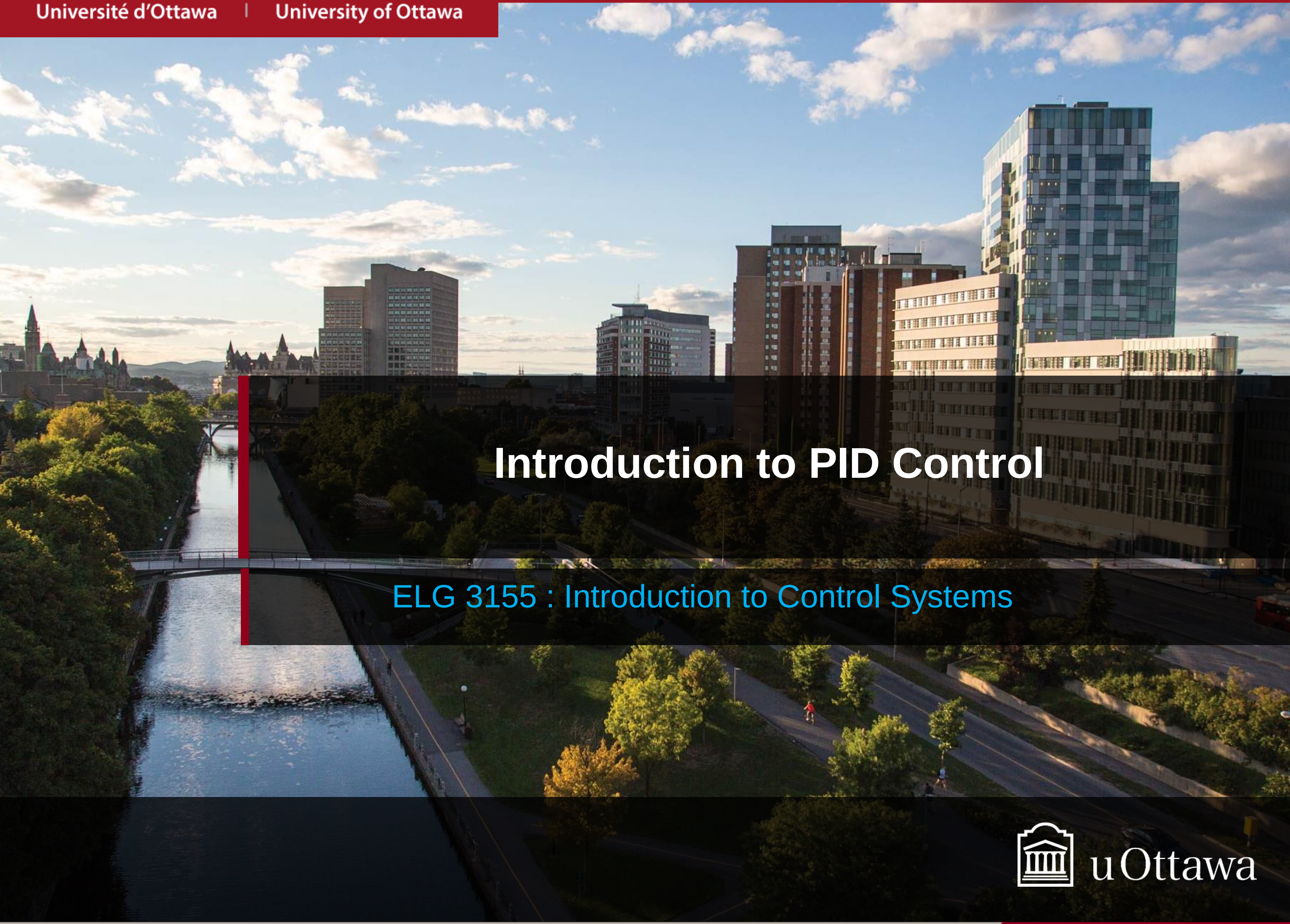


Introduction to PID Control

ELG 3155 : Introduction to Control Systems

uOttawa

Hitham Jleed

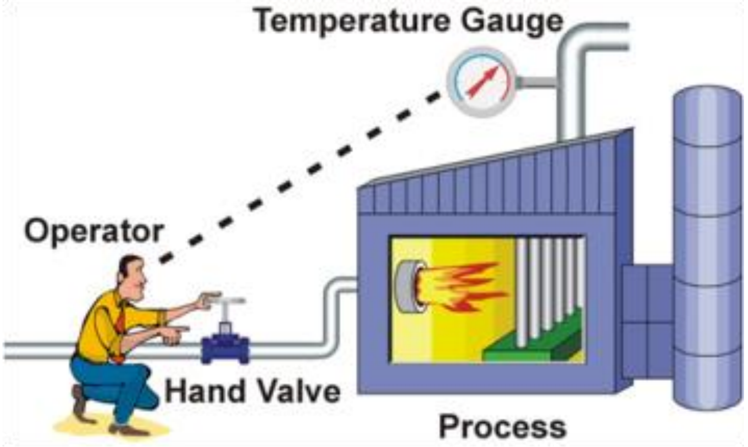
What is PID Control?

- Let’s take a step back… What is **control**?
- **Control** is just making a dynamic process behave in the way we want
- We need 3 things to do this:
  - A way to **influence** the process
  - A way to see how the process **behaves**
  - A way to **define** how we want it to behave

uOttawa

Feedback Control

- Now we have a measurement (**MV**), some value that we want it to be (**SP**), and some way to make changes to the process (**control input**)
- We can ‘**close the loop**’



The Controller as a System

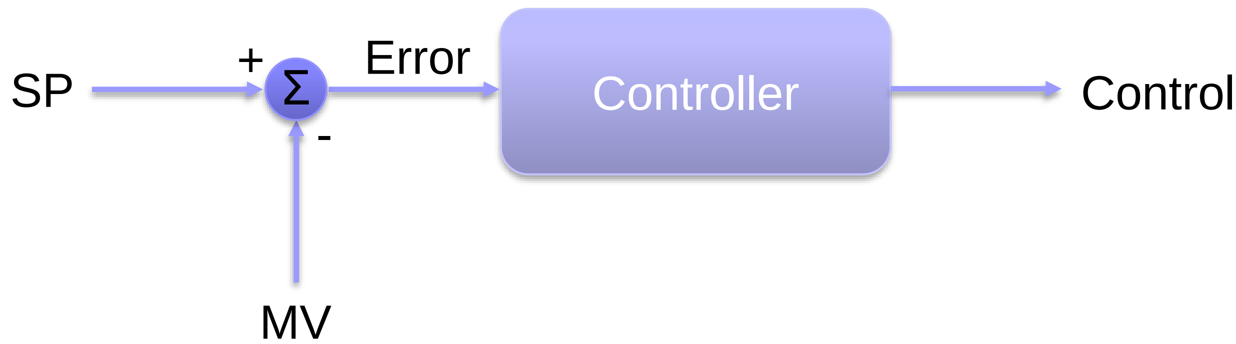
- Now we can see that any controller can be thought of as a system that takes a **setpoint** and a **measured value** as inputs, and gives a **control** signal as an output

The Controller as a System

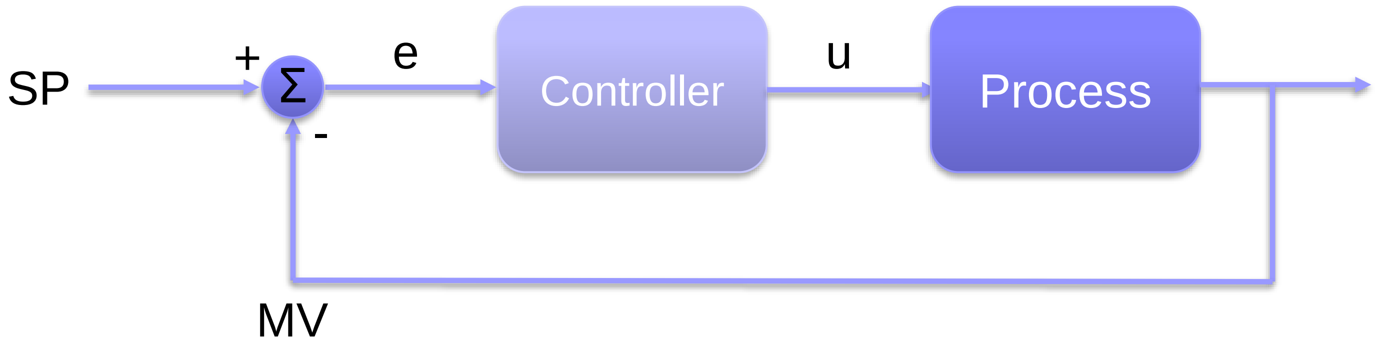
- The controller needs to convert two signals of one physical quantity (such as temperature) into one signal of another (such as valve position)

The Error Signal

- Very often we can think of the controller acting on the **difference** between SP and MV:

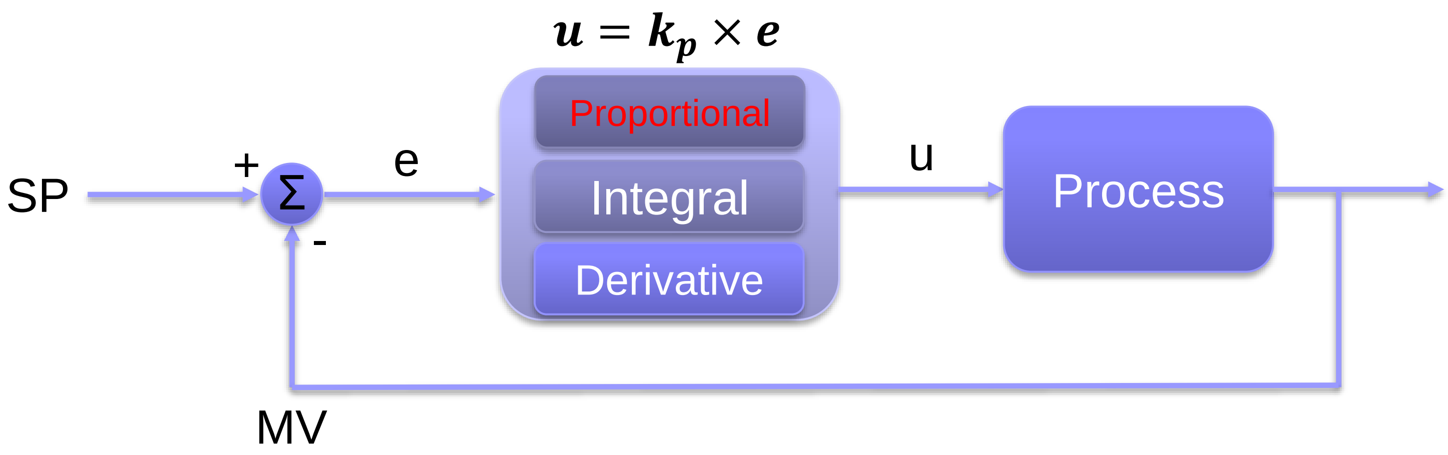


The Closed Loop



This is the ‘classic’ closed loop block diagram representation of a control system

Proportional Control



Proportional Control

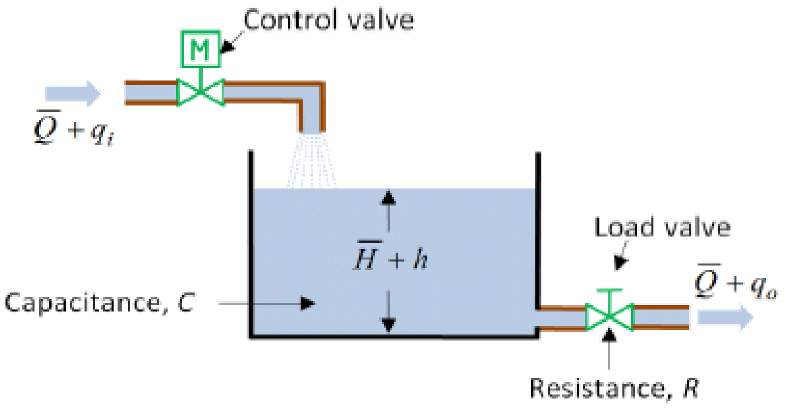
- This is what is referred to as **proportional control.** The control action at any instant is the same as a constant times the error at the same instant
- The constant k_p is the **Proportional Gain,** and is the first of our controller's parameters

Is Proportional Control enough?

- Intuitively it seems like it should be fine on its own: when the error is big, the control input is big to correct it. As the error reduced so does the control input.
- But there are problems…

Problem

- What happens when the error is zero?
- Control input is zero.
- Causes problems if we need to have a nonzero control value while at our setpoint



Problem : steady state error

This problem is normally called **steady state error**

It’s a confusing name. The issue is just that the controller can’t produce any output when the error is zero.

Easiest to see for tank level control: If there is a constant flow out of the tank, the controller must provide the same flow in, while the level is at the setpoint.

Problem : steady state error

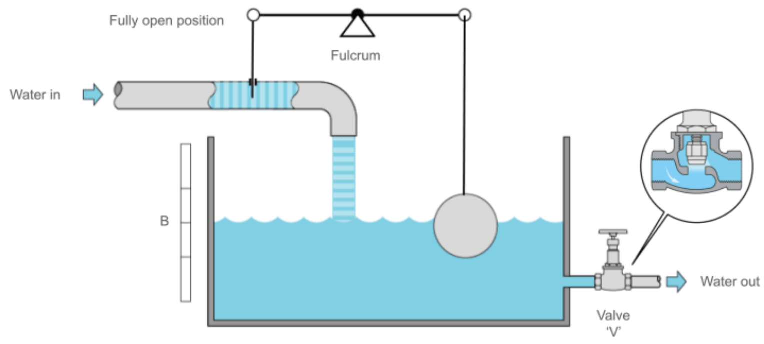


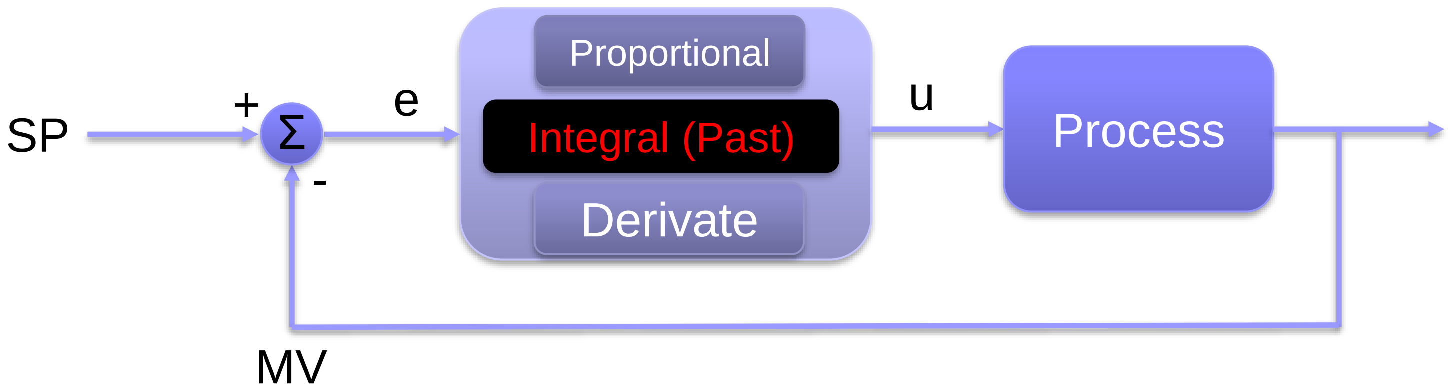
Fig. 5.2.6 Valve open

With P control, once the error has reached a value where $k_p \times e$ is equal to the flow out, the level will stabilize. But it will be different to the setpoint

Solving P control's problems

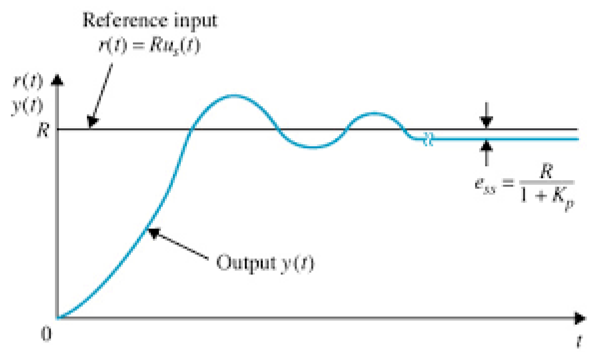
- How to get rid of steady state error?
- Let's ignore the present for the moment and concentrate on what has happened in the **past**

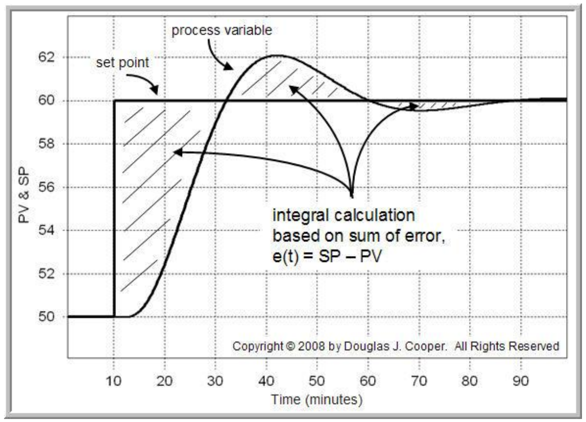
Solving steady state error: ‘the Past’



- Let’s look at the error in the **past**

Solving steady state error





We can examine how the controller error has evolved in the **past**

If we sum up the past values of the error, we can get a value that increases when there is a constant error

Integral Action

We can let the control be given by the sum of past values of the error, scaled by some gain.

In continuous time the sum is an **integral**:

$$\boldsymbol{u = k_i \int_0^t e(\tau) d\tau}$$

Integral gain, Integral time

Here is where confusion can start…

We have an **integral gain $\boldsymbol{k_i}$** which converts an integrated error to a control signal

We would actually like to parameterize this as a **time**, as in how fast we can remove a steady state error

Integral gain, Integral time

Let’s rewrite:

$$\boldsymbol{u = k_i \int_0^t e(\tau) d\tau}$$

As:

is our Proportional Gain, and $\boldsymbol{T_i}$ is our **Integral Time**

The Integral time

Why do we use both $\boldsymbol{k_p}$ and $\boldsymbol{T_i}$ here?

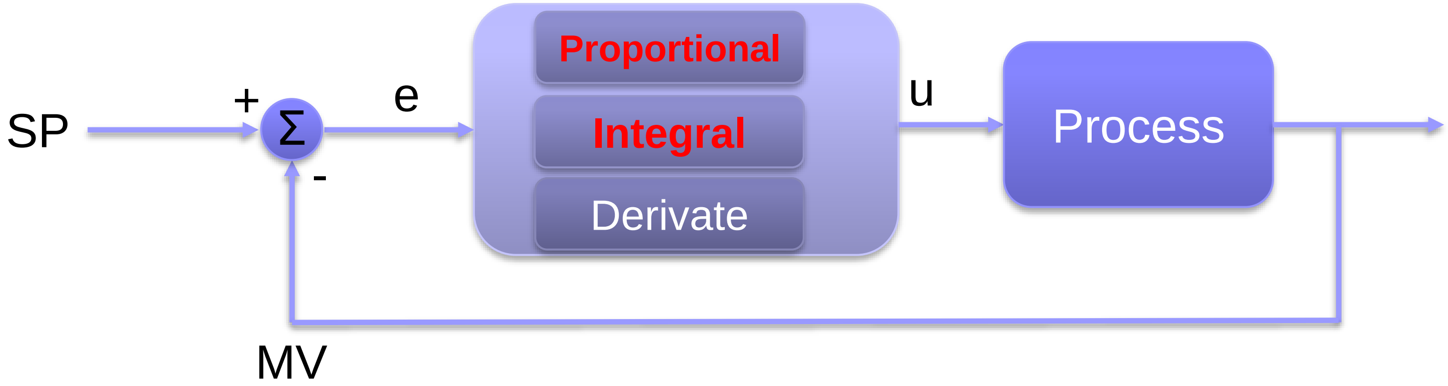
$$\boldsymbol{u = \frac{k_p}{T_i}\int_0^t e(\tau)d\tau}$$

Let's add the proportional and integral parts:

$$\boldsymbol{u = k_p e(t) + \frac{k_p}{T_i}\int_0^t e(\tau)d\tau}$$

Proportional and Integral Controller

$$u = k_p e(t) + \frac{k_p}{T_i}\int_0^t e(\tau)d\tau$$



Proportional and Integral Controller = PI Controller

We can rewrite the control law:

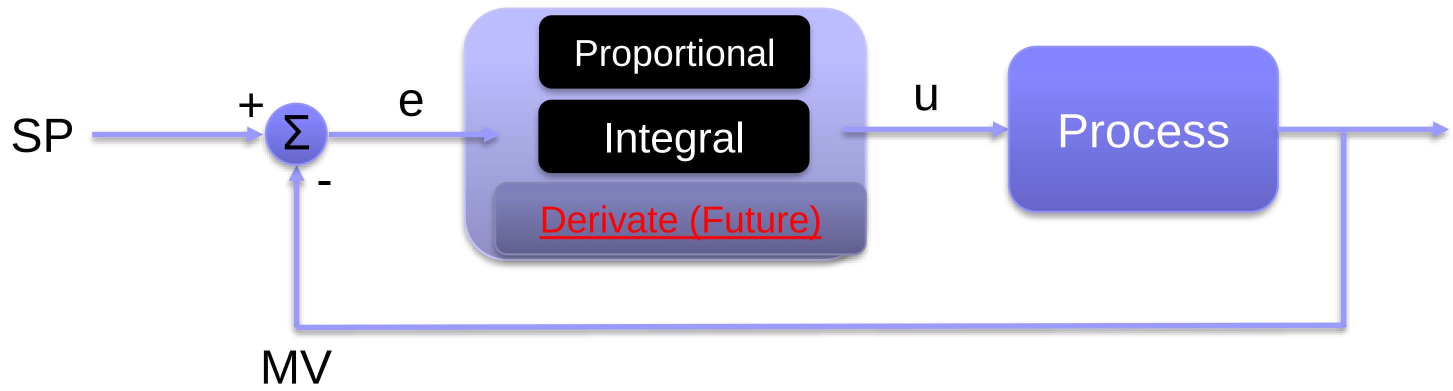
$$\boldsymbol{u = k_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau\right)}$$

$\boldsymbol{k_p}$ is the gain parameter, $\boldsymbol{T_i}$ is the time it takes to fix a steady state error

Solving P control's problems revisited

- We solved the steady state error by adding **integral action** (summing the past)
- How can we solve the oscillation problem?
- Let's look at the future!

Solving oscillations: ‘the Future’



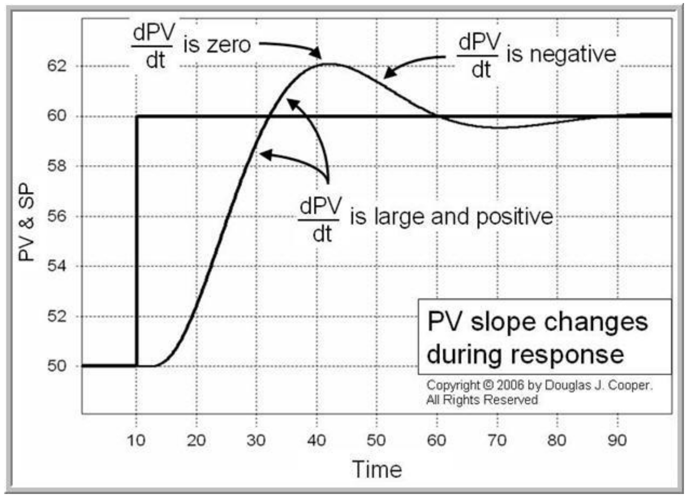
- Let’s look at the error in the **future**

Solving oscillations: ‘the Future’

How do we predict the future of the error?

Look at its **gradient**!

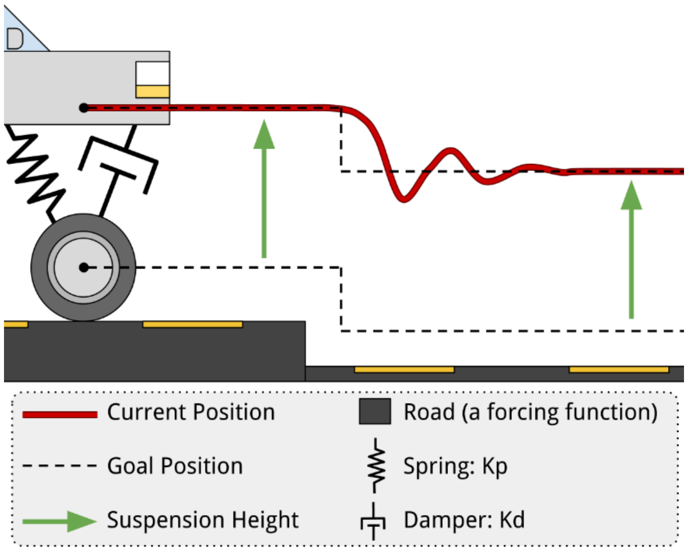
If the gradient (the time derivative) of the error is in a direction that makes the error smaller, we can reduce the control input



Damping

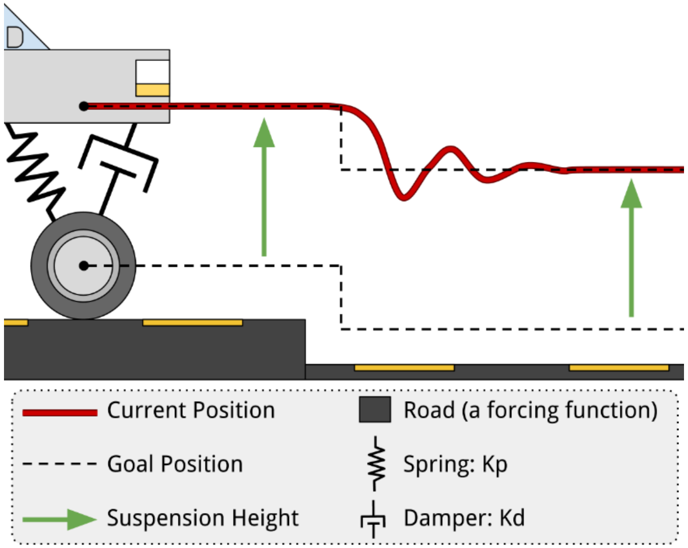
It can be easier to think of this as **damping,** something that resists velocity

Think of the wheel on your car…



Damping

The spring is a proportional controller for the wheel position. The damper adds a **derivative action** by opposing the **velocity** of the wheel



Derivative action

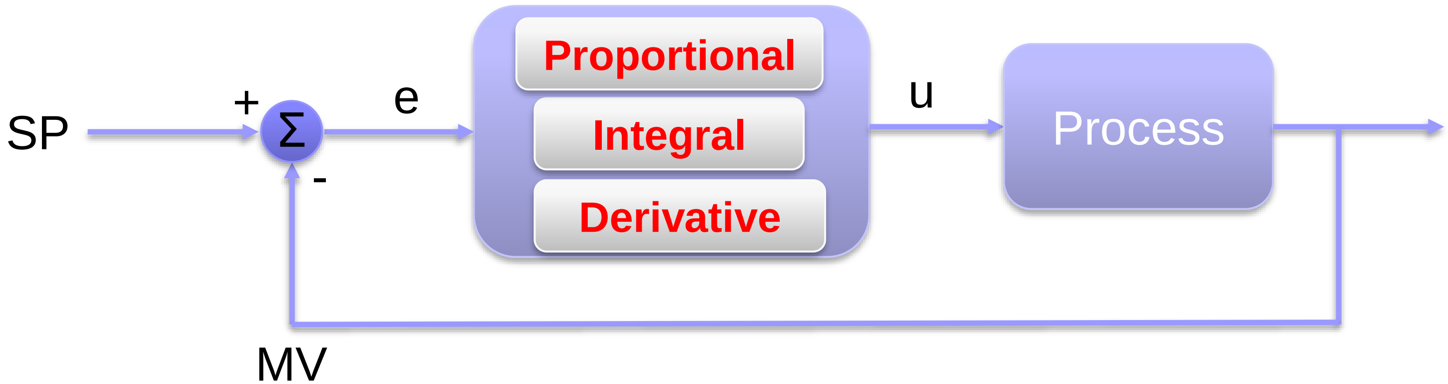
Let's let the control be dependent on the derivative of the error:

$$\boldsymbol{u = k_d \frac{de(t)}{dt}}$$

Here k_d is the derivative gain. Let's again split this into $k_p T_d$, where T_d is the **derivative time**

PID Controller!

$$u = k_p e(t) + \frac{k_p}{T_i}\int_0^t e(\tau)d\tau + k_p T_d \frac{de(t)}{dt}$$

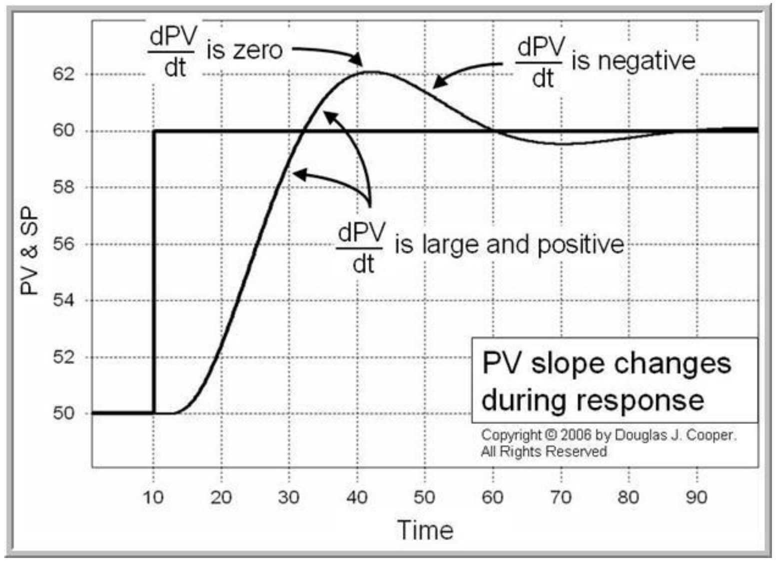


Derivative Time

Why do we want T_d as a parameter?

We can think of it as how far ahead we want to predict!

Easier to relate to process

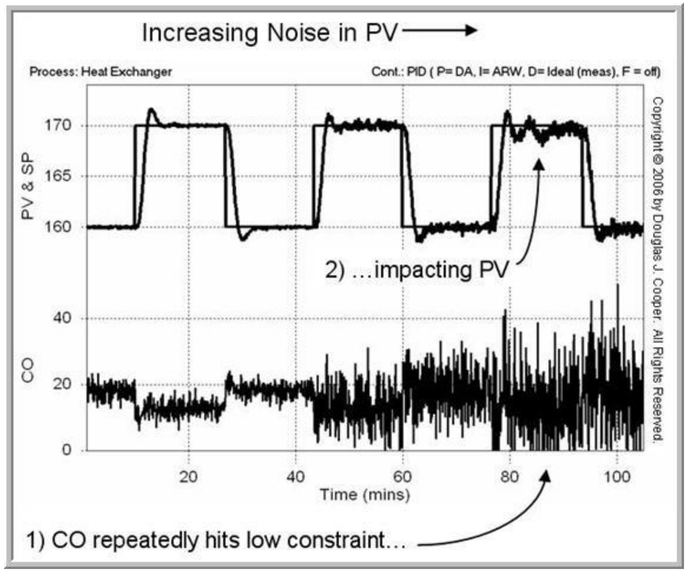


Problems with Derivative action

We know that a derivative amplifies quick changes.

We can get problems if MV is noisy.

Solution is to add low pass filter



Derivative with filter

We already have:

$$\boldsymbol{u = k_p T_d \frac{de(t)}{dt}}$$

Equations will get messy if we add a filter in time domain! Let's use Laplace! Then we can use algebra instead of calculus.

Derivative with filter

Laplace transform frequency variable *s* is also an operator. Multiplication by *s* is derivation, and division is integration. So the derivative part is now

$$U(s) = k_p T_d s E(s)$$

Derivative with filter

We add a low pass filter:

$$U(s) = \frac{k_p T_d s}{1 + \frac{T_d}{T_{ds}} s} E(s)$$

Now we have another parameter T_{ds}which is the filter bandwidth. So introducing derivative action requires two more parameters!

Full PID equation

The complete PID controller now looks like:

$$U(s) = k_p E(s) + \frac{k_p E(s)}{T_i s} + \frac{k_p T_d s}{1 + \frac{T_d}{T_{ds}} s} E(s)$$

Full PID equation

This simplifies to:

$$U(s) = k_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d}{T_d s} s}\right) E(s)$$

ISA PID Form

This is the ISA ‘standard form’ for a PID

$$\frac{U(s)}{E(s)} = k_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d}{T_{ds}} s}\right)$$

We have one gain, and three time constants

P, PI, PD, PID?

For the complete PID controller we have 4 parameters k_p, T_i, T_d, T_{ds}

But we can also choose to use only parts of the controller, for example just, PI, giving 2 parameters to choose.

How do we know when to use what?

The Process Model

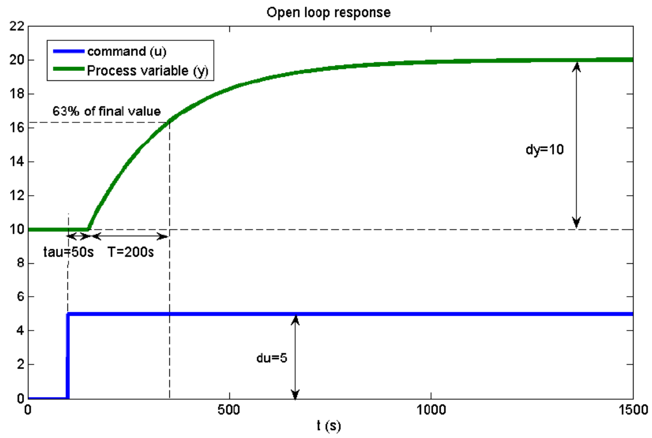
We now know exactly what’s inside our controller, but so far we haven’t said anything about what’s inside our process!

To be able to tune the controller, we need to know something about it.

Open Loop Step Response

- A good starting point is to make an open loop step test on the process
- Note that this is a step on the controller output, not on the setpoint!

Open Loop Step Response



If our step response looks something like this, we have a stable system with a first order response

$$P(s) = \frac{K}{T \cdot s + 1} \cdot e^{-\tau \cdot s}$$

Problem 1

- What kind of compensation improves the steady-state error?

  PI or lag compensation

- What kind of compensation improves transient response?

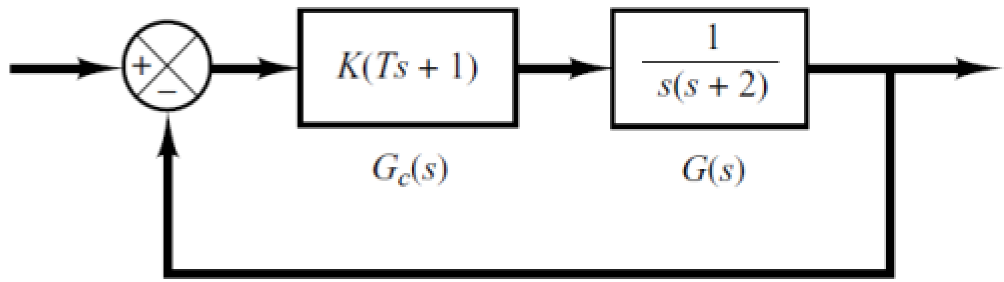
  PD or lead compensation

- What kind of compensation improves both steady-state error and transient response?

  PID or lag-lead compensation

Problem .2

For the system shown in the below figure,



find K, T such that the desired closed loop system poles are located at $-2 \pm j2$.

-Plot the step response before and after the compensator design

$$G_c(s) = K(Ts+1) \qquad G(s) = \frac{1}{s(s+2)}$$

$$\text{CLTF:} \quad \frac{G_c(s)\,G(s)}{1+G_c(s)\,G(s)H(s)} = \frac{K(Ts+1)\left(\frac{1}{s(s+2)}\right)}{1+K(Ts+1)\left(\frac{1}{s(s+2)}\right)}$$

$$\boxed{= \frac{K(Ts+1)}{s(s+2)+K(Ts+1)}}$$

$$\text{CLTF} = \frac{KT_s + K}{s^2 + (2s + KT_s) + K}$$

Standard form:

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

loop system poles:

$$-2 \pm j2$$

poles:

$$= \frac{-(2+KT) \pm \sqrt{(2+KT)^2 - 4(1)(K)}}{2}$$

$$= \frac{-(2+KT)}{2} = -2$$

$$= 2 + KT = 4$$

$$KT = 2$$

$$\frac{\sqrt{(2+KT)^2 - 4k}}{2} = 2j$$

$$\left(\frac{\sqrt{(2+KT)^2 - 4k}}{2}\right)^2 = (2j)^2$$

$$\frac{(2+2)^2 - 4K}{4} = -4$$

$$16 - 4k = -16$$

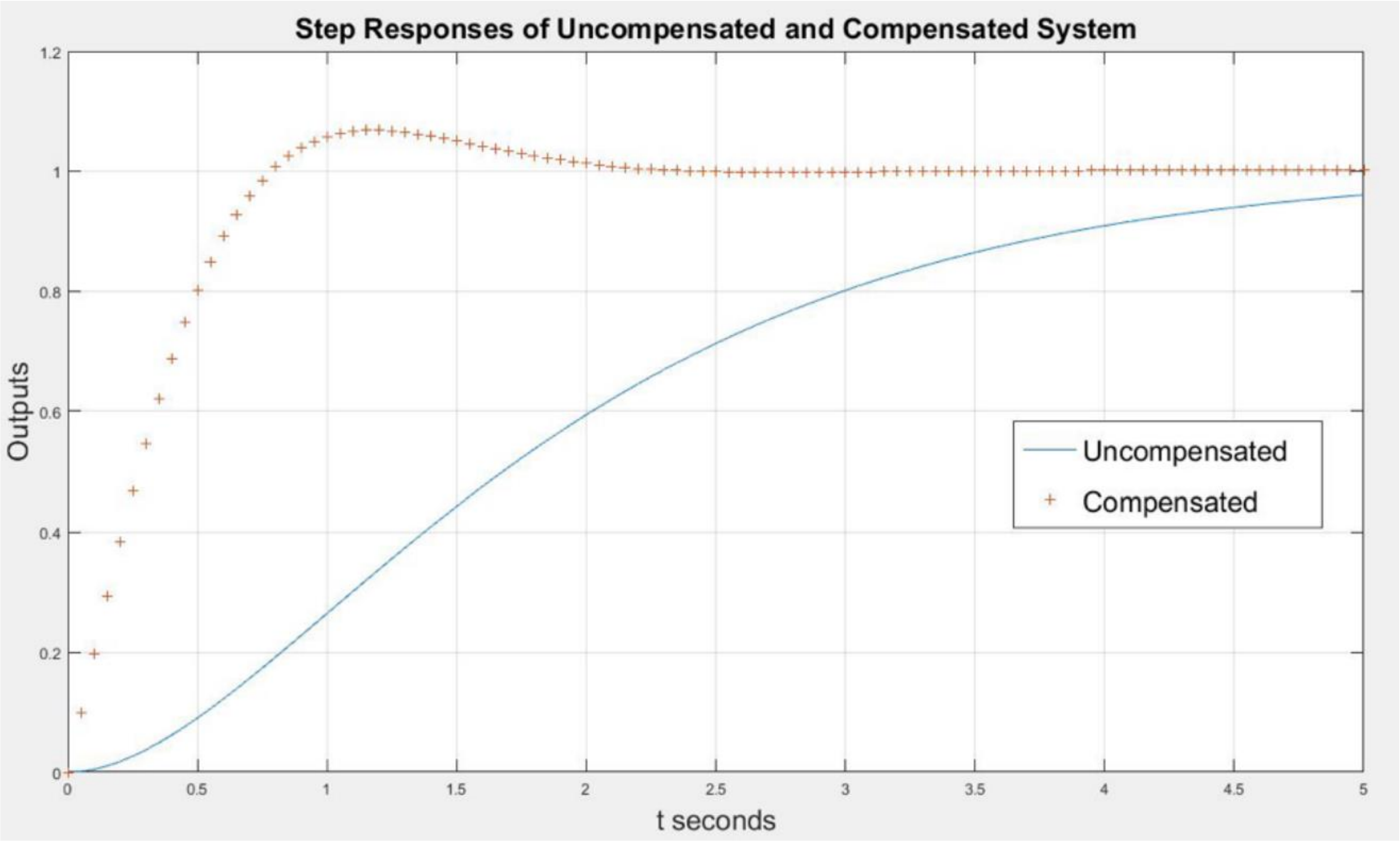
$$k = 8$$

$$T = \frac{2}{8} = \frac{1}{4}$$

CLTF: $$\frac{KTs + K}{s^2 + (2 + KT)s + K}$$

$$= \frac{8\frac{1}{4}s + 8}{s^2 + (2 + 8\frac{1}{4})s + 8} \Rightarrow \frac{2s + 8}{s^2 + 4s + 8}$$

Compensated CLTF


Step Responses of Uncompensated and Compensated System
Outputs
t seconds
Uncompensated
Compensated
0
0.2
0.4
0.6
0.8
1
1.2
0.5
1.5
2
2.5
3
3.5
4
4.5
5

Problem . 3

The open-loop transfer function of a control system is given by

$$KGH(s) = \frac{K}{s(s+1)(s+4)}$$

(a) Obtain the gain K_0 of a proportional controller such that the damping ratio of the closed-loop poles will be equal 0.6. Obtain root-locus, step response and the time-domain specifications for the compensated system.

Solution

The root-locus plot is shown in For $\zeta = 0.6$,

$$\theta = \cos^{-1} 0.6 = 53.13^\circ$$

The line drawn at this angle intersects the root-locus at approximately, $s_1 \simeq -0.41 + j0.56$.

The vector lengths from s_1 to the poles are marked on the diagram

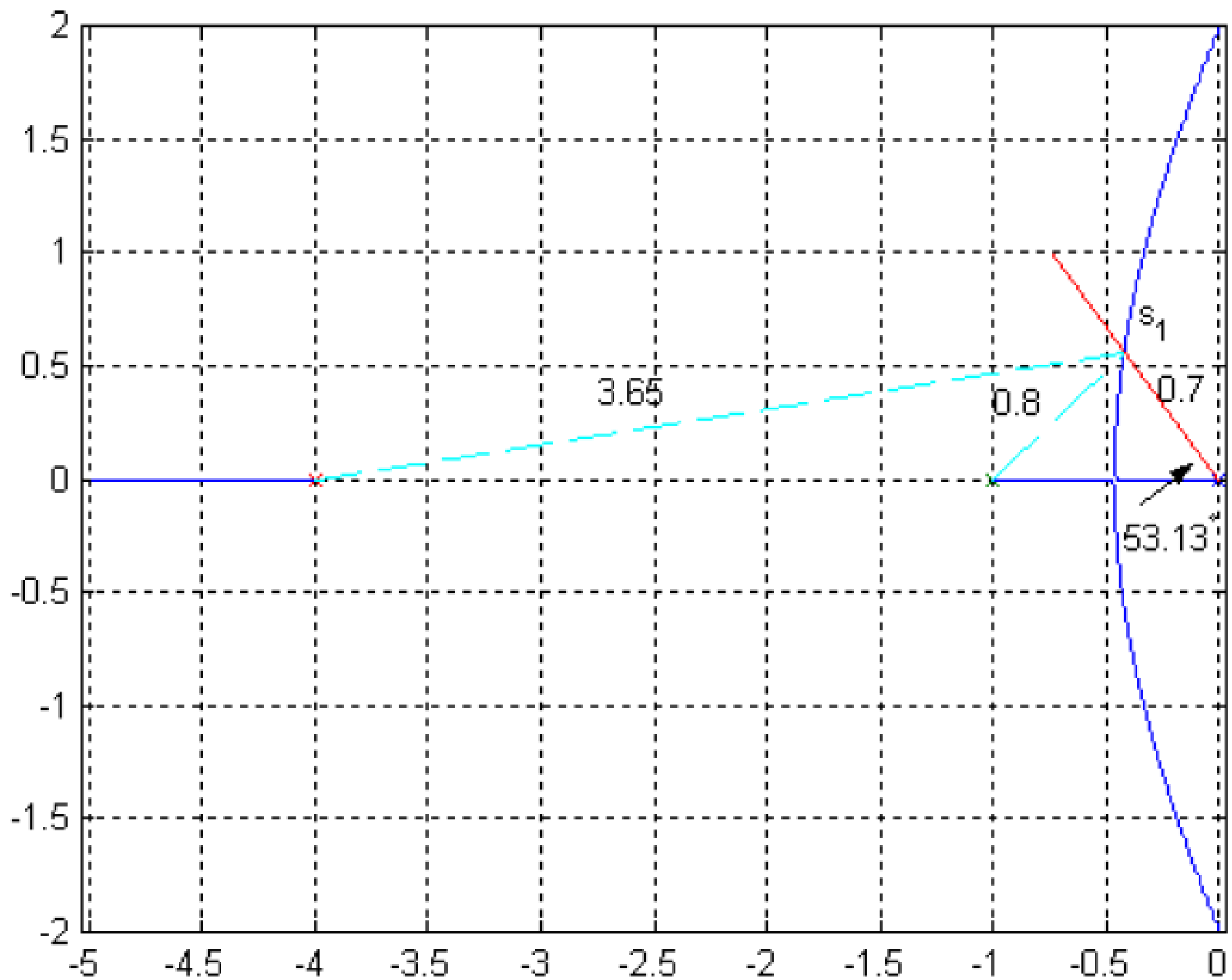


Figure 2 P-Controller Design

. From (1), we have

$$K = (0.7)(0.8)(3.65) = 2.04$$

This gain will result in the velocity error constant of $K_v = \frac{2.04}{4} = 0.51$. Thus, the steady-state error due to a ramp input is $e_{ss} = \frac{1}{K_v} = \frac{1}{0.51} = 1.96$.

The compensated closed-loop transfer function is

$$\frac{C(s)}{R(s)} = \frac{2.05}{s^3 + 5s^2 + 4s + 2.05}$$

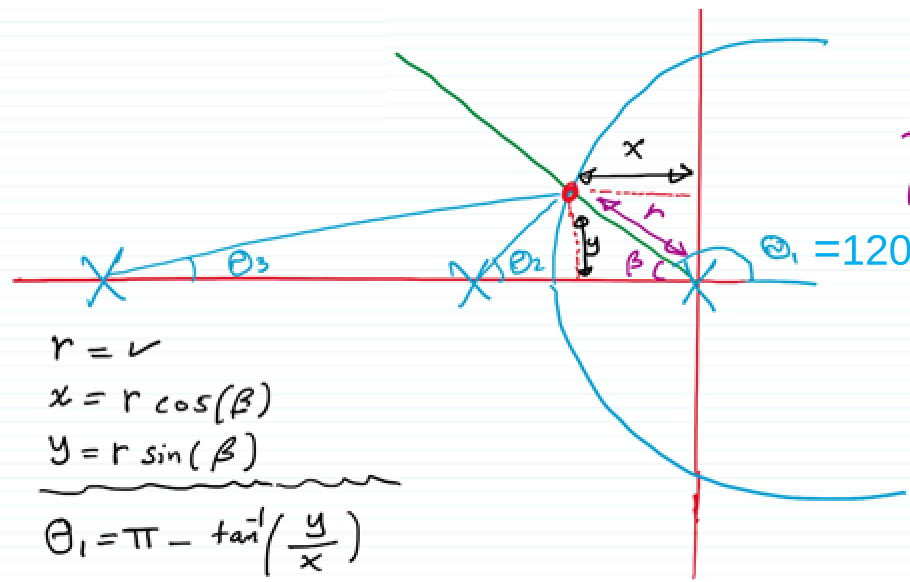
Problem .4

Sketch the root locus for the open loop transfer function of a unity feedback control sytem given below and determine the value of K for $\xi = 0.5$

$$G(s) = \frac{K}{s(s+1)(s+3)}$$

The line drawn at this angle intersects the root-locus at approximately $s = -0.4 + j0.7$,

which will give $K \approx 1.8$



$\xi = 0.5$

$\beta = \cos^{-1}(\xi)$

$= 60^\circ$

$r = ✓$

$x = r\cos(\beta)$

$y = r\sin(\beta)$

$\theta_1 = \pi - \tan^{-1}\left(\frac{y}{x}\right)$

$\theta_2 = \tan^{-1}\left(\frac{y}{|-1| - x}\right)$

$\theta_3 = \tan^{-1}\left(\frac{y}{|-3| - x}\right)$

For an accurate answer to this particular problem ($\zeta = 0.5$), write the CE as:

The characteristic equation is: $s(s+1)(s+3) = -K$

$$s(s+1)(s+3) + K = (s+\alpha)(s^2 + \omega_n s + \omega_n^2) = 0$$

solve for α and ω_n, and hence find K. Thus:

$$s^3 + (\alpha + \omega_n)s^2 + (\alpha\omega_n + \omega_n^2)s + \alpha\omega_n^2 = s^3 + 4s^2 + 3s + K = 0$$

Giving $K = 1.83$.

Problem.5

a. The system is controlled with a P-controller with gain K, see Figure 1. For what values of K is the closed loop system asymptotically stable? (answer for both positive and negative values of K).

b. Suppose that we make a unit step change in setpoint R. Determine the stationary control error E. How small can we make the error using the P controller?

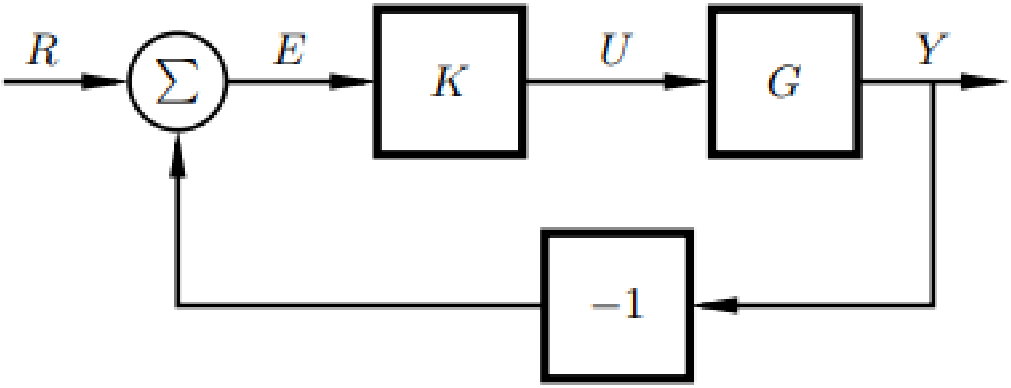


Figure 1: The closed loop system for problem 1.c.

The transfer function of the closed loop system is given by $G_{yr}(s) = Y(s)/R(s)$:

$$\begin{aligned} G_{yr}(s) &= \frac{KG(s)}{(1+KG(s))} \\ &= \frac{K(s-4)}{(s^2+9s+8)+K(s-4)} \\ &= \frac{K(s-4)}{s^2+s(9+K)+(8-4K)} \end{aligned}$$

The system is asymptotically stable when both poles are strictly in the left-half plane (have negative real parts).

This is true if and only if a1, a2 > 0 for the characteristic polynomial s^2 + a1s + a2. Therefore, **9 + K > 0** **K > −9** **8 − 4K > 0** **8 > 4K** **−9 < K < 2**	TR; S2 | 1 8-4k S1 | 9+k S0 | 8-4k **9+k >0 & 8-4k>0** **K>-9 k<2** **-9<k<2**

b

The static error of the response to a unit step can be computed applying the final value theorem to the Laplace transform of the error in response to a step input. The transfer function from the reference to the error is:

$$G_{er}(s) = \frac{E(s)}{R(s)} = \frac{s^2 + 9s + 8}{s^2 + 9s + 8 + K(s-4)}$$

We then apply the final value theorem. With a unit step input $R(s) = \frac{1}{s}$ the static error ($t \to \infty$) is given by evaluating $E(s) = sG_{er}(s)\frac{1}{s}$ where $s \to 0$.

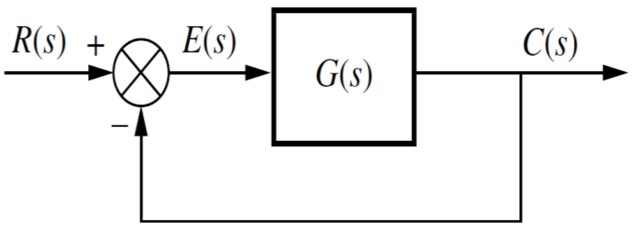
$$G_{er}(0) = \frac{8}{8-4K}$$

Therefore we want to choose K within the stability range $-9 < K < 2$ to maximize the denominator in order to minimize the static error. This is obtained with $K = -9$, which gives a static error of $e(\infty) = 8/44 \approx 0.18$.

Problem.6

9. Consider the unity feedback system shown in Figure P9.1 with [Section: 9.3]

$$G(s) = \frac{K}{(s+4)^3}$$



a. Find the location of the dominant poles to yield a 1.6 second settling time and an overshoot of 25%.

b. If a compensator with a zero at -1 is used to achieve the conditions of Part **a**, what must the angular contribution of the compensator pole be?

c. Find the location of the compensator pole.

d. Find the gain required to meet the requirements stated in Part **a**.

e. Find the location of other closed-loop poles for the compensated system.

f. Discuss the validity of your second-order approximation.

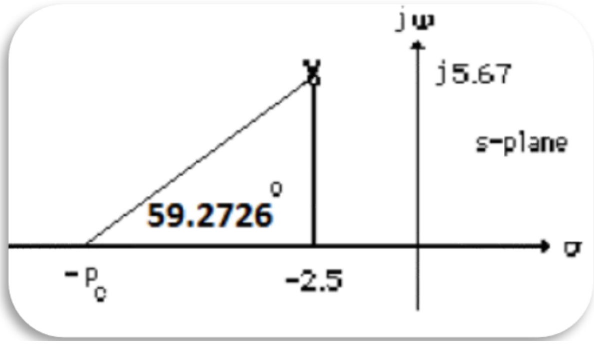
g. Use MATLAB or any other computer program to simulate the compensated system to check your design.

MATLAB ML

a. $\zeta\omega_n = \frac{4}{T_s} = 2.5;\ \zeta = \frac{-\ln\left(\frac{\%OS}{100}\right)}{\sqrt{\pi^2 + \ln^2\left(\frac{\%OS}{100}\right)}} = 0.404$. Thus, $\omega_n = 6.188$ rad/s and the operating point is $-2.5 \pm j5.67$.

b. Summation of angles including the compensating zero is -120.7274°. Therefore, the compensator pole must contribute 120.7274° - 180° = -59.2726°.

c. Using the geometry shown below, $\frac{5.67}{P_c - 2.5} = \tan 59.2726^\circ$. Thus, $P_c = 5.87$.

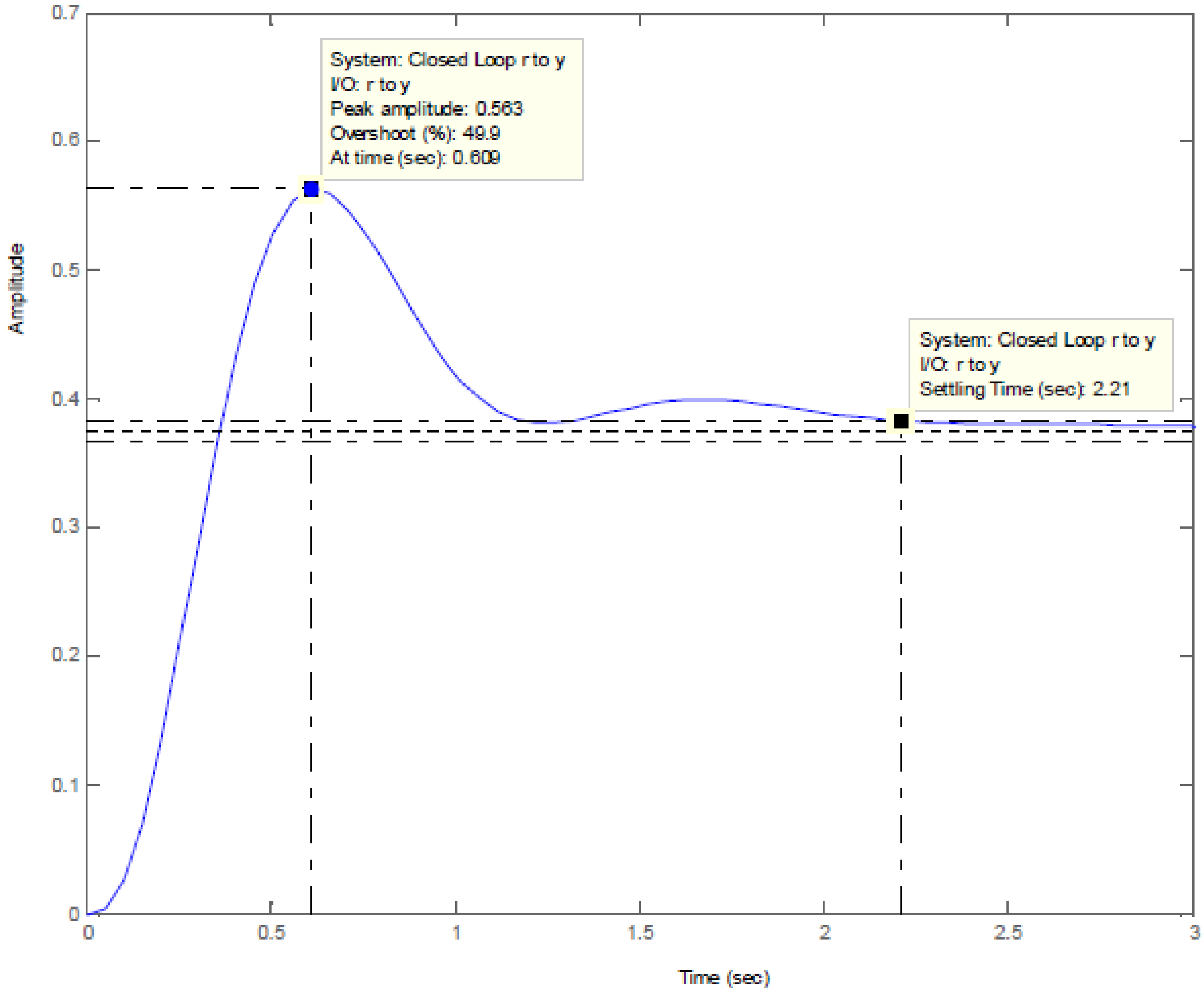


d. Adding the compensator pole and using -2.5 + j5.67 as the test point, K = 225.7929.

e. Searching the real axis segments for K = 225.7929, we find higher-order poles at -11.5886, and --1.3624.

f. Pole at -11.5886 is 4.64 times further from the imaginary axis than the dominant poles. Pole at --1.3624 may not cancel the zero at -1. Questionable second-order approximation. System should be simulated.

g.



System: Closed Loop r to y
I/O: r to y
Peak amplitude: 0.563
Overshoot (%): 49.9
At time (sec): 0.609
System: Closed Loop r to y
I/O: r to y
Settling Time (sec): 2.21
Amplitude
Time (sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.5
1
1.5
2
2.5
3

References

Nise, N. S. (2015).

Control Systems Engineering.

Wiley, 7 edition.