled from the starting point (0) at various times. That is after 10 seconds, it has traveled 100 feet, and after 20 seconds it has traveled 400 feet. At any time, the distance it has traveled is equal to the time squared.

The Derivative



Rate of change of y with respect to t

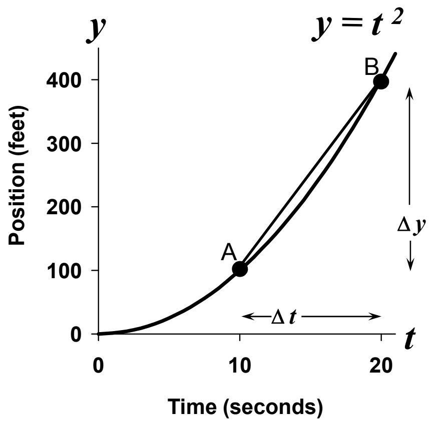
= **Slope of Graph**

NOTE: THE ONLY CHANGE FROM THE PREVIOUS PAGE IS THAT THE TITLE CHANGES FROM "RATE OF CHANGE" TO "THE DERIVATIVE."

This is where calculus and the derivative come in.

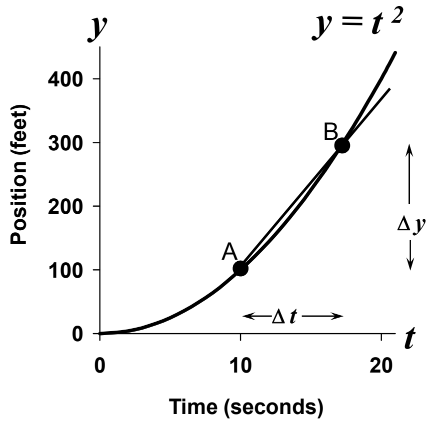
Here we have a function that describes the position of an object that is moving along the y axis with respect to time.

The Derivative



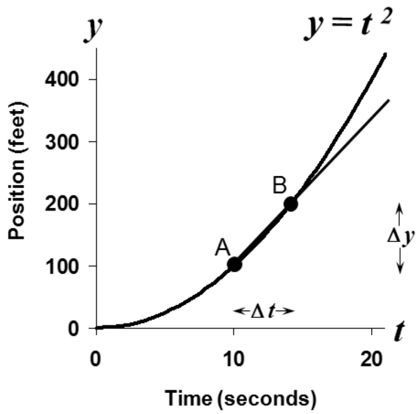
The way calculus addresses the situation of finding the instantaneous slope of a changing curve at some point, A, is to start with a second point, B, some distance away. The CHORD AB and its slope delta y / delta t could be considered a rough approximation of the instantaneous slope (and rate of change) at Point A, but it is a very rough approximation. Then we start decreasing delta t, causing point B to move closer and closer to point A. Delta t and delta y get smaller and smaller and delta y / delta t becomes a better approximation of the instantaneous slope at A. As delta t approaches zero, the distance between B and A gets extremely small, and the chord AB becomes a tangent to the graph at A and its slope (delta y over delta t) becomes the instantaneous rate of change of the graph at point A. (The following pages demonstrate this showing how its slope decreases to approximate the instantaneous slope at A.)

The Derivative



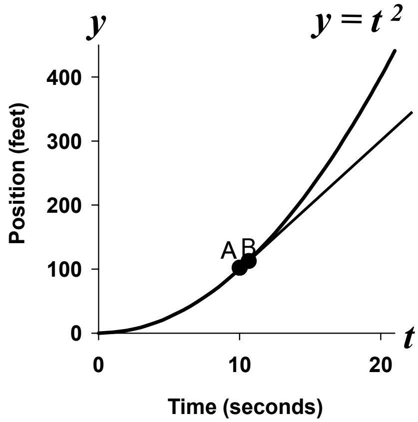
Delta t has become smaller, and the slope of the chord AB has become a better approximation of the slope and instantaneous rate of change of y at Point A.

The Derivative



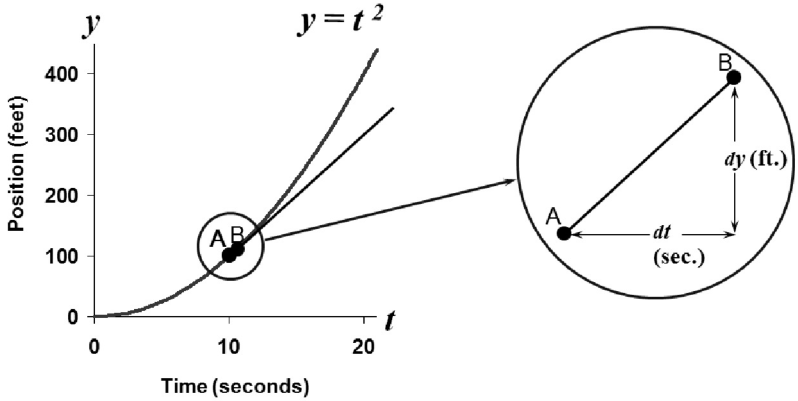
Delta t has become even smaller, and the slope of the chord AB has become an even better approximation of the slope and instantaneous rate of change of y at Point A.

The Derivative



Here is the same graph when B has become extremely close to A and delta t has become arbitrarily close to zero. (I can't even draw in the labels for delta t and delta y any more because they are so close together.)

The Derivative

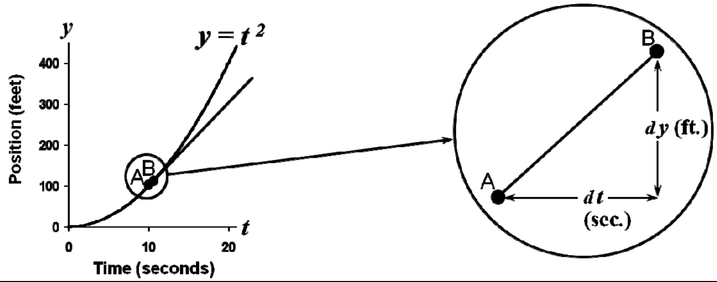


Instantaneous rate of change (first derivative) of position = $\dfrac{dy}{dt}$ = Velocity

In order to see delta t and delta y, the area where A and B are is magnified at the right. As delta t approaches zero calculus changes to deltas to lower case d's (called differentials). We then say that the instantaneous rate of change of the function at point A is dy over dt and define this as what is called the DERIVATIVE, which describes the instantaneous rate of change of y with respect to t.

In the case of this example where the graph is of position (feet) vs. time (seconds), the rate of change of position is velocity (feet per second).

The Derivative



Instantaneous rate of change (first derivative) of position = $\frac{dy}{dt}$ = Velocity

if: $y = t^A$ *then:* $\frac{dy}{dt} = At^{A-1}$ *so for:* $y = t^2$: $\frac{dy}{dt} = 2t$ feet per sec.

$$\frac{dy}{dt} = y' = \dot{y}$$ **(First derivative)**

By going through a lot of steps that are beyond the scope of this discussion, calculus comes up with formulas that give the equation of the derivative if we know the equation of the original function.

For a formula of the form y =t^A the equation of the derivative is shown on the figure above. (For the purpose of this discussion, you don't need to remember this formula.)

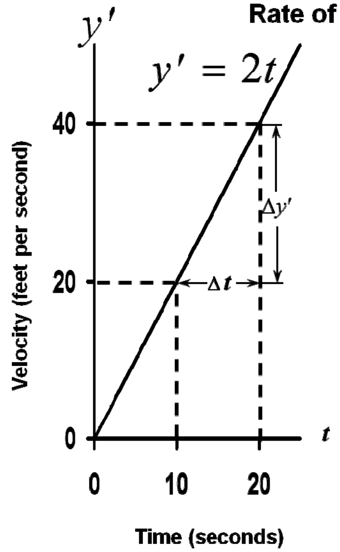
In the case of y =t^2 , the appropriate formula gives a derivative or instantaneous rate of change 2t for all values of t.

That is, when t equals 10 seconds, the velocity is 20 feet per second and when t equals 20 seconds, the velocity equals 40 feet per second.

By definition the instantaneous rate of change is called the first derivative and is denoted by dy / dt (very often read as "dy, dt" , "y prime" or in the case of time varying functions "y dot."

The Second Derivative

Rate of Change of a Rate of Change



Rate of change of velocity $= \dfrac{\Delta y'}{\Delta t}$

$= \dfrac{20}{10} \dfrac{\text{ft. per sec.}}{\text{second}}$

$= 2$ feet per second per second

= Acceleration

Derivative of a derivative: second derivative

$$\frac{d^2y}{dt^2} = y'' = \ddot{y}$$

Very often, the rate of change of a derivative is useful and has physical meaning. This is the graph of the derivative of the function on the previous page. The rate of change of velocity is acceleration.

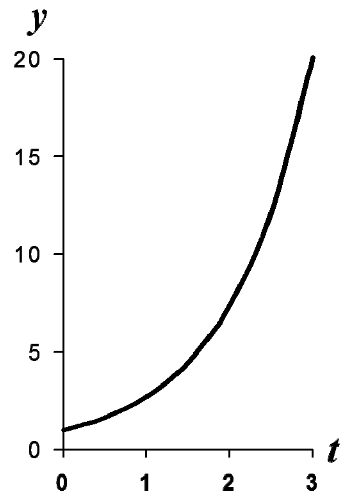
This one is easy to solve, because the function is a linear one. Its rate of change at all points is 2, and because the units of y prime are feet per second, and the units of t are seconds, the rate of change of y prime (velocity) is 2 feet per second per second which is acceleration.

The derivative of a derivative is defined as the second derivative and can be denoted by d^2y / dt^2 (very often read as “d two y by dt squared,” “y double prime” or in the case of time varying functions “y double dot.”)

Natural Growth

In nature it is common for the time rate of change of things ($\dot{y}$) to equal their size (y) at that instant.

- $\dot{y} = y$
- This is called an **“Exponential Relation”**



The next subject is the **exponential function**.

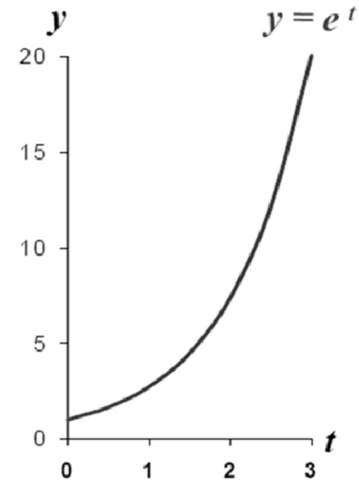
In nature, one of the most common ways for things to change is the way in which their rate of change with respect to time (y dot) is proportional to their size (y) at that instant. A population that is twice as big grows at twice the rate. When you are four times as wealthy, you will accumulate wealth four times as fast. This is called an **exponential relation.**

When y = 5, the instantaneous slope of the graph (its derivative) equals 5.

When y = 10, the instantaneous slope of the graph (its derivative) equals 10.

The Exponential Function

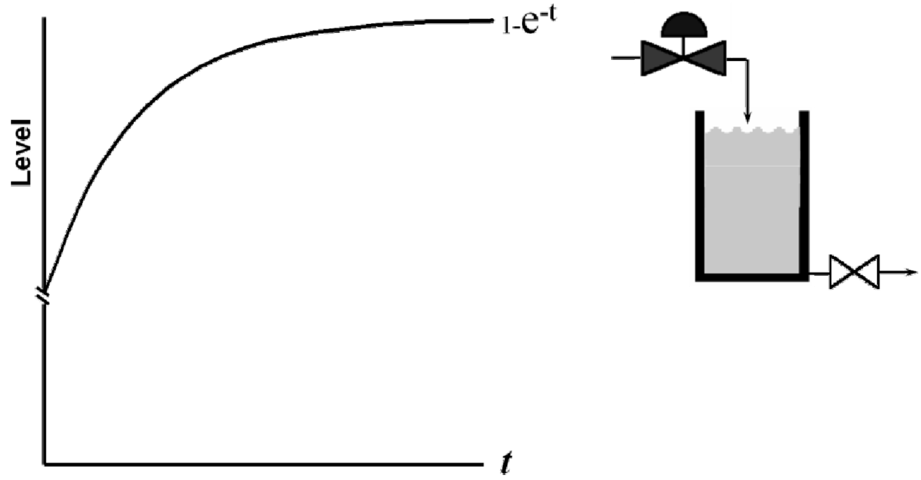
$$y = e^t$$
$$\dot{y} = y$$
$$e = 2.7182...$$



In order to describe the behavior of things that follow the principle of natural growth we define what we call the **Exponential Function,** which has the formula $y = e^t$. For the exponential function to describe things that change at a rate equal to their size (that its derivative (y dot) at any point is equal to its value) it turns out that e must have the value 2.7182 ... (to as many decimal places as desired).

This makes e one of those special numbers like pi, that describes a natural relationship.

Exponential Decay

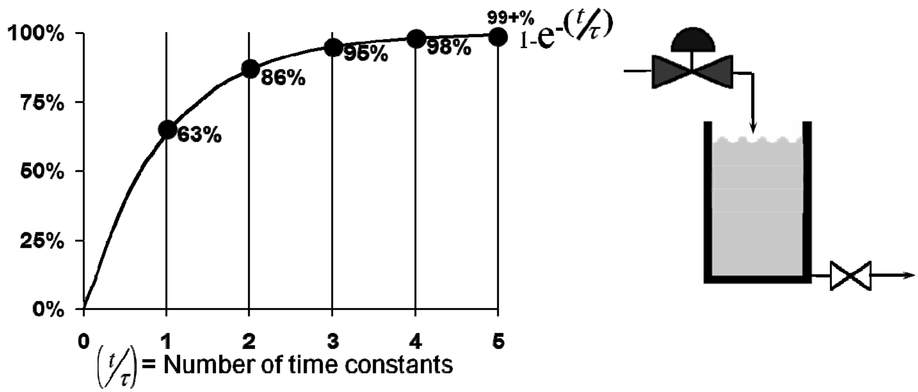


A very important waveform or graph that describes the behavior of many process systems is that of the **Exponential Decay** where $y = 1-e^{-t}$. Here the rate of increase in y is INVERSLY proportional to y. (The rate of increase becomes LESS as y gets bigger.

In the diagram, the flow of water into and out of the tank are constant and the level has stabilized and is also constant. If we were to **suddenly** move the control valve stem so as to make a **step** increase the flow into the tank by, say 5%, the level in the tank would begin to rise. As soon as the level increases, the head at the bottom of the tank also increases and the flow out of the tank increases. As a result, the rate of change of the level decreases. At some point the outflow equals the new higher inflow and the level stabilizes at a new, higher point. The graph shows how the level will change with time.

Exponential Decay

First Order Lag



Time Constant: The time required for a system that follows the exponential decay **(a first order system)** to complete 63% of its total response. (Actually 63.21205…)

This is a more generalized and more common way of looking at the exponential decay or what is often called a **first order lag.** Here we look at the response between zero (before the process has started to respond) and 100% (when the process has reached its final value). For example, if the tank level went from 16 feet to 18 feet, the total response (100%) is 2 feet. We also normalize the time scale, using what is called the **Time Constant.** The scale is in numbers of time constants**. (NOTE** that because the horizontal scale is now in numbers of time constants and not time, the equation is now 1-e raised to the minus t over tau instead of 1- e to the -t.)

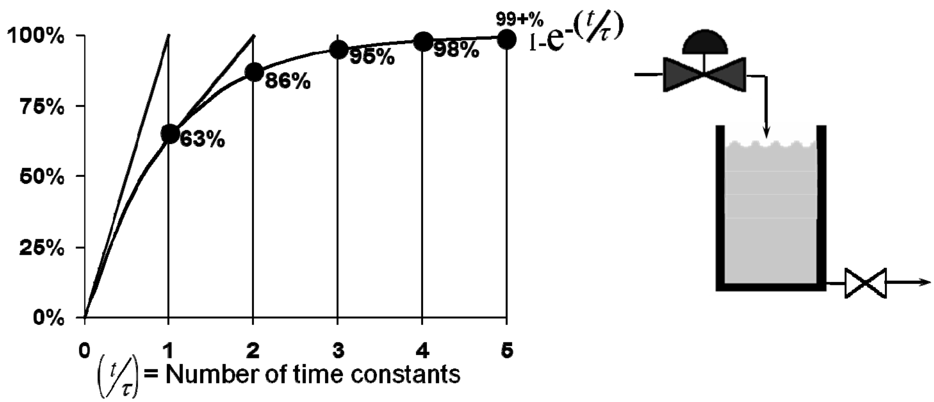
The time constant is another of those special numbers (like pi and e). The time constant is defined as “The time required for a system that follows the exponential decay to complete 63% of its total response.” (Actually 63.21…% to as many decimal places as you want.) Usually when talking about time constants, the decimals of a percent are ignored.) We use the symbol Tau for the time constant, so t over tau of 1.0 means that a time equal to one time constant has elapsed. A t over tau of 2.0 means that a time equal to two time constants has elapsed.

There are two interesting features of the time constant: (1) During the first time constant, the process responds through 63% of the total response. During the second time constant it responds through 63%

of the remaining portion of the total response and so on. (This only is true if the time constant has the value of 63.2...%.) Most people agree that for all practical purposes a system has reached its full response after 5 time constants. If we were to be mathematically correct, it would never reach final value.

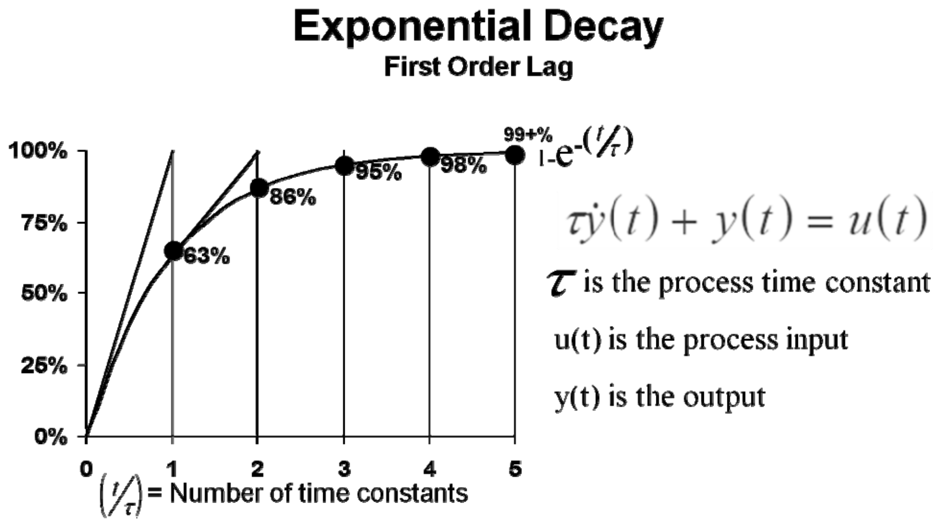
Exponential Decay

First Order Lag



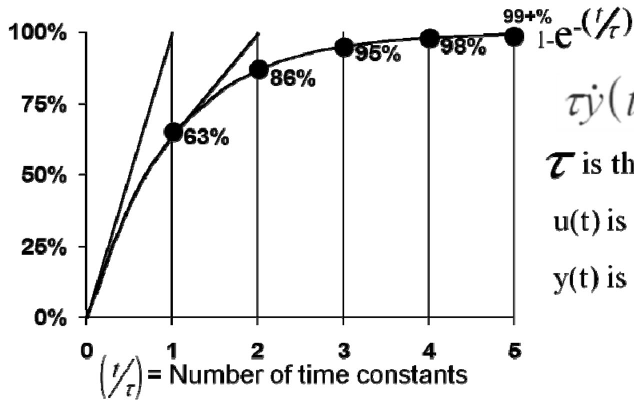
Time Constant: The time required for a system that follows the exponential decay (**a first order system)** to complete 63% of its total response. (Actually 63.21205…)

(2)If the response of a first order system continued at its initial rate (slope), the response would reach 100% in 1 time constant. In fact, if you take the slope at any point, it will intersect the 100% line on the graph after 1 time constant.



The reason this type of response is called a “First Order Lag” is because this type of response is described by a first order differential equation. That is an equation that contains a first order derivative (y dot) but no higher derivatives. (See the equation at the right side of the figure.)

Exponential Decay
First Order Lag



$$\tau\dot{y}(t) + y(t) = u(t)$$

τ is the process time constant

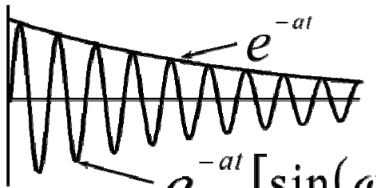
u(t) is the process input

y(t) is the output

$$\int\left[\tau\dot{y}(t) + y(t)\right]dt = \int u(t)dt$$

The way this type of equation is solved is by integrating it as shown at the bottom of the figure. It turns out that integrating many of the waveforms (functions of time) that are encountered is not all that easy to do, and in general solving these equations is very difficult.

The Laplace Transform

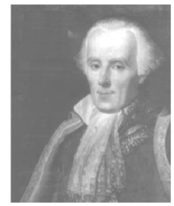


$$e^{-at}[\sin(\omega t)]$$

$$e^{-at}\,[e^{-j\omega t}]$$

$$Let s = a + j\omega$$

$$e^{-st}$$



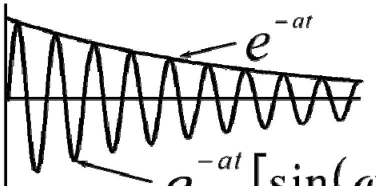
Pierre-Simon Laplace
(Mathematics, astronomy, probability theory)
1749 to 1827

The Laplace transform is named in honor of mathematician and astronomer Pierre-Simon Laplace, who used the transform in his work on probability theory.

Laplace noted that although the waveforms of most time functions were difficult to integrate, the exponential decay and the sin function have derivatives that are of the same form as the original function and thus are easy to integrate. Also integrating a decaying sine wave (as shown at the top left of the figure) is also fairly easy (at least for mathematicians).

There are several ways to express a decaying sine wave. The one with the “sin”function is the most obvious. If you are willing to sit through some tedious mathematical manipulation you can show that the others are equivalent. The one with e^{-st} is the one used in the Laplace transform where “s” is called the Laplace operator.

The Laplace Transform

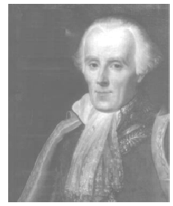


$$e^{-at}\left[\sin(\omega t)\right]$$

$$e^{-at}\left[e^{-j\omega t}\right]$$

$$Let s = a + j\omega$$

$$e^{-st}$$



Pierre-Simon Laplace
(Mathematics, astronomy, probability theory)
1749 to 1827

Definition of the Laplace Transform

$$L\left[f(t)\right] = \underbrace{\int_0^\infty e^{-st}\overbrace{f(t)dt}^{\text{Difficult}}}_{\text{Easy}}$$

The figure on this page is the same as on the previous page except for the definition of the Laplace Transform shown inside the box.

The Laplace transform is a method of transforming hard to solve differential equations into easy to solve algebraic equations. The method is based on multiplying any (hard to integrate) function of time by the easy to integrate decaying sine function, and then integrating the resulting function.

The Laplace Transform

f(t)	Laplace Transform
$\dot{y}$	$sY(s)$
$\ddot{y}$	$s^2Y(s)$
1 Unit step at t=0	$1/s$
$1-e^{-t/\tau}$	$\frac{1}{s(\tau s+1)}$
$f(t-L)$ Delay of L	e^{-Ls}

Constants, such as τ (time constant) and k (gain), are not transformed, but keep their original value.

Extensive tables of common functions and their Laplace transforms have been developed, so we do not have to bother ourselves with the integration part of the transform in order to use it.

At the right of the figure is a very abbreviated table of Laplace transforms. You can see that the transformed functions do not contain time or derivatives and thus are simple algebraic expressions.

The Laplace Transform

First order lag

$$\tau \dot{y}(t) + y(t) = u(t)$$

$f(t)$	Laplace Transform
$\dot{y}$	$sY(s)$
$\ddot{y}$	$s^2Y(s)$
1 Unit step at t=0	$1/s$
$1-e^{-t/\tau}$	$\dfrac{1}{s(\tau s+1)}$
$f(t-L)$ Delay of L	e^{-Ls}

Constants, such as τ (time constant) and k (gain), are not transformed, but keep their original value.

The next pages demonstrate how the Laplace Transform is used to solve a differential equation. This demonstration will solve the differential equation for a “First order lag” which we saw previously and has been repeated on the left side of the figure.

You will note that no calculus or integration is involved, only algebraic manipulation.

The Laplace Transform

First order lag

$$\tau \dot{y}(t) + y(t) = u(t)$$
$$\tau s Y(s) + Y(s) = U(s)$$

$f(t)$	Laplace Transform
$\dot{y}$	$sY(s)$
$\ddot{y}$	$s^2Y(s)$
1 Unit step at t=0	$1/s$
$1-e^{-t/\tau}$	$\dfrac{1}{s(\tau s+1)}$
$f(t-L)$ Delay of L	e^{-Ls}

Constants, such as τ (time constant) and k (gain), are not transformed, but keep their original value.

Replace each term with it's Laplace transform as shown on the second line on the left side of the figure.

Tau, the time constant is a constant, and the transform of a constant is simply the same constant, so tau in the time domain simply transforms to tau in the Laplace domain.

From the table, the first derivative (y dot) transforms to sY(s). The s in parentheses is to remind us that functions in the Laplace transform are functions of s, the Laplace operator.

y(t) and u(t) are at this point unknown time functions (the input signal or waveform u, and the resulting output waveform, y), so we simply transform them as "the Laplace Transform of a time function to be determined later." That is, y(t) transforms to Y(s) and u(t) transforms to U(s). By convention we use capital letters to clarify that we are talking about the transform. We also write it as a function of "s" the Laplace operator, to further clarify that we are talking about a function that is in the Laplace domain.

The Laplace Transform

First order lag

$$\tau \dot{y}(t) + y(t) = u(t)$$
$$\tau s Y(s) + Y(s) = U(s)$$
$$Y(s) \times (\tau s + 1) = U(s)$$
$$Y(s) = \frac{U(s)}{\tau s + 1}$$

$f(t)$	Laplace Transform
$\dot{y}$	$sY(s)$
$\ddot{y}$	$s^2Y(s)$
1 Unit step at t=0	$1/s$
$1-e^{-t/\tau}$	$\dfrac{1}{s(\tau s+1)}$
$f(t-L)$ Delay of L	e^{-Ls}

Constants, such as τ (time constant) and k (gain), are not transformed, but keep their original value.

Using simple algebra, rearrange the transformed equation so that it is solved for the Laplace transform of the output [Y(s)].

Y(s) was factored out of the left hand side of the second equation as shown in the third line.

Finally both sides of the third equation were divided by (tau s + 1), giving us the fourth line where the transformed differential equation is solved for the Laplace transform of the output from a first order system, Y(s) given the Laplace transform of any given input, U(s).

The Laplace Transform

First order lag

$$\tau \dot{y}(t) + y(t) = u(t)$$

$$\tau s Y(s) + Y(s) = U(s)$$

$$Y(s) \times (\tau s + 1) = U(s)$$

$$Y(s) = \frac{U(s)}{\tau s + 1}$$

Find the output, Y(s), for a unit step at t = 0

Set U(s) = 1/s

$$Y(s) = \frac{1/s}{\tau s + 1} = \boxed{\frac{1}{s(\tau s + 1)}}$$

$f(t)$	Laplace Transform
$\dot{y}$	$sY(s)$
$\ddot{y}$	$s^2 Y(s)$
1 Unit step at t=0	$1/s$
$1 - e^{-t/\tau}$	$\frac{1}{s(\tau s + 1)}$
$f(t-L)$ Delay of L	e^{-Ls}

Constants, such as τ (time constant) and k (gain), are not transformed, but keep their original value.

Now that we know the general form of the transformed output waveform given any transformed input waveform (the fourth line), let's see what the response would be for a unit step change in input.

Replace U(s) with the transform of a unit step which is 1/s and simplifying we get the expression in the box which is the Laplace transform of the output from a first order system when the input is a unit step at time zero.

The Laplace Transform

First order lag

$$\tau \dot{y}(t) + y(t) = u(t)$$

$$\tau s Y(s) + Y(s) = U(s)$$

$$Y(s) \times (\tau s + 1) = U(s)$$

$$Y(s) = \frac{U(s)}{\tau s + 1}$$

Find the output, Y(s), for a unit step at t = 0

Set U(s) = 1/s

$$Y(s) = \frac{1/s}{\tau s + 1} = \boxed{\frac{1}{s(\tau s + 1)}}$$

Taking the inverse transform:

$$\boxed{y(t) = 1 - e^{-t/\tau}}$$

f(t)	Laplace Transform
$\dot{y}$	$sY(s)$
$\ddot{y}$	$s^2 Y(s)$
1 Unit step at t=0	$1/s$
$1 - e^{-t/\tau}$	$\frac{1}{s(\tau s + 1)}$
$f(t - L)$ Delay of L	e^{-Ls}

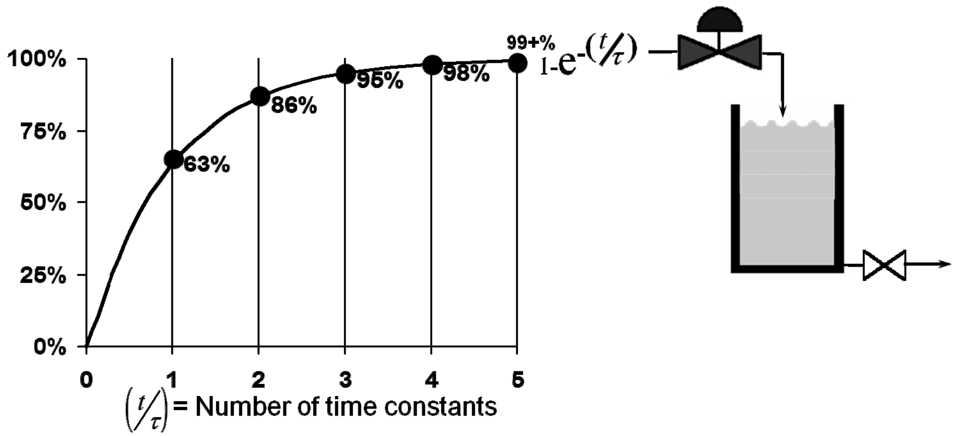
Constants, such as τ (time constant) and k (gain), are not transformed, but keep their original value.

The final step is to convert the Laplace transform of the output waveform to a time based function. Looking down the list of Laplace transforms, we see that the one next to the bottom of the list is identical to the Laplace transform of the output, Y(s), of the first order system to a unit step input.

The corresponding time function is shown in the box at the lower left. Look at the next page and you will recognize it as the time function that describes the response of a first order system to a step change in input.

Using the Laplace Transform

First order lag



$$y(t) = 1 - e^{-t/\tau}$$

In this case, the time function we came up with is the familiar first order response to a step increase. (No surprise here since in this case we already know the time response of a first order system to a step input.)

The same process is used by control engineers for determining the output of dynamic systems based on various types of inputs.

Using the Laplace Transform

First order lag

$$\tau \dot{y}(t) + y(t) = u(t)$$
$$\tau s Y(s) + Y(s) = U(s)$$
$$Y(s) \times (\tau s + 1) = U(s)$$
$$\rightarrow Y(s) = \frac{U(s)}{\tau s + 1}$$

Find the output, Y(s), for a unit step at t = 0

Set U(s) = 1/s

$$Y(s) = \frac{1/s}{\tau s + 1} = \boxed{\frac{1}{s(\tau s + 1)}}$$

Taking the inverse transform:

$$\boxed{y(t) = 1 - e^{-t/\tau}}$$

$f(t)$	Laplace Transform
$\dot{y}$	$sY(s)$
$\ddot{y}$	$s^2 Y(s)$
1 Unit step at t=0	$1/s$
$1 - e^{-t/\tau}$	$\frac{1}{s(\tau s + 1)}$
$f(t-L)$ Delay of L	e^{-Ls}

Constants, such as τ (time constant) and k (gain), are not transformed, but keep their original value.

Our final topic is that of "Transfer Functions", which is a convention used by control system engineers to describe how a dynamic element acts on a time changing input.

Going back two pages (repeated in the above figure) we had derived the Laplace transform of the output from a first order system for any type of input signal. (See the fourth line on the left side of the figure.)

Transfer Function

First order lag

$$Y(s) = \frac{U(s)}{\tau s + 1}$$

Transfer Function: Output / Input

$$\frac{Y(s)}{U(s)} = \frac{1}{\tau s + 1}$$

f(t)	Laplace Transform
$\dot{y}$	$sY(s)$
$\ddot{y}$	$s^2Y(s)$
1 Unit step at t=0	$1/s$
$1-e^{-t/\tau}$	$\frac{1}{s(\tau s+1)}$
$f(t-L)$ Delay of L	e^{-Ls}

Constants, such as τ (time constant) and k (gain), are not transformed, but keep their original value.

By convention, control system engineers talk about the "Transfer Function" of a dynamic element. The "Transfer Function" is defined as the "Output of a dynamic element divided by the input to that element." (Always expressed in the Laplace domain.)

We obtain the transfer function (output divided by input) of our first order system by dividing both sides of the output equation by U(s), yielding the transfer function of 1/tau s + 1.

Control system engineers very often describe dynamic systems by drawing a box, or group of connected boxes, each with a transfer function written inside, and these are referred to as block diagrams. In the chapter on "Dynamics of Industrial Processes" we will use simple block diagrams (mainly with the transfer function of first order systems). When you see a block in that chapter with the transfer function shown in the above figure, there are two things you should recognize: 1) The form of the transfer function (1/tau s + 1) tells you that the box represents a process element that has a first order response (also referred to as a "first order lag") and 2) the number that replaces the placeholder "tau" is the element's time constant.

Chapter 2

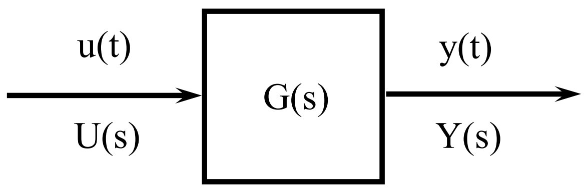
Dynamics of Industrial Processes

Transfer Function

Block Diagram

Transfer Function: Output / Input

$$\frac{Y(s)}{U(s)} = G(s)$$



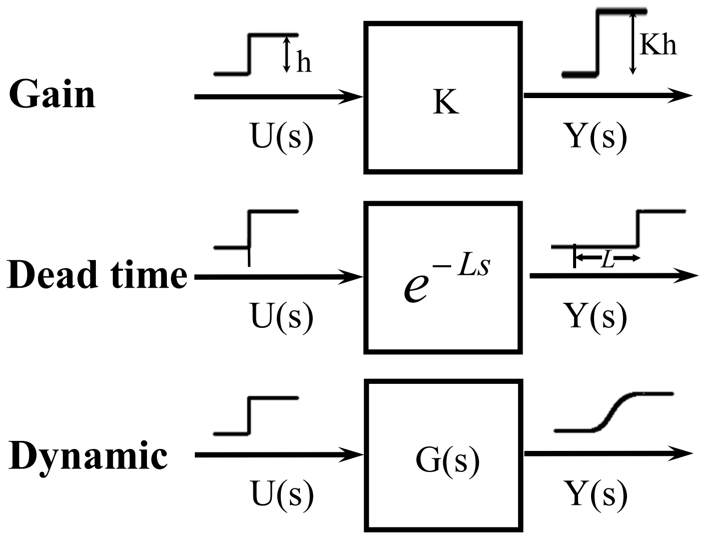
In the Review of Mathematics for Process Dynamics chapter we worked up to the subject of transfer functions.

A transfer function tells us what will happen to an input signal as it passes through the process element and therefore tells us what sort of dynamic behavior that element exhibits. The transfer function of a process element is the Laplace transform of its output divided by its input.

A common way of discussing the dynamics of process elements is to draw block diagrams of the various elements, with their transfer function inside the box. The generic transfer function is often referred to as “G”, very often with the “s” to remind us that the transfer function is in the “s” domain, that is the Laplace transform of a time function.

Usually lower case letters are used when referring to functions or signals in the time domain, and upper case letters used to describe functions or signals in the Laplace or s domain. Including the t or s helps to clarify whether we are talking about the Laplace domain or the time domain. We will be using block diagrams in the rest of this chapter.

Transfer Function
Block Diagram



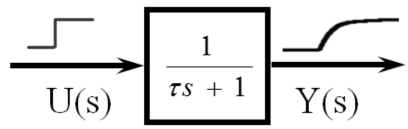
We will discuss three types of transfer functions.

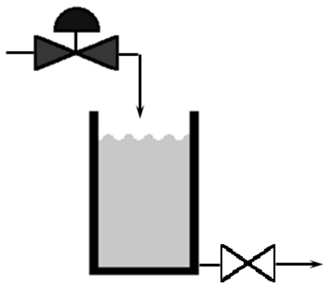
1. Gain. This has no dynamics, but the output waveform is the same as the input waveform except multiplied by the gain, a numerical value. there is no time dependence. the transfer function is simply K, which is numerically equal to the gain in the time domain (t).

2. Dead time does not change the shape of the input waveform, but simply delays it. An example would be a change in pulp consistency at time zero, but it is not measured by the consistency transmitter until L seconds later because there is a finite distance between the dilution valve and the transmitter. The "L" is the time lag and s is simply the Laplace operator. The fact that it is e^{-Ls} is simply the result of the integration that gives the Laplace transform. All you need to remember is that when you see "e^{-Ls}" you know it represents a time delay of "L."

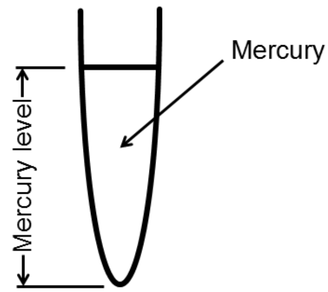
3. Dynamic elements change the shape of the input waveform. We will look at these in some detail.

First Order Lag





Tank Level



Thermometer

We discussed the first order lag, and its transform in the Review of Mathematics for Process Dynamics chapter .

This one is worth remembering, because it keeps coming up.

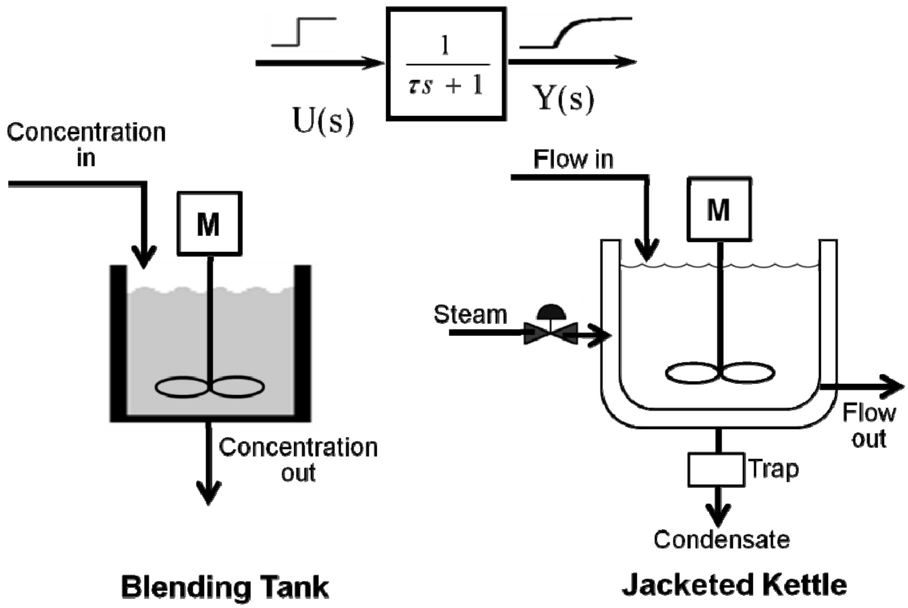
The “1” in the numerator is for a generic gain of 1. The Tau is the time constant in minutes or seconds.

Some examples of processes that exhibit a first order response are:

Tank Level: response to changes in incoming flow rate.

Thermometer: response of the height of the mercury column to changes in surrounding water temperature.

First Order Lag



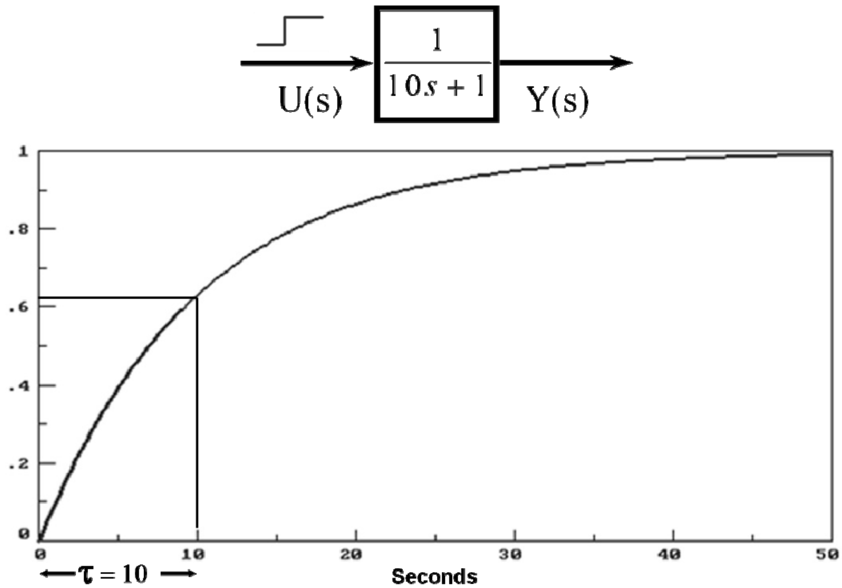
Some additional examples of first order processes:

Blending tank: response of exiting concentration to changes in incoming concentration. (Assuming no chemical reaction and constant level.)

Jacketed kettle: response of exiting temperature to changes in either feedstock temperature or steam temperature.

Next we will look at the responses of a first order system to step, ramp and sinusoidal inputs.

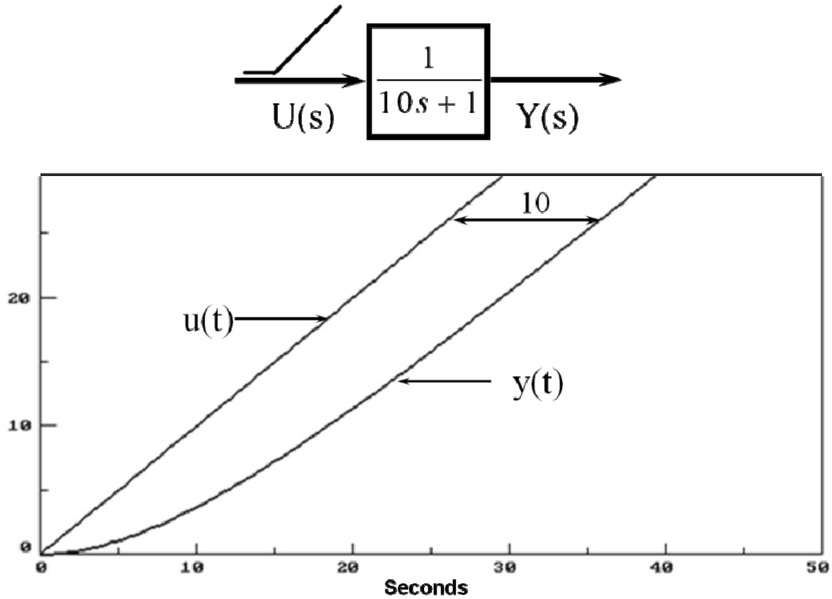
First Order Lag



This is the step response of a first order system with a time constant of 10 to a step input. No surprises here, since we have discussed this case previously. The placeholder, tau, in the transfer function on the previous page has been replaced with the numerical value of the time constant of this particular system which is ten seconds.

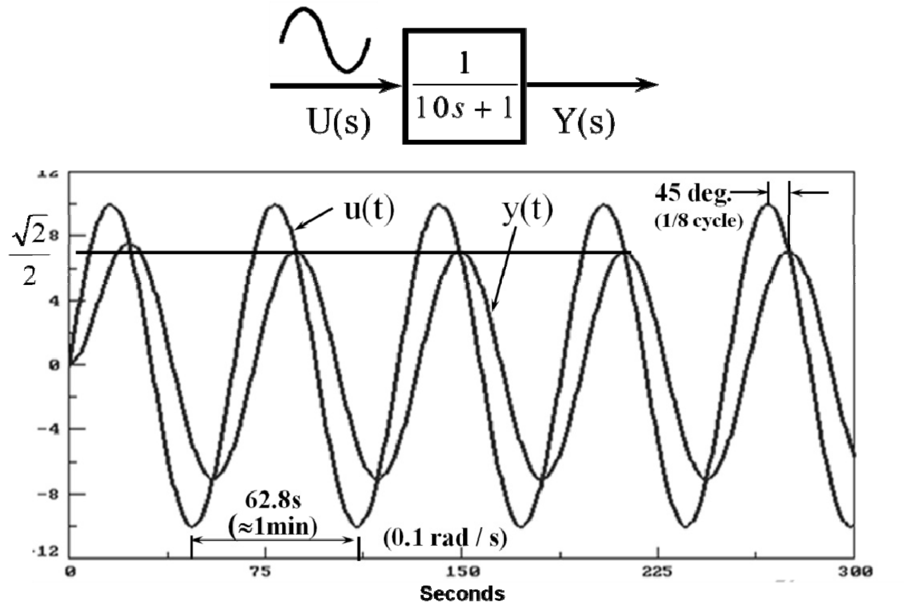
On this page the units of the time constant are clearly "seconds" because the graph's time scale is labeled "Seconds." However, when the scale is not known, a time constant of 10 could be seconds or minutes (or even hours). Remember that the "s" in the transfer function does not stand for seconds, but is the Laplace operator.

First Order Lag



This is the response of a first order system with a time constant of 10 to a ramp input. Note that after the transient has died out, the response becomes a ramp of the same slope as the input, which lags the input by one time constant.

First Order Lag



This is the response of a first order system with a time constant of 10 seconds to a sinusoid input with a period of 62.8 sec (an angular frequency of 0.1 rad / second).

Note that there is a small transient in the first cycle which is slightly higher than the rest.

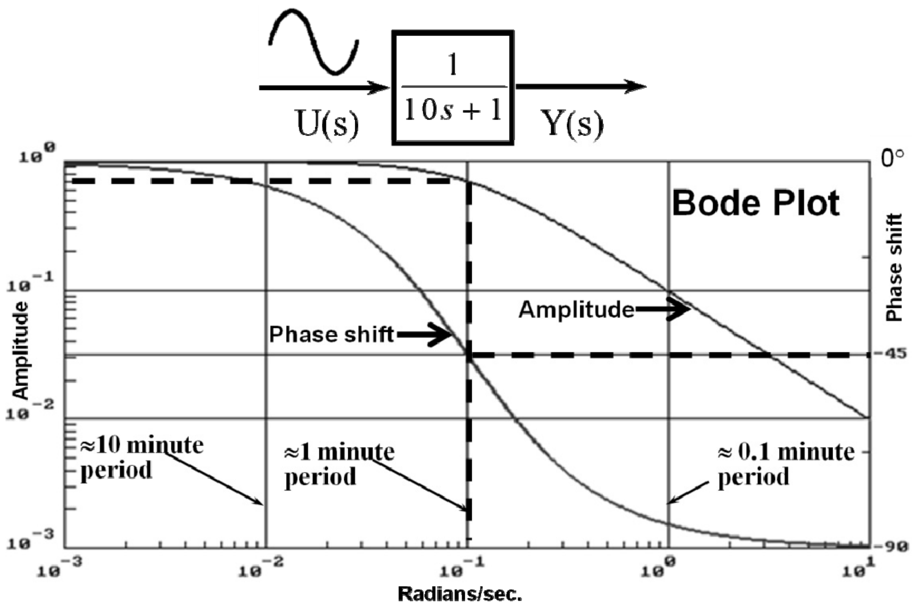
In this case, the response has an amplitude of approximately 0.7 (actually 1/ sqrt. 2) times the input. (The mathematics police require us to write this as “the square root of two over two.”)

The output is lagging the input by 45 degrees. (1/8 th of a cycle)

If the frequency of the input were changed, both the amplitude and phase shift of the output would also change, but its waveform would still be a sinusoid.

It is not practical to draw graphs of every possible input frequency, so we have a different way of expressing the amplitude and phase shift for various frequencies. (See next page.)

First Order Lag



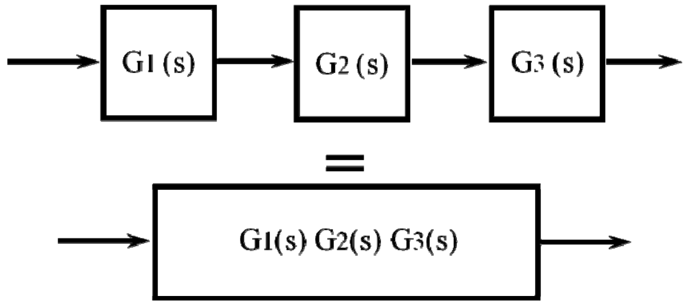
A Bode plot graphs both the magnitude and phase shift of the response of a system to sinusoids of various frequencies.

Here we see that the amplitude ratio of 0.7 (1/ sqrt.2) and the phase shift of 45 degrees correspond to the time response in the previous slide.

At frequencies significantly greater than 0.1 rad/sec (periods significantly less than about 1 minute, say 1/10th of a minute), the system will attenuate the disturbance. At frequencies considerably slower than 0.1 rad/sec. (periods greater than one minute (say 10 minutes) there will be no attenuation of the disturbance.

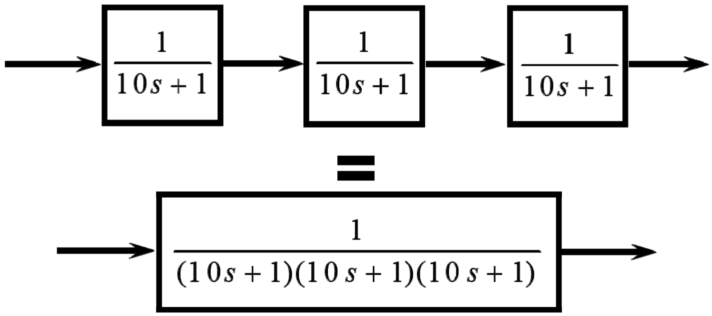
This might be a mixing tank where the incoming temperature is fluctuating about an average point. The tank can smooth out rapid changes, but we would need a control system to smooth out the slow changes.

Blocks in Series



One important principle of transfer functions and block diagrams, is that if you have several blocks in series, they behave just like a single block whose transfer function is the product of the individual transfer functions.

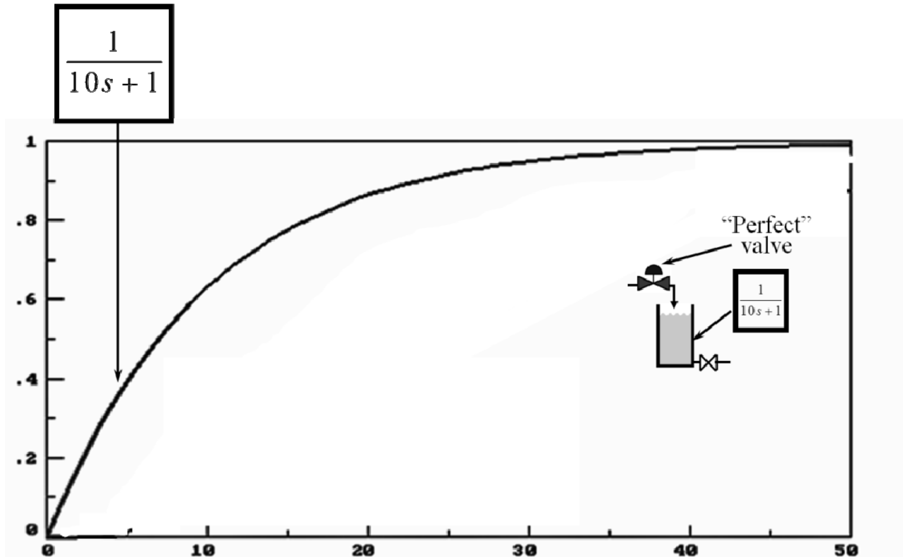
Blocks in Series



For example three process elements (blocks), which each exhibit a first order lag with a time constant of 10, would be equivalent to the block below and could be drawn and analyzed as a single block with the transfer function shown.

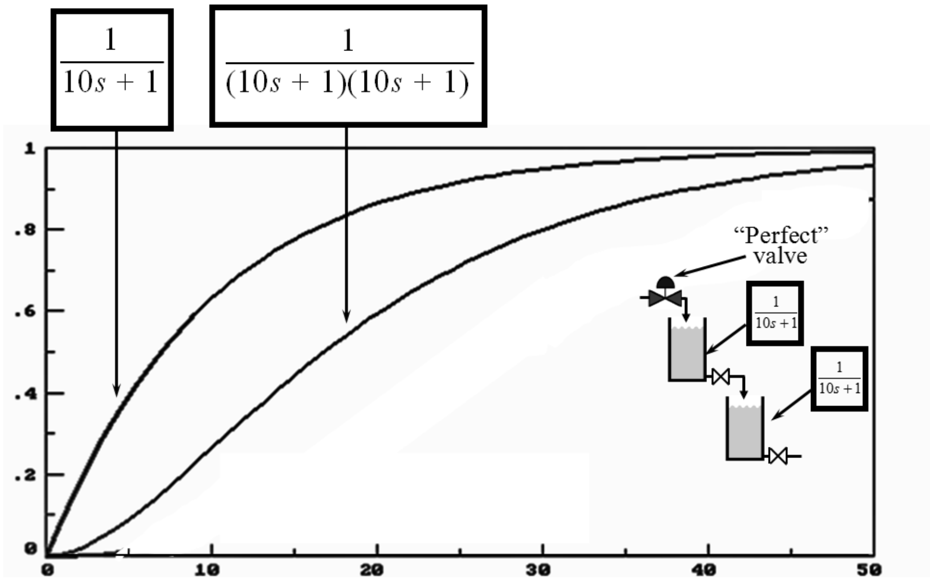
By becoming familiar with the form of the transfer functions of first order lags and how blocks combine, you should be able to look at a block like the one at the bottom and recognize that it is probably the result of three blocks with time constants of 10 in series.

Equal Time Constants in Series



Now we will take a look at the dynamic behavior blocks with equal time constants in series. We start out with the above figure where we have a single process element consisting of a tank where there is a constant inflow from the valve above the tank and a constant outflow at the bottom of the tank. As a result, the level remains constant. The valve is a "Perfect" valve, meaning that it can go instantly from one position to another. At time t = 0 the valve opens a small amount causing a step change in inflow. The tank has a time constant of 10, so the response of the level in the tank is as shown in the figure.

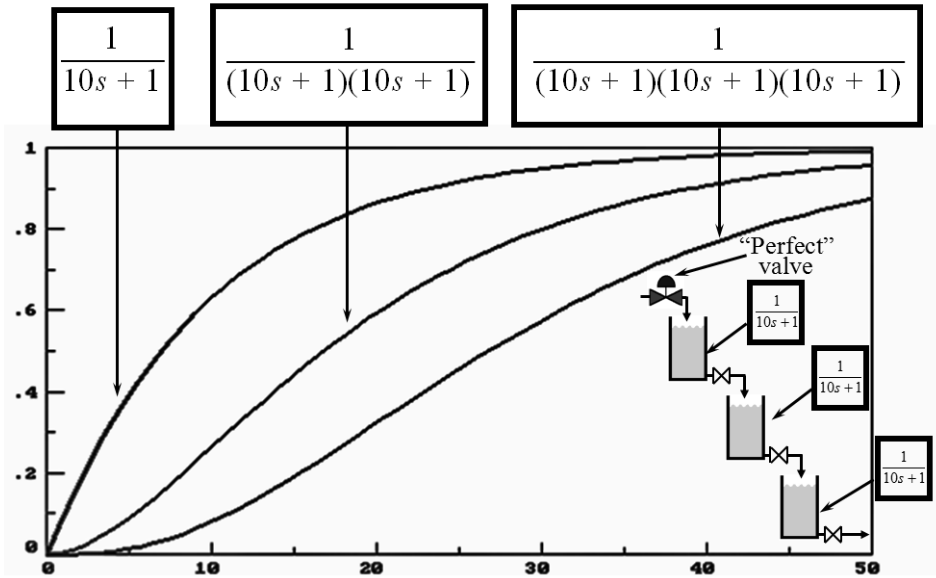
Equal Time Constants in Series



In this figure we have added a second tank in series with the first which also has a time constant of 10 giving a system with two equal time constants in series.

The response of the level in the second tank to a step change in input to the first tank (which represents the response of a system with two equal time constants in series to a step change in input) is much slower than that of a system with a single time constant.

Equal Time Constants in Series

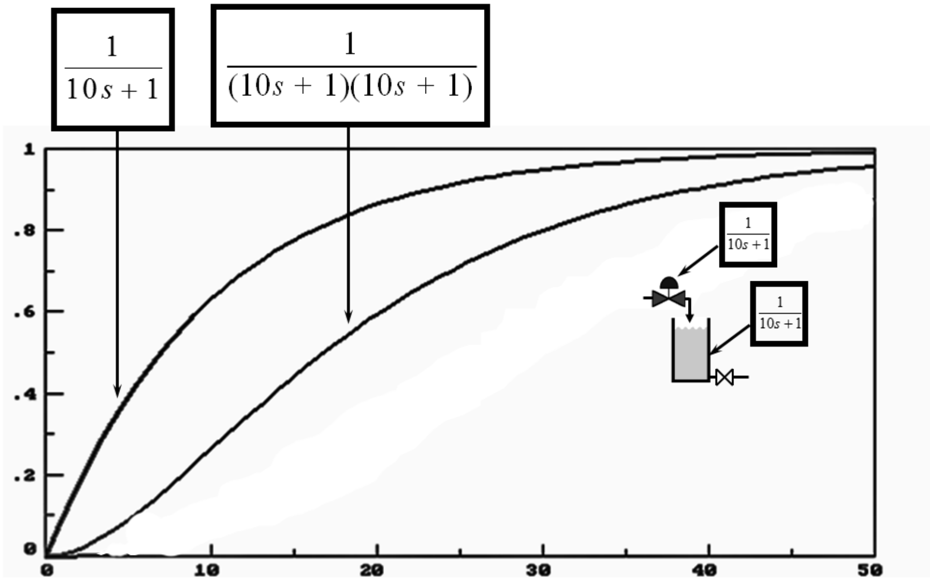


Here we have added a third tank in series, again with a time constant of 10, giving a system with three equal time constants in series.

The response of the level in the third tank to a step change in input to the first tank (which represents the response of a system with three equal time constants in series to a step change in input) is very slow.

The very slow response at the beginning of the response for the second tank and especially the third tank is difficult to distinguish from dead time and has the effect of making the process difficult to control.

Equal Time Constants in Series



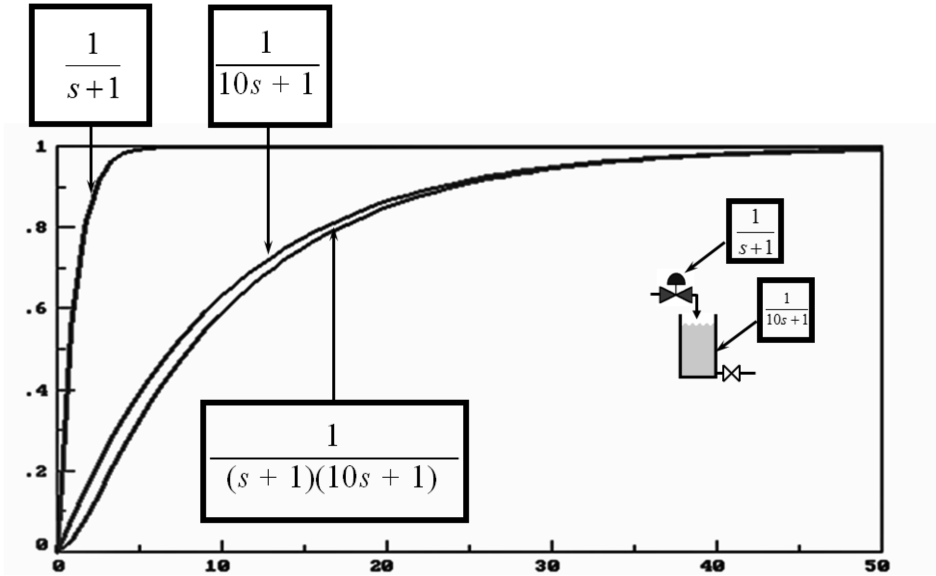
In this example, we have only one of the tanks of the previous example (with a ten second time constant).

Now we will use a real vale that has dynamics of its own. The valve also has a time constant of ten seconds.

The combined response of the system (valve plus tank) is the same as what we got on the previous slide for the first two tanks with a perfect (no dynamics) valve. Two ten second time constants behave the same regardless of what sort of device or process is contributing its time constant to the response of the overall system.

Note that control valves very often have responses that are more complex than first order, but that is a subject for another day.

Unequal Time Constants - One Dominant

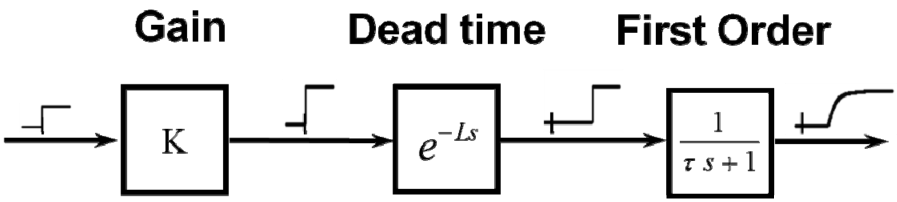


Here we have replaced the control valve that has a ten second time constant with a much faster valve, one with a one second time constant. (When the "s" in the denominator of the transfer function has no number in front of it, that is the same as multiplying it by "1.0", that is, writing "s" is the same as writing "1s" for a time constant of 1.)

The combined transfer function for this system is the product of the one second first order lag transfer function of the valve and the ten second first order lag transfer function of the tank.

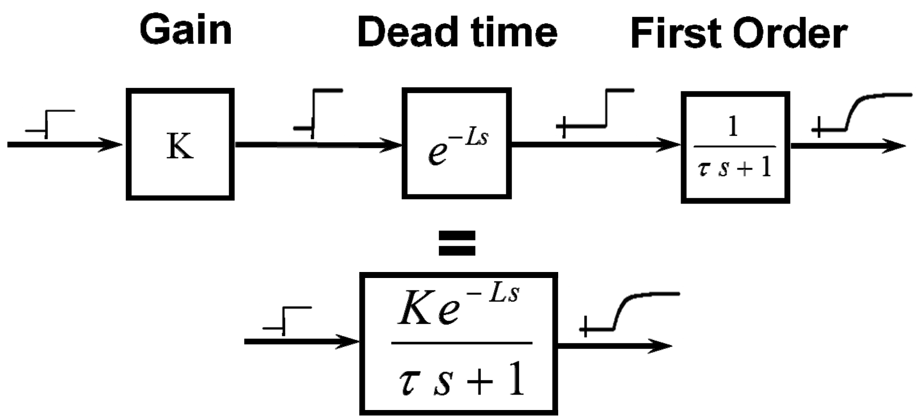
When we have a system with one long time constant and one short time constant, we say the long one is "***dominant***," that is the overall response is very close to what would result if only the long time constant were present, and the short one has very little effect as shown here.

Typical Process Model



Many common processes at least act very much like a combination of GAIN, DEAD TIME and A FIRST ORDER response.

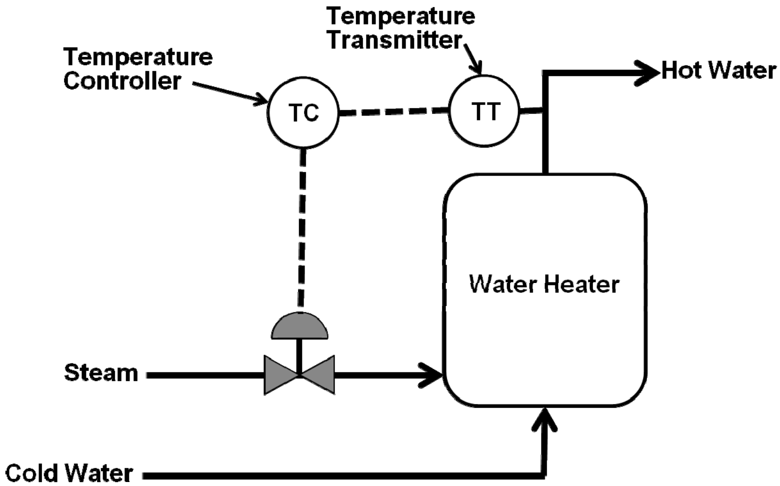
Typical Process Model



For a step input, the output of this system will be a first order response with a time constant of τ seconds (or minutes), it will be delayed by **L** seconds (or minutes) and will be **K** times as big as the input was.

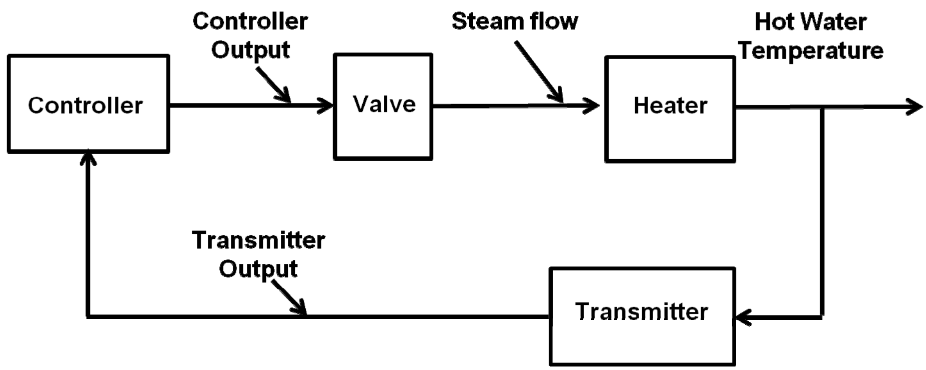
Since it is not always even clear which part of the process is responsible for each part of the response we use the rule of combining the gain, the dead time (when present) and the first order response (or an approximation of a first order response) into a process model we call a FIRST ORDER PLUS DEAD TIME model.

Typical Process



This is a water heater that we will examine the response of and attempt to model it as a first order plus dead time element.

Typical Process Block Diagram



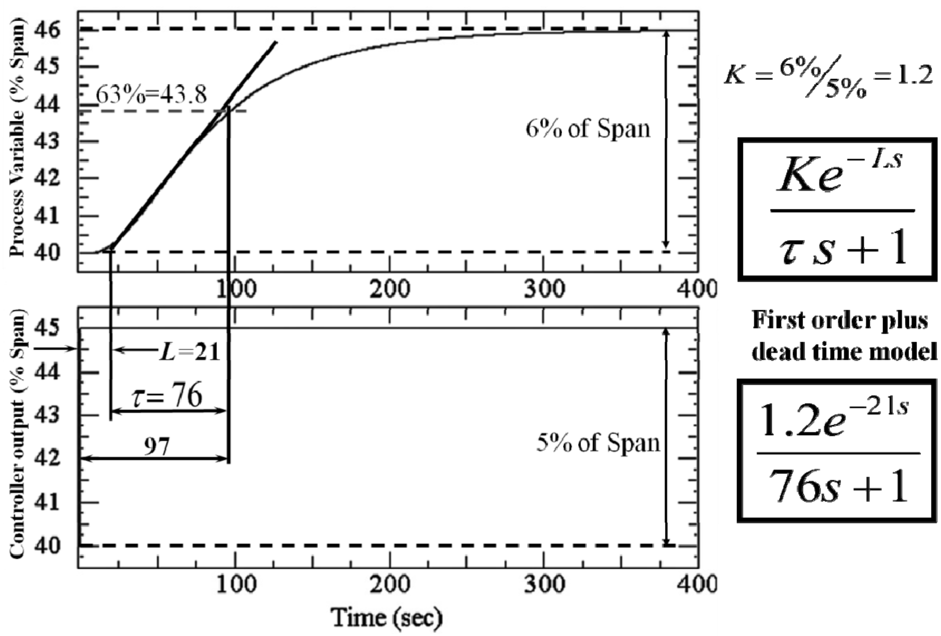
This is a block diagram of the water heater. Each major component is represented by a box. If we knew the transfer functions of each component, we could write it in each box. In this case we do not have specific information on each block.

We could probably approximate the valve as a first order element, hopefully with a minimum amount of dead time.

The heater itself may have more than one time constant and might appear to have dead time because of the interaction of similar time constants.

The transmitter will have a time constant, but probably much faster than the heater, however a temperature sensor in a thermowell will also exhibit a time constant.

Determination of First Order + Dead Time Model



To develop a first order plus dead time model of the water heater, we would perform a "bump" test. We put the controller in manual mode and then step the controller output by a small amount, perhaps 5%, and then record the response of the process. Here the lower graph is the controller output, and the upper graph is the response of the temperature of the water coming out of the water heater as measured by the temperature transmitter.

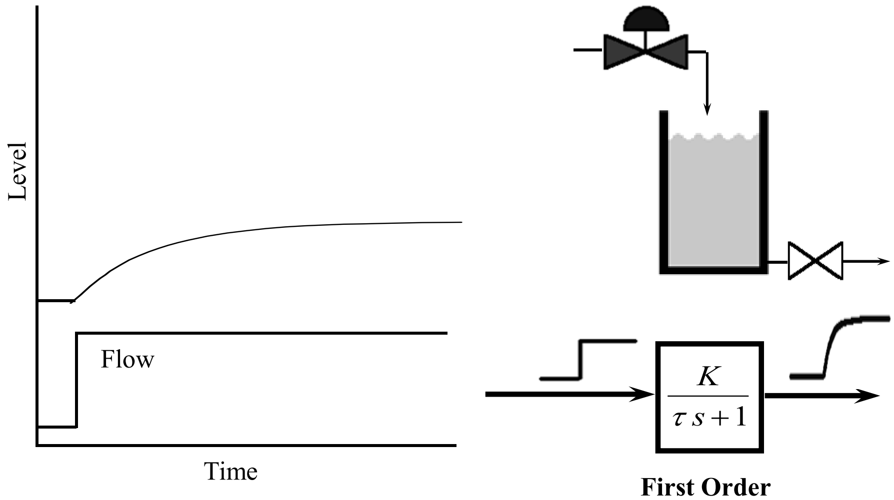
The gain of anything is defined by the change in output (temperature) divided by the corresponding change in input (controller output). In this example the gain is 1.2.

To find the approximate effective dead time, we draw a line that is tangent to the reaction curve at the point of inflection (where the slope is the greatest). We take the effective dead time to be the time between the controller output step and where the tangent line intersects the graph at the process value before the step. In the example, by carefully reading the graph, we find the dead time (L) is 21 seconds.

We will define the approximate time constant as the time between the end of the approximate dead time and the time the process variable reaches 63% of its final value. In the example, by carefully reading the graph, this turns out to be 76 seconds.

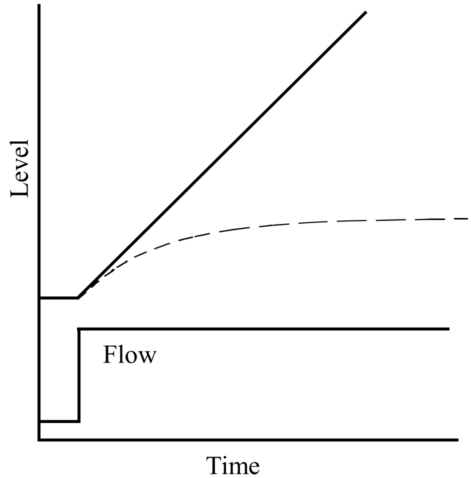
We can now substitute the values we experimentally determined for K, L and tau into the transfer function, and we have a first order plus dead time model of the water heater.

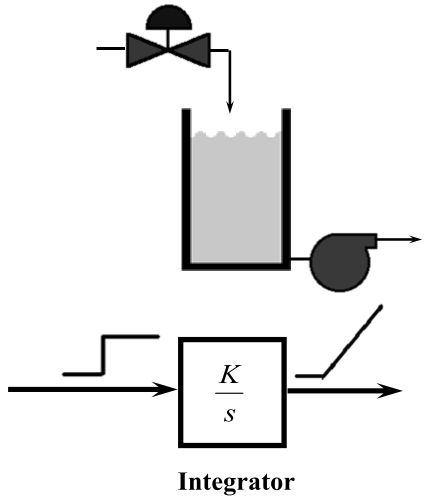
Tank, Outflow Proportional to Head



We talked earlier about the response of the level in a tank that has a valve in the outlet. When the inlet flow suddenly increases by a small amount the level starts going up, but because the pressure (head) at the bottom of the tank is increasing, the flow out also increases. The level will finally settle out at a new higher point. This is a first order process.

Tank, Outflow Constant



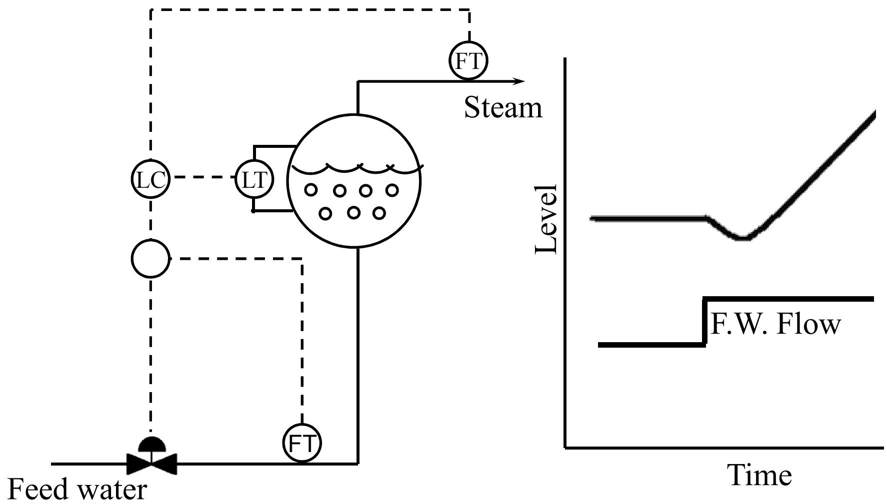


If the flow out of the tank is fixed, say by a pump, a small increase in the inlet flow starts increasing the level, but the increased pressure at the bottom of the tank does not affect the outflow. The result is that the level just keeps going up!

This dynamic system is called and INTEGRATOR. Liquid tanks with fixed outlet flow are the most often encountered integrators.

The pressure in a closed gas tank will also continue to rise under a constant flow into the tank and will also be an integrating process.

Boiler Steam Drum Level



This is an interesting one.

If the feed water flow suddenly increased, you would expect the level to start going up.

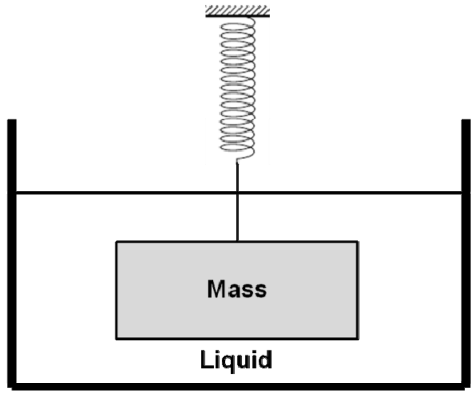
Instead, the level first goes down, then starts going up.

The “Liquid” in the bottom of the drum is actually composed of both water and steam bubbles existing in equilibrium at the saturation temperature.

If feed water flow is increased, the liquid temperature will initially drop and many of the bubbles will condense, with the result that the “liquid” becomes denser and the level drops (called shrink). The opposite would happen if the feed water flow decreased (called swell).

Eventually, the temperature will increase to the saturation point, and the level will rise as the bubbles once again begin to form.

Second Order



$$\ddot{y}(t) + 2\boxed{\zeta}\omega_n \dot{y}(t) + \omega_n^2 y(t) = \omega_n^2 u(t)$$

$$\frac{\omega_n^2}{s^2 + 2\boxed{\zeta}\omega_n s + \omega_n^2}$$

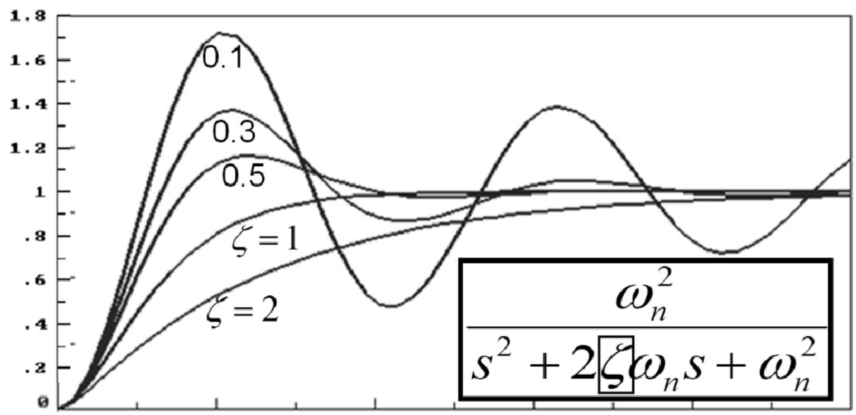
A second order system is one which is described by an equation that includes a second derivative. See the “Y double dot” term in the upper time based differential equation and the s squared term in the transfer function. (Remember from the table of transforms that s squared represents a second derivative.)

In this example, a quick change in the position of the top end of the spring (input to the system) can give different responses depending on how much energy is dissipated by the viscosity of the liquid. If there was no liquid, the response would be very oscillatory. With a very thin liquid, the response would be oscillatory, but the oscillations would dampen out very quickly. If the liquid was very viscous, the mass would slowly move to its new position with no overshoot at all.

The amount of oscillation or lack thereof depends on how quickly energy is dissipated, and is accounted for by the DAMPING FACTOR, ZETA.

(My American Heritage dictionary shows the preferred pronunciation of “zeta” is with the “E” pronounced like a long “A” and the accent on the first syllable.)

Second Order

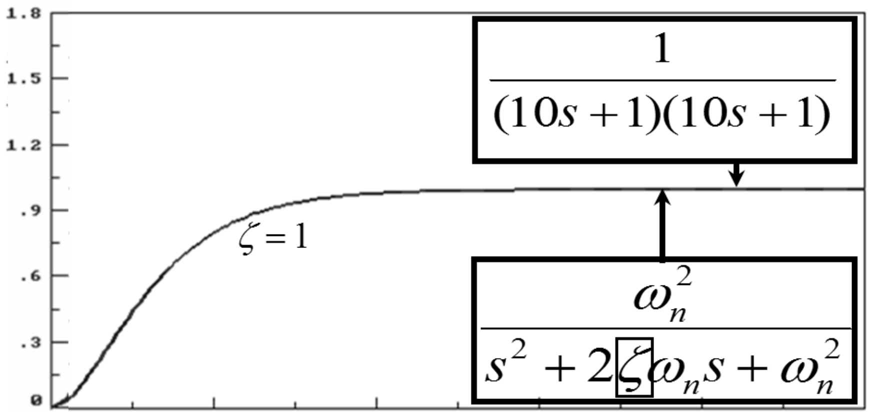


Here is the response of the mass spring system for various values of zeta.

A system with a damping factor of 1.0 is referred to as critically damped. The response is the fastest possible without overshooting.

Larger values than 1 are referred to as over damped, and smaller values are referred to as under damped.

Second Order
Critically Damped

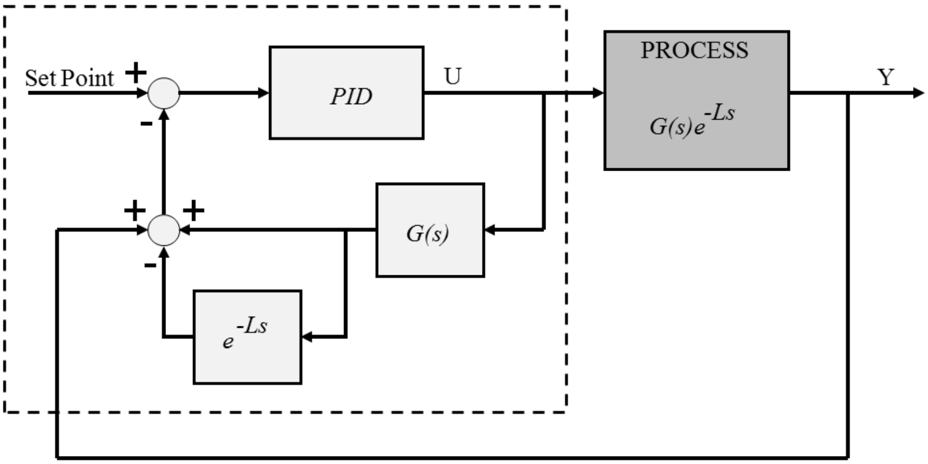


An interesting fact is that a critically damped system (zeta equal to 1.0) behaves exactly the same as two equal time constants in series.

Note that there is no significance to the time constants of 10 except that they are equal to each other.

Smith Predictor

Improves control in systems with large dead time



We can use this block diagram of a “Smith Predictor” to practice reading block diagrams with transfer functions.

Here we have a PID controller controlling a process that has dynamics, G(s) and dead time, e^{-Ls}.

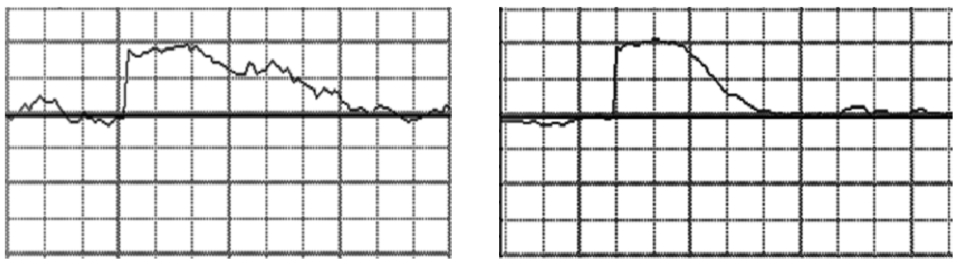
The dead time makes the process difficult to control, since if the controller output changes it takes some time before the controller input knows what the process did.

To help speed up control, we add a block into the controller with the same dynamics as the process (or at least as close as we can come) so that the controller will know right away approximately what the process is doing. Eventually the actual process response will get back to the controller. To avoid a double response, we also send the approximate process response from the G(s) block in the controller through a dead time equal to that of the process and subtract out the simulated process response at exactly the same time as the actual response reaches the controller. Since it is impossible for the simulated response to exactly match the real response, now the simulated response has been cancelled out, all that is left is the difference between the simulated and actual response to trim the controller’s output to put the process exactly where it needs to be.

Load Disturbance in System with Dead Time

PID

PID with Smith Predictor



This is a comparison of the process response to a load upset in a system with dead time, both with and without the Smith Predictor.

Chapter 3

PID Control and Controller Tuning Methods

PID Controller

Control Modes

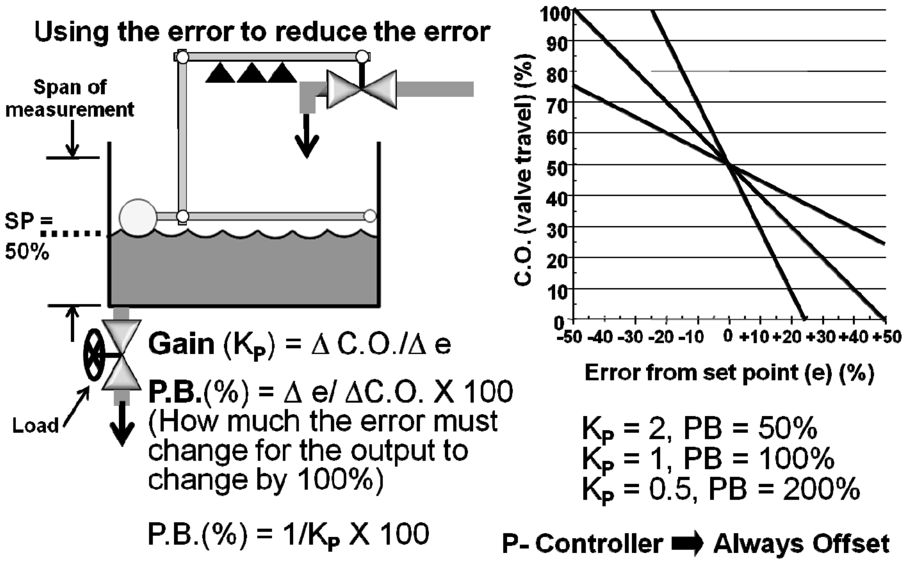
Proportional
Integral
Derivative

Tuning Methods

Ziegler-Nichols
Lambda
Proprietary (ExperTune)

Here we see the definitions of what “P”, “I” and “D” stand for. We will discuss in detail each of these control modes on the following pages, followed by a brief discussion of three tuning methods.

Proportional



The basic control mode is the “Proportional” mode, which “uses the error to reduce the error.” To help in understanding this control mode we use a diagram of a mechanical proportional controller which is a float that operates a valve to maintain the level at the desired set point of a level of 50%.

Lets first see what would happen if the fulcrum point was set at the left most position (for now ignore the other two fulcrums). The operation of the controller is graphically represented by the line with the steepest slope. The horizontal axis is the percent error from the set point of 50% full. The vertical axis is valve position. (If this were a pneumatic or electronic controller the vertical axis would be the controller output signal, but because this is a mechanical controller the controller output ***is*** the valve position.)

The definition of gain of any device is “change in output divided by the corresponding change in input.” For a proportional controller the output is the controller output (abbreviated “C.O.”). The input is the error between set point and measurement. Throughout this chapter “e” is the error between the set point (SP) and the measurement of the process variable. The symbol for gain is usually “K.” Here we are talking about the “proportional” gain, so the symbol is K with a subscript “p” for “proportional.” For the fulcrum in the left most position and the resulting graph of error versus valve position with the steepest slope when the error changes from minus 25% to plus 25% (a total of 50%) the valve position changes by 100%. So the proportional gain is 100 divided by 50 or ***2.*** With the fulcrum in this far left position we will get the largest change in valve position for a given change in float position, so, of the three fulcrum positions, this one will give the highest gain (or the greatest sensitivity).

If we move the fulcrum to the center position, the valve travel does not change as much for the same amount of error, and the action of the controller is represented by the center line on the graph. In this case a change of error from minus 50% to plus 50% (a total of 100%) causes the valve travel to change by 100%. So the gain is now 100 divided by 100 or ***1***.

Moving the fulcrum to the far right position gives us the least sensitivity. The action of the controller is represented by the graph with the smallest slope. In this case a change of error from minus 50% to plus 50% (a total of 100%) causes the valve travel to change by 50%. So the gain is now 50 divided by 100 or ***0.5***.

Sometimes, instead of talking about proportional gain, people talk about the "Proportional Band" abbreviated in the figure "P.B." Mathematically, the proportional band is the reciprocal of the proportional gain times 100 and expressed as a percent.

Once there is a change in load (in this example the flow out of the tank) there will always be some offset (the difference between the set point and the actual measurement, in this example the measurement is the tank level). In order for the valve to open farther so that the inflow will match the new higher outflow, the float will have to be lower than it was originally.

This is a characteristic of all controllers that only have the proportional mode. **The proportional mode uses the <u>error to reduce the error</u> so it is necessary for there to be an error (in control terms called "offset") for the error reduction to take place.**

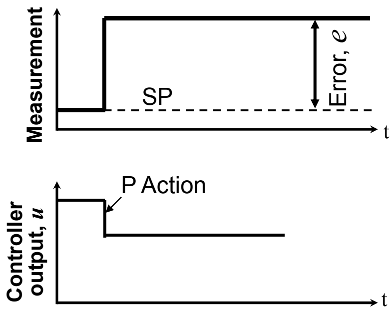
Proportional

Using the error to reduce the error

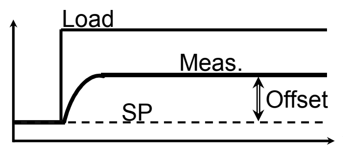
$$u(t) = K_P e(t)$$

$$\boxed{K_P}$$

u = **Controller output**
Kp = **Proportional gain**
e = **Error (difference between set point and measurement)**



Open loop Response



Closed loop Response

P- Controller ➡ Always Offset

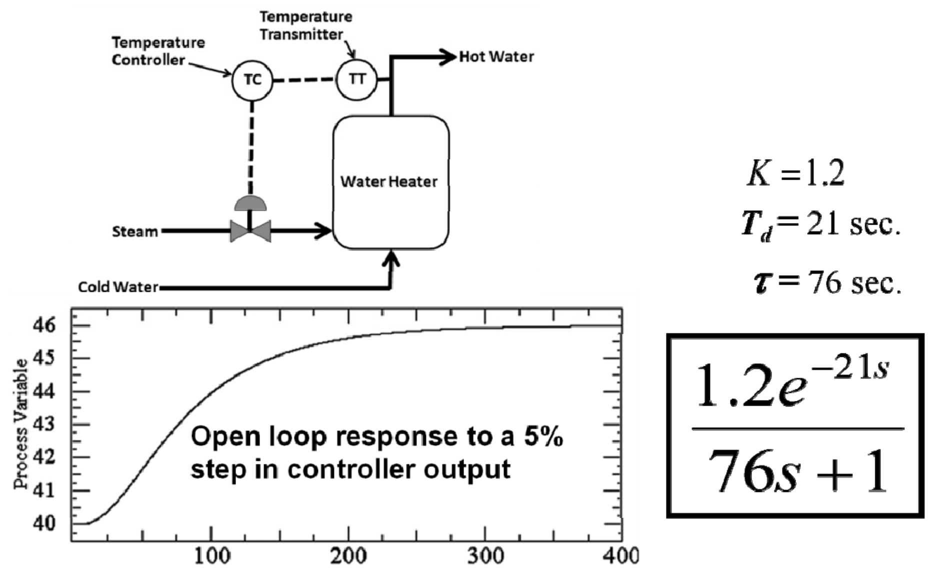
The equation at the top mathematically describes the proportional control action. “u” (the controller output) which is a function of time equals the proportional gain times the error between measurement and set point which is also a function of time. K sub p in the box is how we would draw the block diagram of a proportional controller, and K sub p is the transfer function of a proportional controller. For our purposes in understanding control modes, it is not so important to remember the equations or transfer functions, but since we spent a lot of time understanding what a function is and what a transfer function is, these are included to make the presentation complete.

The two graphs on the left show the relationship between the measurement and the controller output from a proportional controller. As soon as an error (e) occurs between measurement and set point, the controller output changes to exactly mirror the error, except that the magnitude of the controller output change is dependent on the proportional gain of the controller. In this case, the proportional gain is less than one, since the change in output is less than the change in error. The direction of the controller output change is chosen so as to be in the direction that will tend to correct the error. The graphs on the left are showing the “open loop” interaction between error and controller output, that is we are seeing how the controller responds to an error, but the output is not connected to the process. Shortly we will see screen shots from the Expertune Loop Simulator that show us exactly what happens when the loop is closed and the controller is controlling the process.

The graph on the right shows how a first order process would respond to a step change in load while being controlled by a proportional controller. The important point here is that with proportional control we are using the error to reduce the error, so there will always be some residual error, which we call offset.

Proportional

Using the error to reduce the error



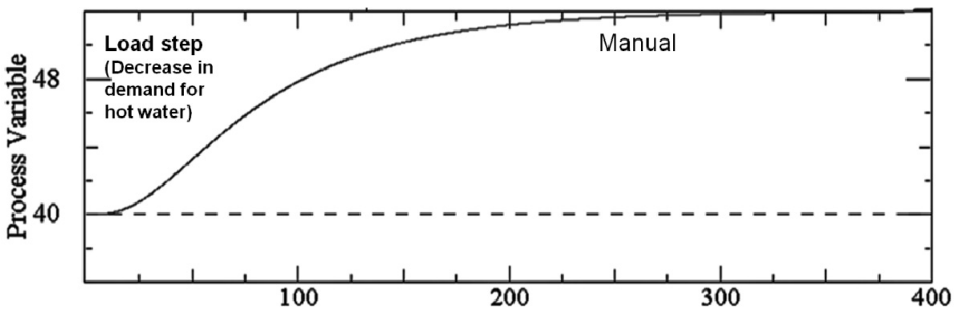
Process graphs generated with the ExperTune Loop Simulator, www.expertune.com

We will use this water heater to study the behavior of the various control modes. Although the water heater consists of several dynamic sub systems (control valve, the heating vessel, the temperature element, and the temperature transmitter) when we perform a step test with the controller in manual we get a response that for all practical purposes can be treated as a first order response with dead time. A careful graphical analysis of the response (we did this in the chapter on process dynamics) yielded the transfer function shown in the box at the right of the figure.

The transfer function tells us that the process gain is 1.2, the dead time is 21 seconds and the time constant is 76 seconds.

Proportional

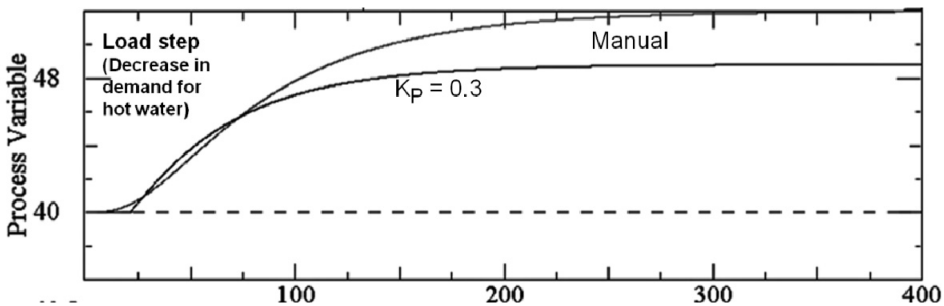
Using the error to reduce the error



To get a reference point for evaluating the performance of the controller, we have left the controller in manual and then introduced a step change in load. We did this by suddenly decreasing the demand for hot water. Since the steam flow does not change, the measured temperature increases to a new value following the approximately first order plus dead time response shown in the graph.

Proportional

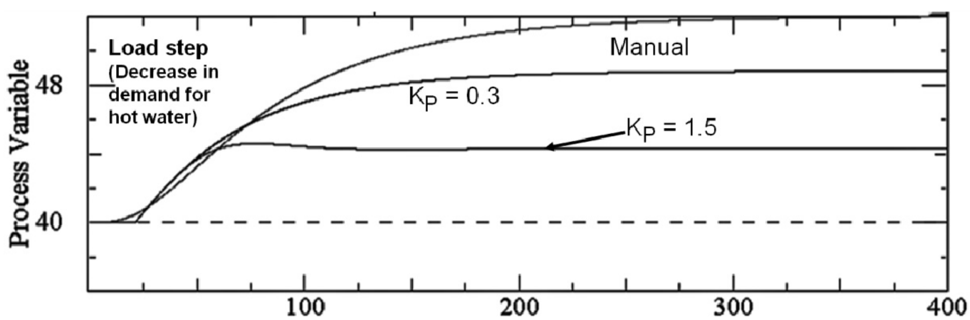
Using the error to reduce the error



Now we put the controller into automatic mode with a small amount of proportional gain. The controller reduces the error some, but we are left with a large residual error, or “Offset.”

Proportional

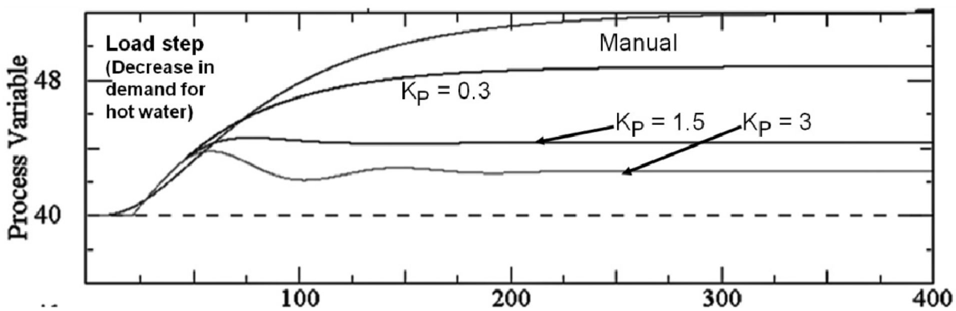
Using the error to reduce the error



Now we have increased the proportional gain to 1.5. This higher gain gives a smaller offset.

Proportional

Using the error to reduce the error

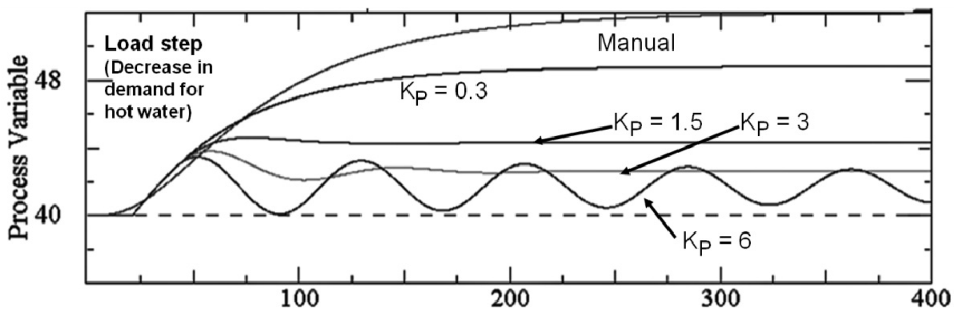


Further increasing the proportional gain to 3 gives an even smaller error and thus better control. Note that there is a small oscillatory transient at first.

At this point it is tempting to make the assumption that the higher the gain, the better the control, and that it might be possible to decrease the offset to a very small value by setting a very large proportional gain. Lets try increasing the gain some more and see what happens.

Proportional

Using the error to reduce the error



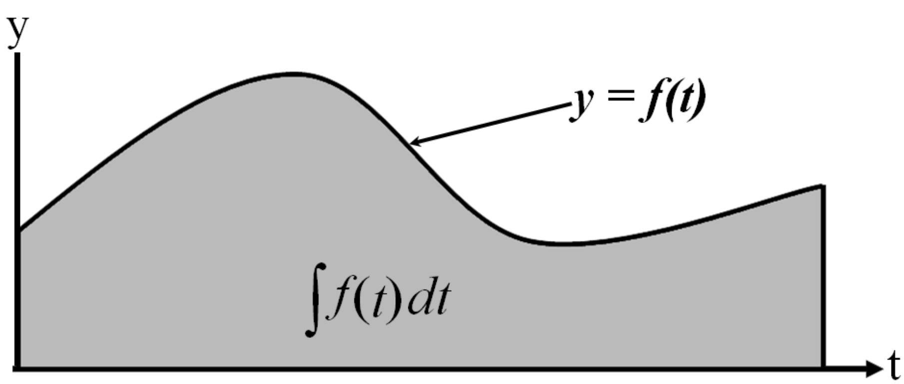
OOPS!!!

At some point, with increasing proportional gain, the system becomes unstable.

If we cannot tolerate some offset, we must look for some way of supplementing the proportional control mode.

Integral (Reset)

The *integral* of a function is the area under its graph



In order to remove the offset of the Proportional control mode, we introduce the Integral (sometimes called Reset) mode.

This page and the next give a very brief overview of what integral means.

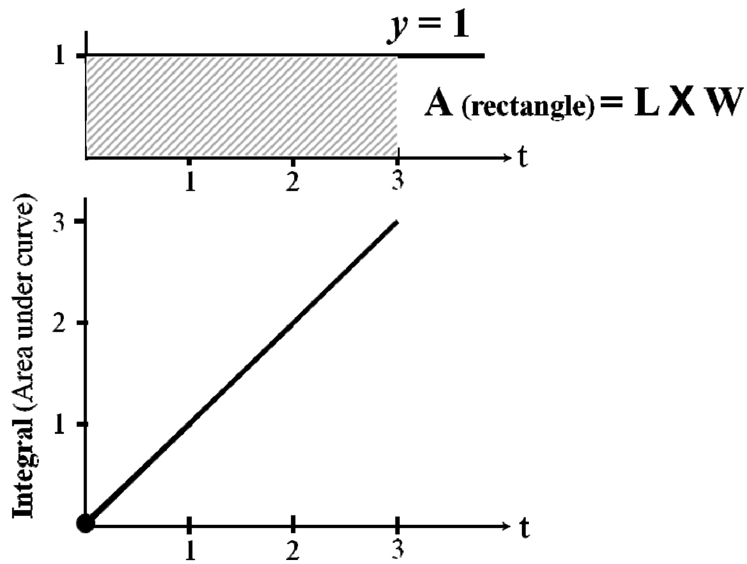
In calculus, the “integral” of a function can be interpreted to mean “the area under the graph of that function.”

Here is shown an arbitrary complex function of time and its graph. If we knew the exact function that gives this graph, we could determine the area under the curve, but it often takes methods that we spent a whole year in calculus learning.

The expression $\int f(t)\,dt$ would be read “the integral of f of t with respect to t.”

Integral (Reset)

The *integral* of a function is the area under its graph



Fortunately, a very simple function, and one that is easy to calculate the area under the graph without any advanced techniques is all we need to make sense out of how the Integral control mode works. (How the integral control mode works is explained on the next page.)

Here we have a time function who's value is always 1.0. Since the function's value remains constant, the area under its graph is always a rectangle, and the area of a rectangle is very easy to calculate without using any advanced techniques.

Imagine starting at time equal to zero, and then watching what happens as time progresses.

At exactly time = zero, the length of the rectangle is zero and its width is 1. The area is zero times one or zero. (Easy!) After one second has passed, (time is now equal to 1 second) the length of the rectangle is 1, and the width is 1, so the area is 1 times 1, or 1, and the lower graph shows how the integral (area under the curve in the upper graph) has changed during the first second. As time continues to progress and the area of the rectangle increases, the lower graph continues to track what the rectangle's area is at any moment. Since the area under the curve is increasing in a linear fashion with time, the graph of the integral is a ramp, also increasing in a linear manner with time.

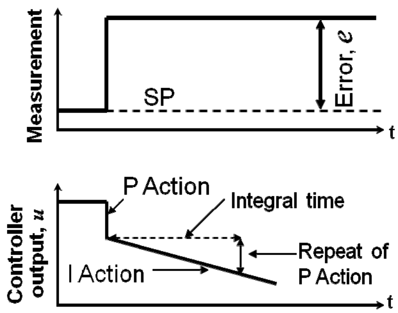
Integral (Reset)

Using the *integral* of the error to eliminate offset

$$u(t) = K_P\left(e(t) + \frac{1}{T_I}\int e(t)dt\right)$$

$$\boxed{K_P\left(1+\frac{1}{T_I S}\right)}$$

T_I **= Integral Time**



Open loop Response

Integral time (T_I): The time it takes the integral action to repeat the correction produced by the proportional action.

Minutes per repeat (or seconds)

(Shorter integral time results in more aggressive correction.)

Integral gain (K_I) repeats per minute

$K_I = 1/T_I$

Adding the "integral" control mode to the proportional mode makes it possible to remove the offset left by the proportional mode. Another term that is used for integral is "reset."

The graphical interpretation of the mathematical "integral" function is "the area under a curve." To understand what this means, consider the upper graph that graphs the error between measurement and set point. At the exact time that the measurement steps from set point to a new higher value, the "integral" of the error (the area under the graph) is zero. You of course recall that the area of a rectangle is the length times the width. The width is whatever the magnitude of the error is, for discussion let's say it is 1.0. The length at the point where the measurement has just increased is zero, so 1 times zero is zero.

One second later, the area under the curve is 1 (the error) times 1 second, or 1. After 2 seconds the area is 1 times 2. After 3 seconds the area is 1 times 3 and so on. Therefore the integral of the error in this example starts at zero and increases at a constant (or linear) rate for as long as the error is present. This controller has been configured so that both the proportional and integral actions are downward instead of upward, because that is the direction that will eliminate the error.

At the moment the error first occurs, there is an immediate proportional action in the controller output. Then the controller output starts ramping down (integral action) in proportion to the area under the graph (error times the constantly increasing time). The parameter that is set into the controller to tell it how strongly the integral action is to act on the controller output is called the “integral time,” T sub I. The integral time is the time it takes the integral action to repeat the correction produced by the proportional action. A short integral time means that the controller ramps its output very quickly to eliminate the error, and a long integral time means that the output ramps very slowly to eliminate the error (or offset). The units are minutes (or seconds depending on the controller manufacturer) per repeat.

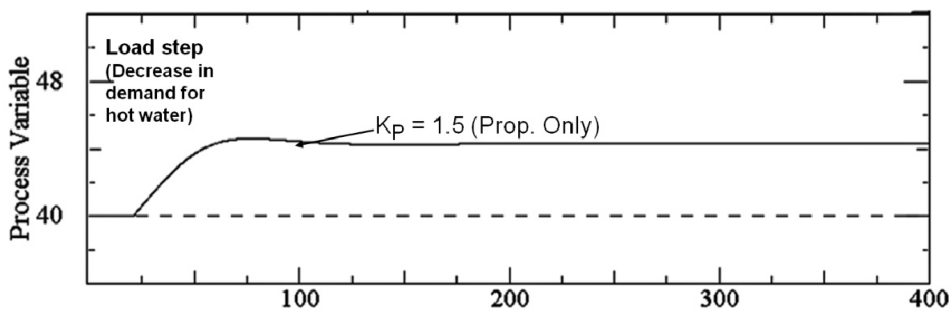
Also some controller manufacturers use “Integral Gain” which is simply the reciprocal if integral time. In that case, the units are repeats per minute (or second).

The graphs on the left are showing the “open loop” interaction between error and controller output, that is we are seeing how the controller responds to an error, but the output is not connected to the process. On the next page we will see graphs that show us what happens when the loop is closed and the controller is controlling the process.

As we did in the explanation of the proportional control mode, the equation of the time function of the controller output and the transfer function of a proportional plus integral (P+I) controller are shown for completeness of the presentation.

Integral (Reset)

Using the *integral* of the error to eliminate offset

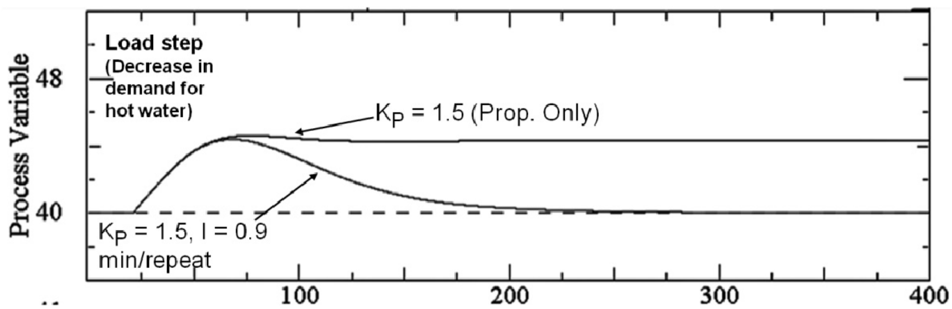


This is the same graph we saw earlier when the process was being controlled by a proportional only controller with a proportional gain of 1.5.

When we were looking at the effect of various values of proportional gain, we had gotten better (but slightly oscillatory) control with a gain of 3, but because I know that integral action is destabilizing, and would have resulted in an oscillatory response, I chose to use the slightly lower proportional gain for this example.

Integral (Reset)

Using the *integral* of the error to eliminate offset

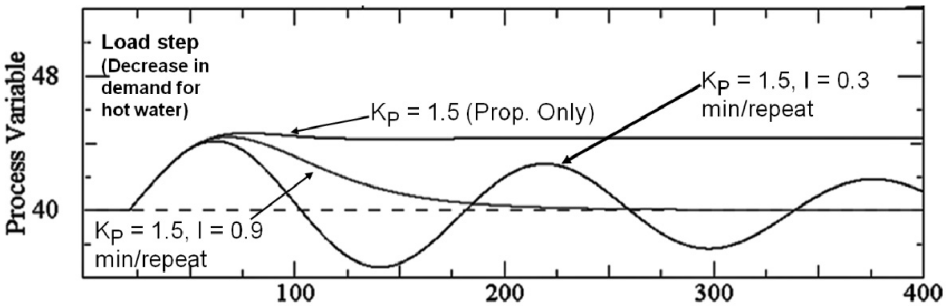


Here we have added some integral action. Initially the proportional action eliminates part of the error, then the integral, or reset, action continues to drive the control valve until all of the offset has been removed. In closed loop, once all of the error has been eliminated, the proportional action settles out at the new value required to hold the error at zero, and since there is no error, the integral of the error is zero, thus there is no further integral action.

The next question might be: can we decrease the integral time to make the error be eliminated more quickly?

Integral (Reset)

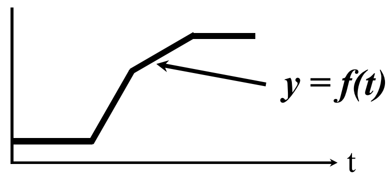
Using the *integral* of the error to eliminate offset

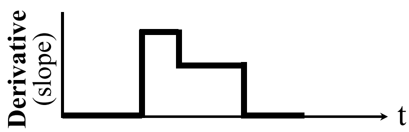


As with proportional gain, some integral is good, but too fast an action destabilizes the process.

Derivative (rate)

The *derivative* of a function is the slope (or rate of change) of its graph.





Before discussing the Derivative (sometimes called rate) control mode, here is a brief review of the meaning of the derivative.

In calculus, the derivative of a function can be interpreted as the instantaneous slope of that function’s graph at any point. We spent the better part of a year learning how to do this for all sorts of functions.

Fortunately, for purposes of discussing the derivative control mode all we need to do is review the behavior of the derivative of straight lines.

In the example, we have the graph of a function of time whose shape consists entirely of straight lines with different slopes.

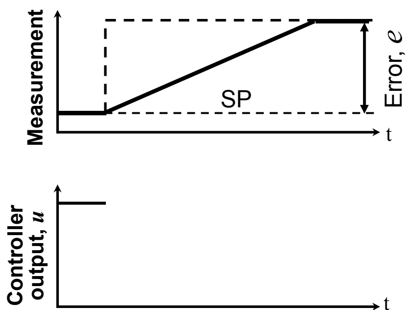
Starting at time zero and continuing for a while, the functions value is zero. Its slope is also zero and thus its derivative is zero, as shown in the lower graph. Suddenly the value of the function begins increasing at a steady rate. Its derivative (slope) instantly becomes a finite (and constant value) again portrayed in the lower graph. Next the function continues to increase, but at a lesser rate (its slope still has a finite and constant value, but a smaller one). Again this smaller, but constant rate of change (slope or derivative) is graphed in the lower graph. Finally the time function stops growing, and levels off at a constant value. At this point there is no more change in the function’s value (its rate of change or slope or derivative becomes zero) and is graphed on the lower graph of derivative as a derivative of zero.

Derivative (rate)

Using the *derivative* of the error to anticipate future error

$$u(t) = K_P\left(e(t) + \frac{1}{T_I}\int e(t)dt + T_D\frac{de(t)}{dt}\right) \qquad \boxed{K_P\left(1+\frac{1}{T_I S}+T_D S\right)}$$

***T*D = Derivative Time**



Open loop Response

The Derivative (sometimes called rate) control mode uses the derivative (rate of change, or slope) of the error graph to anticipate future error.

When we were discussing the proportional control mode and the integral control mode we discussed their action based on the assumption that we were controlling a fairly fast process. The discussion was made much more simple (without loss of meaning) by assuming that upon a process disturbance the measurement made a step increase (like in the dashed line in the upper graph in the figure).

Some processes, such as the water heater we have been using as an example, respond very slowly to process upsets. In such a case, the ramp in the figure is a simplified but more realistic depiction of what happens.

In this example, the process upset could have been a nearly instantaneous decrease in the demand for hot water from our water heater. At the point where the ramp just starts, the damage has already been done and the process is heading toward a large error. The problem here is that because the process responds slowly, the controller does not immediately see the large error that is on its way. The controller only sees a small error at first.

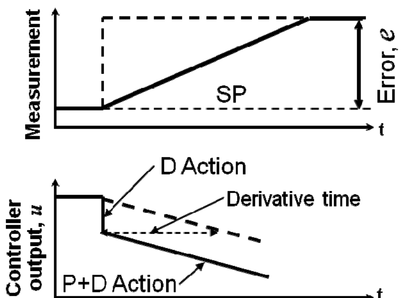
Derivative (rate)

Using the *derivative* of the error to anticipate future error

$$u(t) = K_P\left(e(t) + \frac{1}{T_I}\int e(t)dt + T_D\frac{de(t)}{dt}\right)$$

$$\boxed{K_P\left(1+\frac{1}{T_I S}+T_D S\right)}$$

T_D = Derivative Time



Open loop Response
Only P and D action is shown

Derivative time (T_D): The time it would have taken the proportional action to produce the correction that was immediately produced by the derivative action.

(Longer derivative time results in more aggressive correction.)

Minutes (or seconds)

Derivative gain (K_D)

$K_D = 1/T_D$

In the upper graph, the error starts out being very small, and with proportional only control, the controller's output would only be a small correction at first represented by the sloping dashed line. In a slow process, the disturbance was very likely a large one, but because the process responds slowly, we do not see the large disturbance right away. At the point where the measurement begins to deviate from the set point the slope of the measurement (its derivative) makes a sudden jump from zero to a value equal to the slope of the measurement's graph. This provides an instantaneous jump in the controller output, in anticipation of the large error that we don't see yet, but is coming. The proportional correction gets added to the derivative correction, so that after the initial "boost" of the derivative, the controller output continues with a correction that is proportional to the error. (To avoid unnecessary complication to the explanation, I have not attempted to include the integral action in the above discussion.)

The parameter that is set into the controller to tell it how strongly the derivative action is to act on the controller output is called the "derivative time," T sub D. The derivative time is the time it would have taken the proportional action to produce the correction that was immediately produced by the derivative action. A short derivative time means that the controller adds only a small derivative output to anticipate a future error. A long derivative time means that the controller adds a large derivative output to anticipate a future error. The units are minutes (or seconds depending on the controller manufacturer).

Also some controller manufacturers use “Derivative Gain” which is simply the reciprocal of derivative time. In that case, the units are 1 over minutes (or seconds).

The graphs on the left are showing the “open loop” interaction between error and controller output, that is we are seeing how the controller responds to an error, but the output is not connected to the process. Shortly we will see graphs that show us what happens when the loop is closed and the controller is controlling the process.

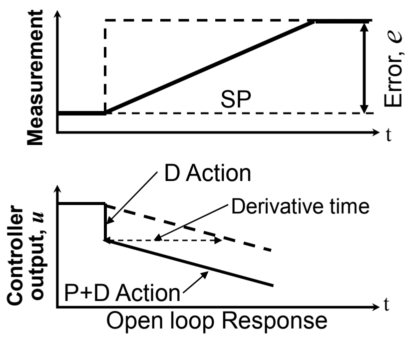
As we did in the previous explanations of the control modes, the equation of the time function of the controller output and the transfer function of a proportional plus integral plus derivative (P+I+D) controller are shown for completeness of the presentation.

Derivative (rate)

Using the *derivative* of the error to anticipate future error

$$u(t) = K_{\mathrm{P}}\left(e(t) + \frac{1}{T_{\mathrm{I}}}\int e(t)dt + T_{\mathrm{D}}\frac{de(t)}{dt} \right) \qquad \boxed{K_P\left(1+\frac{1}{T_{\mathrm{I}}S}+T_{\mathrm{D}}S\right)}$$

T_{D} = Derivative Time



Only P and D action is shown

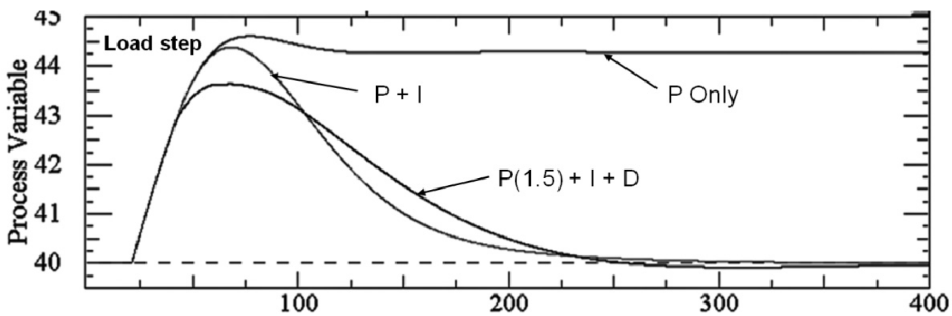
- **Sometimes derivative is taken from measurement (dy/dt) instead of error (de/dt)**
- **If noise in error signal -> filtering before derivative**

Some controllers take the derivative from the measurement rather than the error. This prevents a large derivative correction if the set point is manually changed suddenly.

Noise spikes in a noisy measurement can cause undesired large outputs form the derivative mode. Derivative must be used with caution when the measurement is noisy. Filtering the signal before it goes to the derivative function can help.

Derivative (rate)

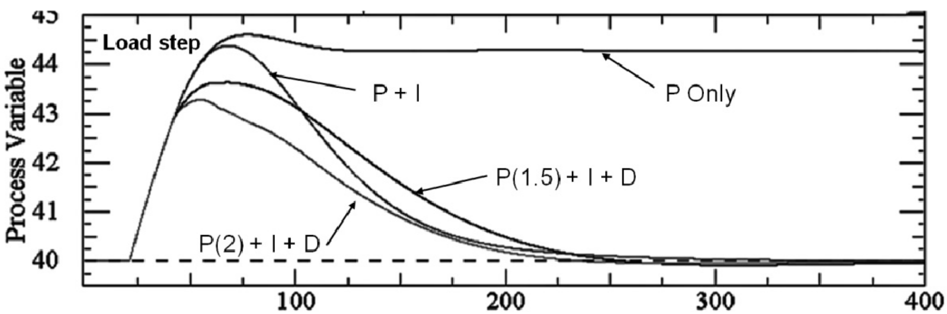
Using the *derivative* of the error to anticipate future error



In this figure, the upper two traces show what could be accomplished with proportional only and with proportional plus integral (P+I). Here we have added derivative (P(1.5) + I + D) to the earlier P+I to further reduce the maximum error.

Derivative (rate)

Using the *derivative* of the error to anticipate future error



The derivative mode, unlike the integral mode which tends to destabilize control, adds stability. Because of this, it is possible to increase the proportional gain from 1.5 to 2. If we had increased the gain to 2 with just integral we would have gotten a response with too much oscillation in it, but with the stabilizing effect of the derivative, we are able to get a response that is better that what we would have gotten with just P+I or with P+I+D using the proportional gain that would have been optimum had we not had added derivative.

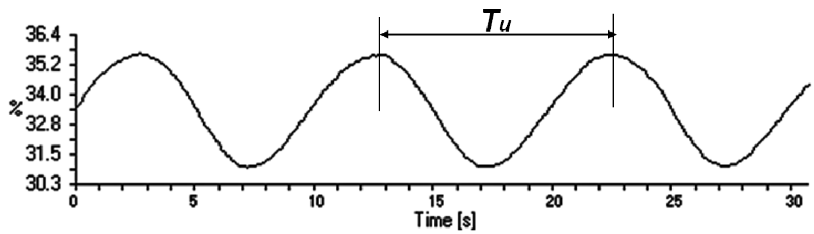
Ziegler-Nichols Tuning (closed loop)

1942

Fast responding aggressive tuning

"Ultimate" Method

- **Set controller to P control**
- **Increase gain until system is oscillating with *constant amplitude. Record K_u (ultimate gain), the controller gain required to achieve a sustained oscillation, and T_u (ultimate period)***



The first formalized method for tuning PID controllers was introduced in 1942 in a paper by Ziegler and Nichols. This method is still popular with many people today.

Ziegler-Nichols tuning is a very aggressive tuning method that results in a fast elimination of the disturbance, but the response is oscillatory in nature as we will see later.

They proposed two methods. The preferred method was a method where data was taken while the process was running in closed loop, called the "Ultimate" method.

With the controller in proportional only mode, the controller's proportional gain is increased in small steps until a sustained oscillation of constant amplitude is achieved. Then the ultimate gain (the controller proportional gain need to obtain the sustained oscillation) and the period of the oscillation are recorded.

Ziegler-Nichols Tuning (closed loop)

Fast responding aggressive tuning

- **Ultimate Gain = K_U**
- **Ultimate Period = T_U**

	K_P	T_I	T_D
P	$0.5\ K_U$		
PI	$0.45\ K_U$	$0.83T_U$	
PID	$0.6\ K_U$	$0.5T_U$	$0.125T_U$

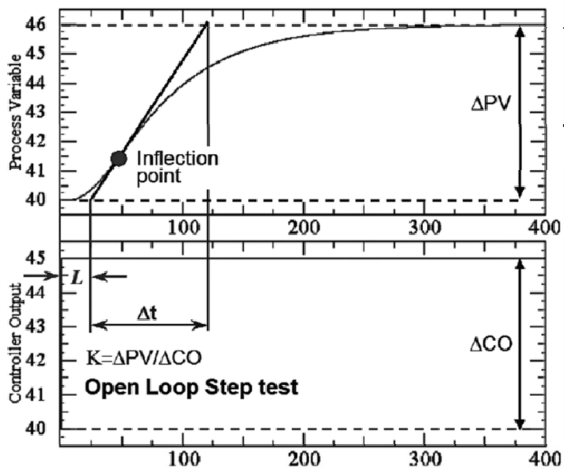
- **More reliable parameters by ultimate method, but most processes will not tolerate a sustained oscillation.**

Based on the ultimate gain and ultimate period, one uses the formulas above to calculate the tuning parameters.

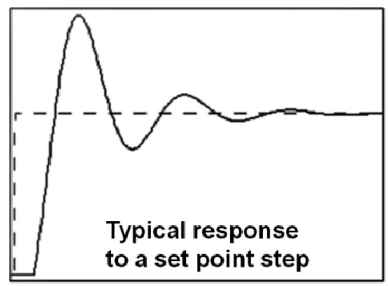
It is usually not practical to use the ultimate method because most processes cannot tolerate being put into a mode with sustained oscillations.

Ziegler-Nichols Tuning (open loop)

Fast responding aggressive tuning



	K_P	T_I	T_D
P	Δt / KL		
PI	0.9 Δt / (KL)	3.3L	
PID	1.2 Δt / (KL)	2L	L/2



A more practical approach to obtaining Ziegler-Nichols tuning parameters is by conducting a step test of the process with the controller in manual (open loop).

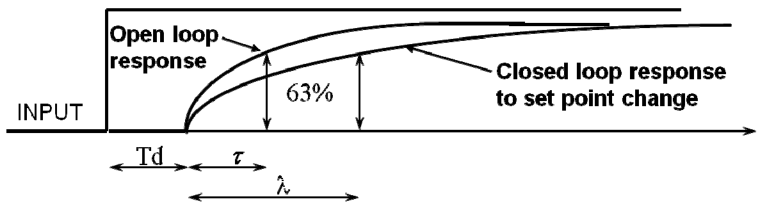
From the step test, the process gain, K, can be determined. The other parameters required by the formulas, delta t and L are obtained by drawing a tangent line to the response curve that is tangent to the curve at the inflection point (the point where the slope is the greatest)

L, which is the time from the step in controller output to where the tangent line intersects the process variable before the step. Delta t is the time between the point where the tangent intersects the process variable value before the step, and where it intersects the value of the process variable after the new steady state value is reached.

Applying Ziegler-Nichols tuning formulas usually results in a response that is oscillatory.

Lambda (set point) Tuning

A tuning method that specifies the *closed loop time constant* for a step change in set point. Slow, non oscillatory.



- **Obtain open loop step test results: Gain (K), Time Constant (τ), Dead time (Td)**

$$T_I = \tau \qquad K_P = \frac{\tau}{K(\lambda + T_d)}$$

The "closed loop time constant," lambda (λ), is usually 2-3 X the open loop process time constant (τ). (Per EnTech λ ≥ 3Td, or λ ≥ 3τ, whichever is largest.)

- **Higher lambda values give slower responses**

A tuning method that is popular in the pulp and paper industry is "Lambda" tuning. With Lambda tuning, the closed loop response to a step change in set point approximates a first order response with a time constant that can be specified by the user.

The integral time, Ti is always set to be equal to the process' open loop time constant. The closed loop time constant can be set to any desired value by simply putting in the appropriate proportional gain, Kp as determined by the above formula.

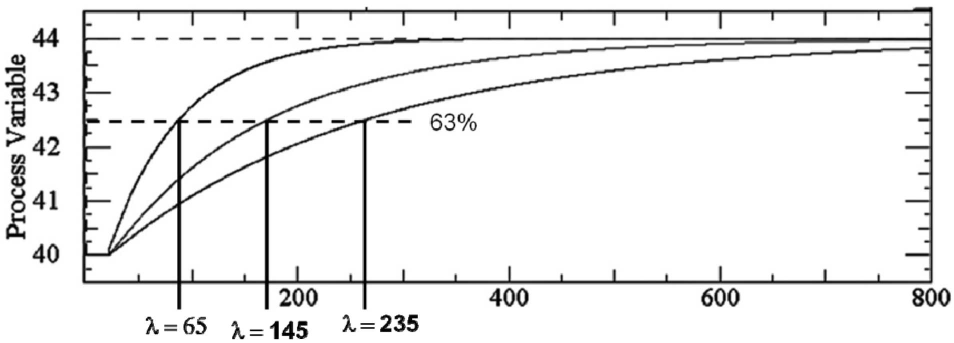
One reason for the popularity of Lambda tuning in the paper industry is that the response is non oscillatory. Oscillations for a surprisingly long way upstream of the paper machine are likely to be imprinted into the thickness of the final product.

When there is coupling between control loops so that they tend to fight each other or "dance together" the way of decoupling them is to tune the fastest loop for quick response, the next fastest for five times slower response, the next slower for five times slower than the previous and so on. With Lambda tuning it is very easy to specify the speed of response of a loop.

Lambda tuning usually results in slow, sluggish responses to upset.

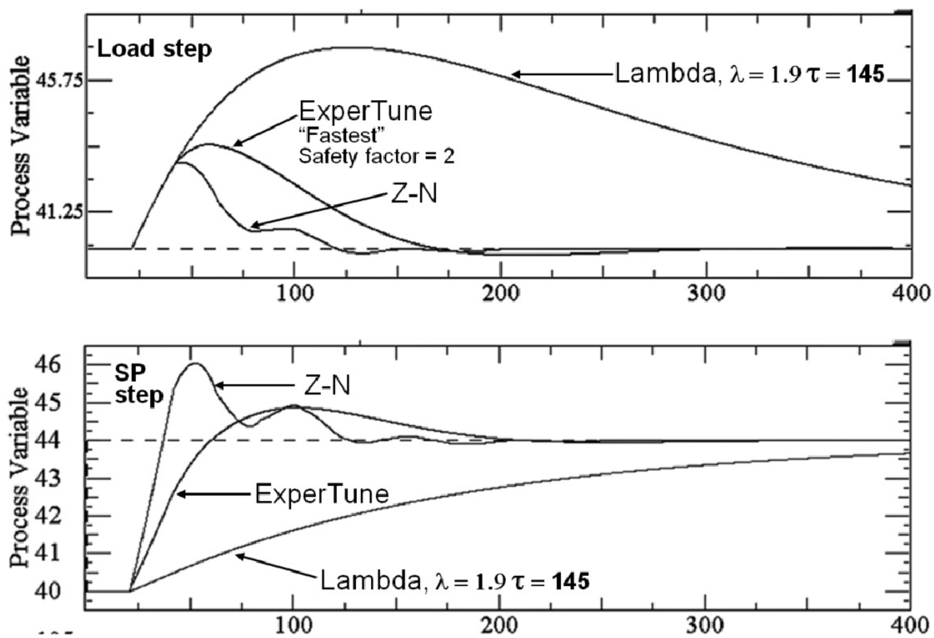
Lambda (set point) Tuning

Closed loop response of the water heater example to a set point step with various values of λ



This is the closed loop response of our water heater example to a step change in set point with Lambda set at three different values.

Tuning Comparison



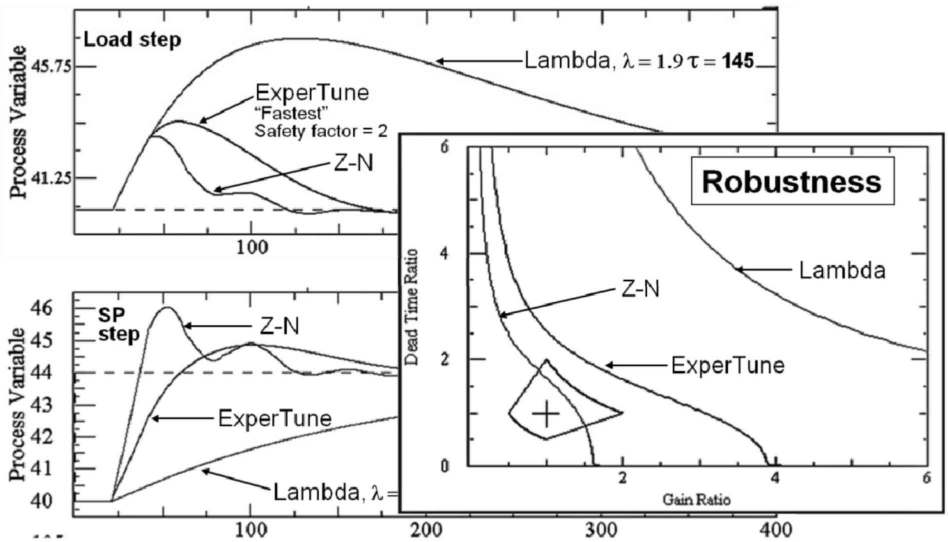
Here we compare the results of both step changes in set point and step changes in load for our water heater example when the loop is tuned by the two methods we have already discussed and also by the ExperTune loop tuning software which does a frequency domain analysis of an open loop step in controller output and selects tuning parameters that find a compromise between speed, oscillatory behavior and robustness.

The Ziegler-Nichols tuning removes the error very quickly, but is oscillatory.

The Lambda tuning is completely non oscillatory and has no overshoot on the set point change, but is very sluggish.

The ExperTune tuning removes the error quite quickly, exhibits very little oscillation on the load step and only a small overshoot in the set point step.

Tuning Comparison



There is one other factor that must be considered when tuning a control loop, and that is "robustness." By robustness, we mean how insensitive is the loop to changes in the process gain and process dead time.

The superimposed graph is an analysis, done by the ExperTune program, of the robustness of the three sets of tuning parameters in our example.

The horizontal axis is "Gain Ratio" and the vertical axis is "Dead Time Ratio." There is a set of "cross hairs" indicating the point where the gain ratio and dead time ratio are both equal to 1.0. This point represents the combination of process gain and dead time that were used to tune the loop for the responses shown in the main graphs. A gain ratio of 2 means the process gain has increased to two times the gain that the loop was tuned for. A dead time ratio of 2 means that the dead time has increased to two times the dead time the loop was tuned for. The three curves in the robustness graph represent the combination of gain and dead time ratios that would result in the loop going into a sustained oscillation for each of the three tuning scenarios in the main graphs. That is, for the Z-N tuning, all combinations below and to the left of the graph would not result in a sustained oscillation, and all combinations of dead time and gain ratio above and to the right of the line would result in sustained oscillation. Therefore for good control with minimal oscillation upon an upset, it is necessary for any new combination of gain and dead time ratio to not get too close to the robustness graph.

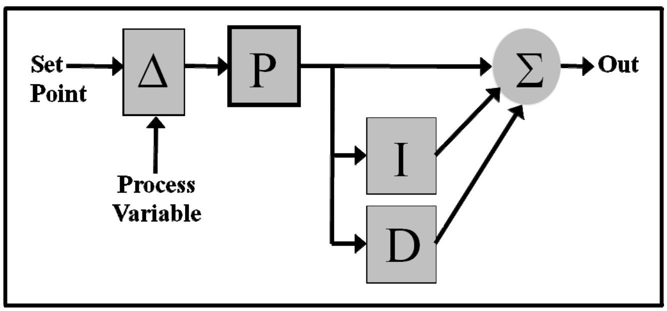
The “box” drawn around the cross hairs is a reminder that it is not unusual for the process gain and/or dead time to vary by a ratio plus or minus 2 to 1 as the process is controlled at different operating points. The ExperTune help file states: “For practical system stability keep the limit of stability line outside the ‘box’. The vertices are connected by lines that are straight on a log log plot.”

In summary, the Ziegler-Nichols tuning is very fast, but not very robust. The Lambda tuning is sluggish but extremely robust, and the parameters selected by the ExperTune software give fast response and good robustness.

There are other loop tuning software packages available. I have no business relationship with Expertune, but it is the only loop tuning system that I have firsthand experience with.

PID Summary

	Proportional	Integral	Derivative
Output proportional to	Error	Duration of Error	Speed of error
Purpose	Correct error	Remove offset	Improve speed, stability
Disadvantage	Offset	Adds instability	Sensitive to noise
Parameter	K_P	T_I	T_D
Alternate parameter	$PB = 1/K_P$	$K_I = 1/T_I$	$K_D = 1/T_D$



This is a summary of the three control modes.

ABOUT THE AUTHOR

Jon Monsen, Ph.D., P.E., is a control system engineer and a control valve technology specialist with more than 35 years of experience and is the author of the chapter on "Computerized Control Valve Sizing" in the ISA Practical Guides book on Control Valves.

The material in this book is based on Microsoft® PowerPoint presentations he has presented over the years to a broad range of engineers and technicians. The content of this book consists of slide images along with detailed explanatory notes. Topics covered are:

- Review of engineering mathematics as it applies to process dynamics including the Laplace transform
- An overview of process dynamics including a discussion of transfer functions
- An introduction to PID control and a comparison of loop tuning methods