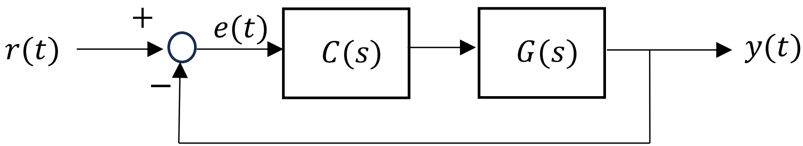
Week 8 Slides



Recall, if

$$T(s) = \frac{C(s)G(s)}{1 + KC(s)G(s)}$$

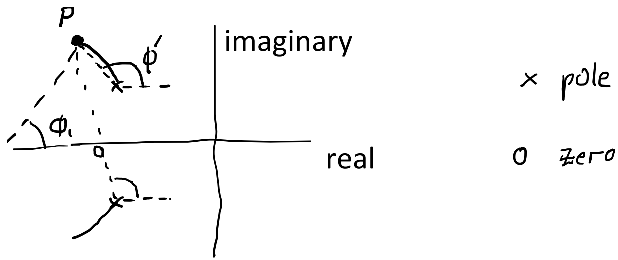
the root locus focuses on the characteristic equation

$$C(s)G(s) = -1$$

Especially the angle sum rule

$$\angle CG = (\phi_1 + \cdots) - (\phi_1' + \cdots) = \pi$$

recall: for any point (P) on the root locus ϕ is and angle drawn from a zero (of CG) and ϕ' is the angle drawn from a pole.



This is because of the “add arguments” property of multiplying complex numbers

$$CG = K\frac{(s - z_1) \ldots (s - z_m)}{(s - p_1) \ldots (s - p_n)}$$

and $\angle(-1) = \pi$

Note: any poles and zeros of $C(s)$ can be included with $G(s)$, and then $C = K$ and the open-loop transfer function is written $KG(s)$.

Root Locus "rules"

- starts at poles of $KG(s)$ and ends at zeros
- real axis segments have an odd number of poles or zeros to the right
- there are $n - m$ asymptotes angled at $(\pi, 3\pi, 5\pi, \dots)/(n-m)$ passing through the real axis at

$$a = \frac{\Sigma p_i - \Sigma z_i}{n - m}$$

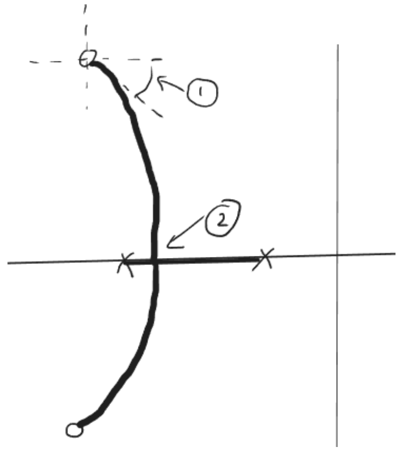
- angle sum: zeros – poles

$$\Sigma\phi_i - \Sigma\phi_i' = \pi$$

- breakaway points (see below)

the **angle of departure** or arrival (1) is the direction when leaving a pole or arriving at a zero.

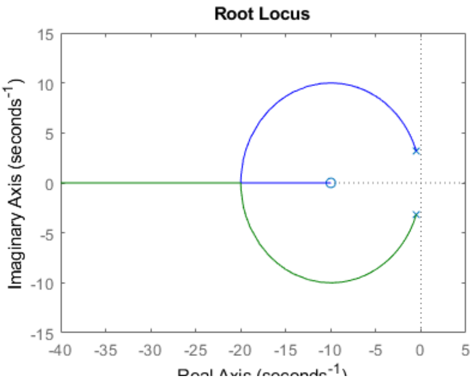
a **breakaway point** (2) is where the root locus jumps out of the real axis or jumps into it.



Breakaway point rule

e.g., Calculate the point where the root locus meets the real axis (break-in)

$$G(s) = \frac{0.1s + 1}{s^2 + s + 10} = \frac{N(s)}{D(s)}$$



The **rule for a breakaway point** is

$$N'(s)D(s) = N(s)D'(s)$$

Hence

$$0.1 \times (s^2 + s + 10) = (0.1s + 1)(2s + 1)$$

This is solved to give $s = 0$ or $s = -20$; clearly it's the second solution that's needed, and this is what we see above.

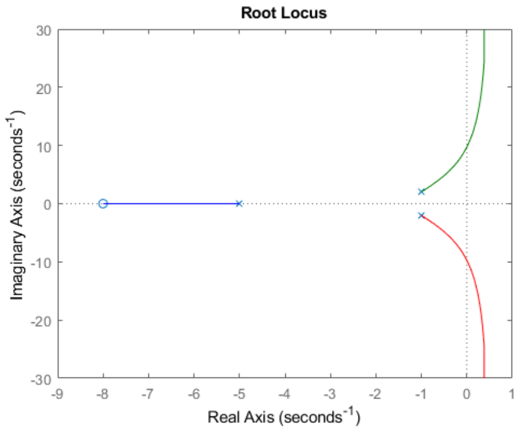
The other solution would have negative K (recall, to find the gain at a point of interest use $KG(s) + 1 = 0$)

Angle of departure – application of the angle sum rule.

For the transfer function

$$G(s) = \frac{s+8}{s^3 + 7s^2 + 15s + 25}$$

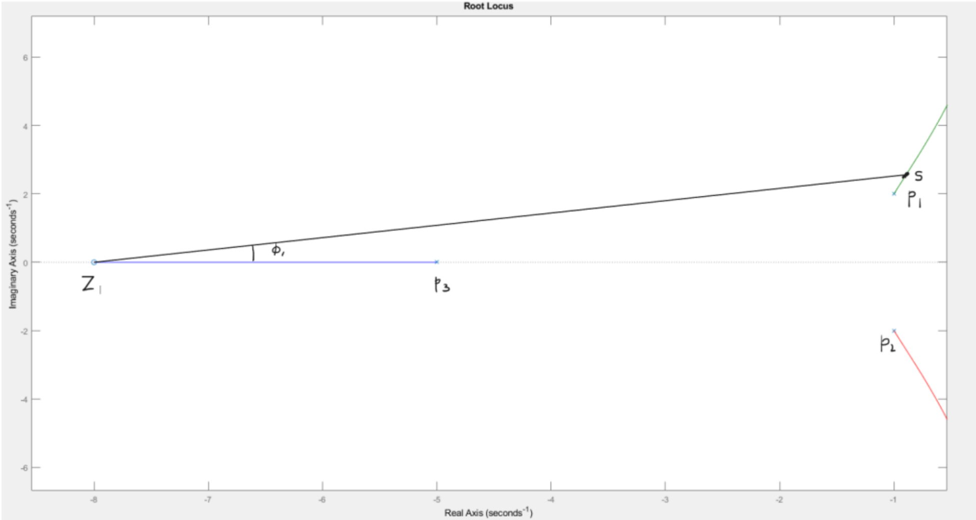
With proportional control we obtain the root locus shown



Consider a test point s very close to the pole at $-1+2j$

Apply the angle sum rule

$$\phi_1 - \phi_1' - \phi_2' - \phi_3' = \pi$$



Since s is close to p_1

$\tan \phi_1 = 2/7$ gives ϕ_1= 0.2783 radians (= 15.95 deg)

Also ϕ'_2 = $\pi/2$ (90 deg), and ϕ'_3=atan(2/4) = 0.464 (26.57 deg)

hence (in degrees)

$$15.95 - \phi_1' - 90 - 26.57 = 180$$

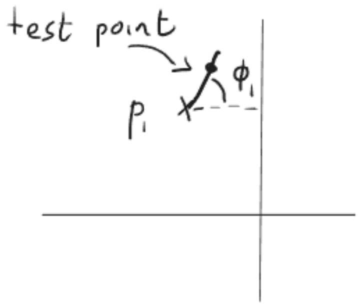
from which

$$\phi_1' = -280.6°$$

or

$$\phi_1' = 79.4°$$

This is the angle for departing p_1



We can do this in a Matlab script, but a few new commands are needed

```
[z,p,k]=zpkdata(G,'v'); %find zeros, poles and
gain all at once

G=zpk(z,p,k); %create a transfer function with
these parameters

g=evalfr(G,s); %evaluate a transfer function at
point s
```

For angle of departure we isolate the pole of interest …

$$\frac{(s-z_1)}{(s-p_1)(s-p_2)(s-p_3)} = \frac{(s-z_1)}{(s-p_1)(s-p_3)} \times \frac{1}{s-p_2} = G_2(s) \times \frac{1}{s-p_2}$$

With s very close to the pole:

$$\angle G_2(p_2) - \theta = \pi$$

or (adding 2π)

$$\theta = \pi + \angle G_2(p_2)$$

Here's the script and the answer (=79.38 deg)

```
%% Angle of departure script
clear, close all
s=tf('s');
G=(s+8)/(s^3+7*s^2+15*s+25);
[z,p,k]=zpkdata(G,'v'); %find zeros, poles and gain all at once

%remove the pole of interest, -1+2j = second element
pp=[p(1);p(3)];
G2=zpk(z,pp,k); %create a transfer function with one pole removed

%find the departure angle at that pole
s=p(2);
g2=evalfr(G2,s); %evaluate the transfer function at point s
ang=pi+angle(g2); %angle of departure in radians
```

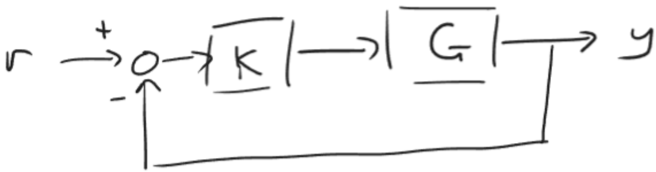
```
ang: 1x1 double =

    1.3854
```

Tutorial Questions

1. For the breakaway point example with transfer function $G(s) = \frac{0.1s+1}{s^2+s+10}$, find the gains associated with $s = 0$ and $s = -20$.
2. Check the positive gain value using **rlocus** in Matlab.
3. Check both gain values using **evalfr** in Matlab.
4. Sketch the general form of the root locus for $G(s)$ with three different stable poles, all with equal real parts, and no zeros. Pay particular attention to the angles of departure.
5. Choose suitable values for the poles and check your sketch using **rlocus**.
6. Repeat questions 4 and 5 in the case with 2 complex stable poles and one zero, all with equal real parts.
7. Calculate the position of the breakaway point in case $G(s) = \frac{s+2}{(s+2)^2+4}$ and check using **rlocus**.

Example root locus when K is not a simple gain.

Normally the parameter to vary is an overall factor in the loop gain.



$$1 + KG = 0 \Rightarrow D(s) + KN(s) = 0$$

(any poles or zeros of the controller have been ‘given a free transfer’ to G)

Sometimes we may want to vary a different parameter such as a derivative gain. In this case we should do some algebra to **rewrite the closed-loop characteristic equation**, K multiplying some new ‘fake numerator’, keeping the form $D(s) + KN(s) = 0$.

Consider again the simple second-order plant

$$G = \frac{1}{s^2 + s + 10}$$

Previously we chose a PD controller of the form $C(s) = K(1 + 0.1s)$ so K was a simple factor.

Now we want to vary the derivative gain as an independent parameter:

$$C(s) = 10 + Ks$$

with the proportional gain is fixed. The characteristic equation becomes

$$1 + \frac{10 + Ks}{s^2 + s + 10} = 0$$

so

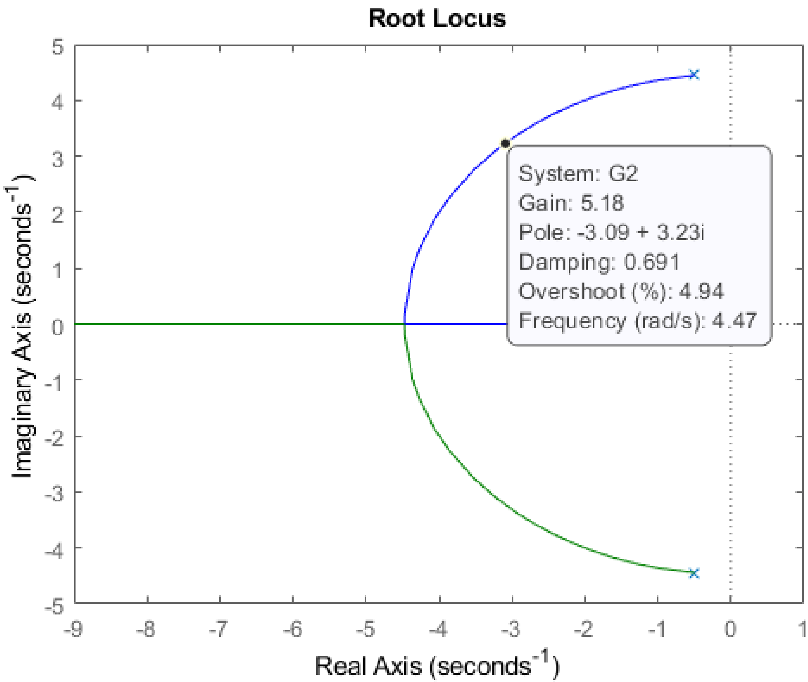
$$s^2 + s + 10 + 10 + Ks = 0$$

$$s^2 + s + 20 + Ks = 0$$

Comparing with the standard case, K multiplies the numerator, and the other term is D. Matlab then gives a root locus for varying the derivative gain

```
clear
close all
s=tf('s');G=1/(s^2+s+10);
N2=s;
D2=s^2+s+20;
G2=N2/D2;
rlocus(G2)
```

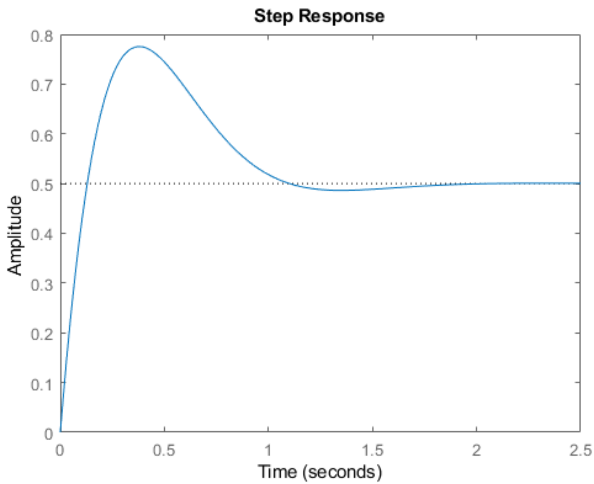
Note that we don't actually use $G(s)$ in this, just the 'fake' N and D.


Root Locus
System: G2
Gain: 5.18
Pole: -3.09 + 3.23i
Damping: 0.691
Overshoot (%): 4.94
Frequency (rad/s): 4.47
Imaginary Axis (seconds^-1)
Real Axis (seconds^-1)
5
4
3
2
1
0
-1
-2
-3
-4
-5
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
1

There may still be some iteration to do, e.g. increasing the proportional gain and repeating, but the design process is predictable and reliable.

Of course we can only vary one gain at a time.

```
K=5.18
C=10+K*s;
T=feedback(C*G,1)
figure
step(T)
```



We have now covered all the methods and rules for root locus, the Matlab function **rlocus** being perhaps the most useful in practice.

On the other hand, knowing some of the theory above is also essential. For example, we know that a system with $m - n = 3$ will always go unstable when the gain is set too high.

In most cases, when asked to sketch a root locus you can check it out first in Matlab, at least to confirm the general shape.

In the following we can use root locus to help design a PID controller.

PID Control

Standard form:

$$C(s) = K_p + K_d s + \frac{K_i}{s}$$

Equivalent form:

$$C(s) = K_p \left(1 + T_d s + \frac{1}{T_i s}\right)$$

This can be be convenient since T_d and T_i have units of seconds, and so related to the time constants of the system being controlled.

PID design using the Ziegler Nichols method

This is a simple PID design method, also applicable to PI control.

A proportional controller is implemented in closed-loop and the gain is increased until a critical gain is reached.

The critical point is when a sustained oscillation is obtained – this is the point where the root locus reaches the imaginary axis.

The critical (or ultimate) gain K_u is recorded, as is the period P_u of the sustained oscillation.

For PID control the Ziegler-Nichols 'optimal' gains are [see for example, Feedback Control of Dynamic System by Franklin, Powell and Emami-Naeini]

$$K_p = 0.6K_u \quad T_i = \frac{P_u}{2} \quad T_d = \frac{P_u}{8}$$

We 'back off' the proportional gain and set the time constants based on the period of oscillation. For PI control the settings are

$$K_p = 0.45K_u \quad T_i = \frac{P_u}{1.2}$$

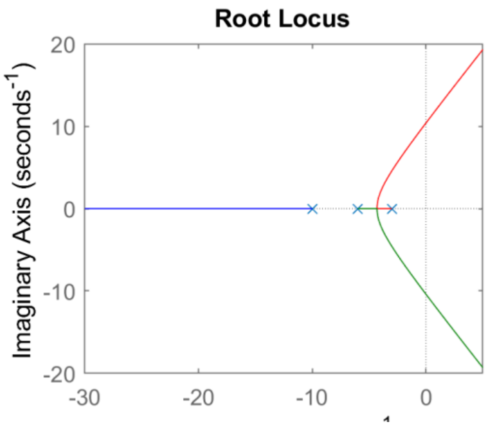
(reduce K_p and K_i more than previously, for stability reasons)

PID Example 1

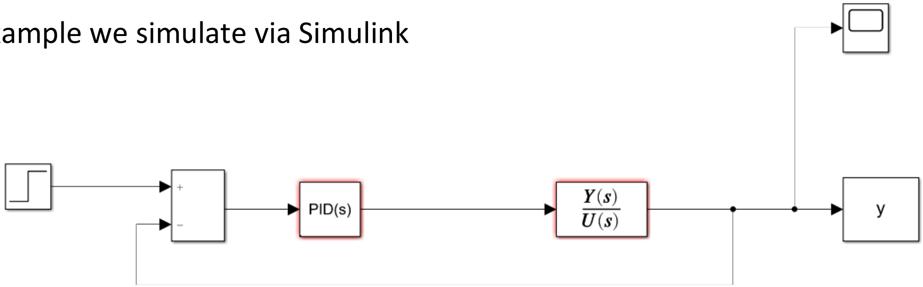
The following plant is to be controlled by a PID controller.

$$G = \frac{10}{(s+3)(s+6)(s+10)}$$

From the asymptotes we know it will become unstable for large (proportional) gain:



In this example we simulate via Simulink

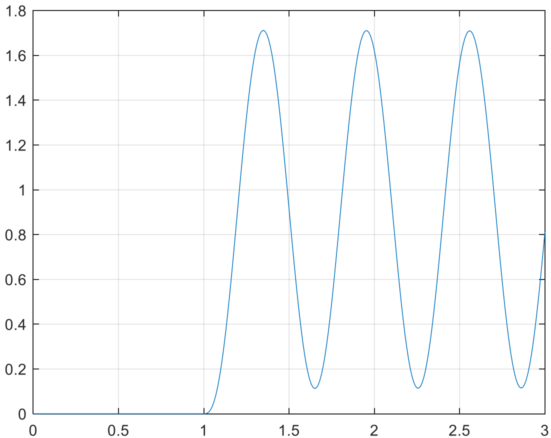


But still using a transfer function block, with coefficients **num** and **den**.
The following will get things started and then Kp is gradually increased

```
clear
close all
s=tf('s');
N=10;
D=(s+3)*(s+6)*(s+10);
G=N/D;
rlocus(G);
[num,den]=tfdata(G,'v');
Kp=1;Kd=0;Ki=0; %to start with
```

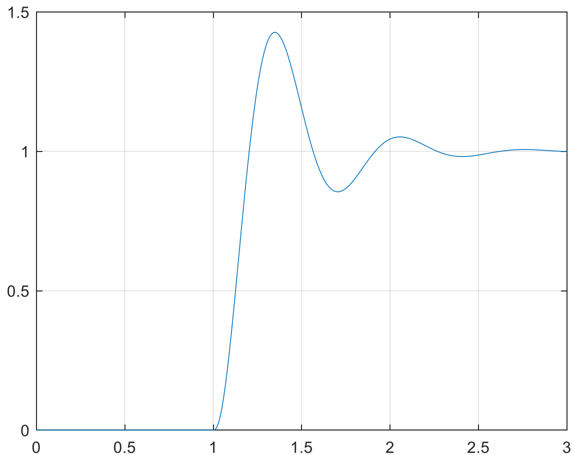
We don't really need the root locus … in the lab we might not even know a suitable transfer function model. Increasing the gain, eventually we hit on the critical (ultimate) gain Kp=187:

```
% PID control
Kp=0.6*Ku;Ti=Pu/2;Ki=Kp/Ti;Td=Pu/8;Kd=Kp*Td;
sim('PID_01',3)
figure,plot(tout,y),grid
```



From the root locus we get the same result for the ultimate gain and period, which is found to be around 0.652 seconds.

The PID controller gains are then found from Ziegler-Nichols:
[Kp,Kd,Ki] =112.2   9.1443 344.1718 and the step response is



This seems to be a reasonable response, though we might want to increase Kd a little more to reduce the overshoot. Ziegler-Nichols gives a very good starting point for further tuning, but not really 'optimal gains'.

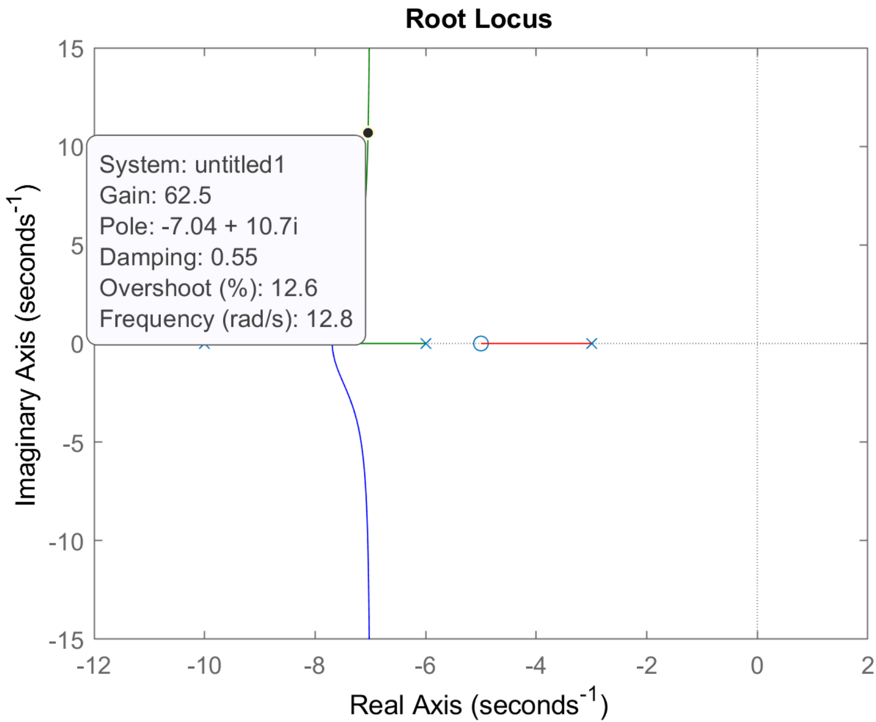
PID Design Example 2 (root locus and pole placement)
Using the same plant as before, use a two-step design process, choosing a real zero to represent a PD controller, then introduce a integral control in a particular way that does not 'mess up' the overall root locus.

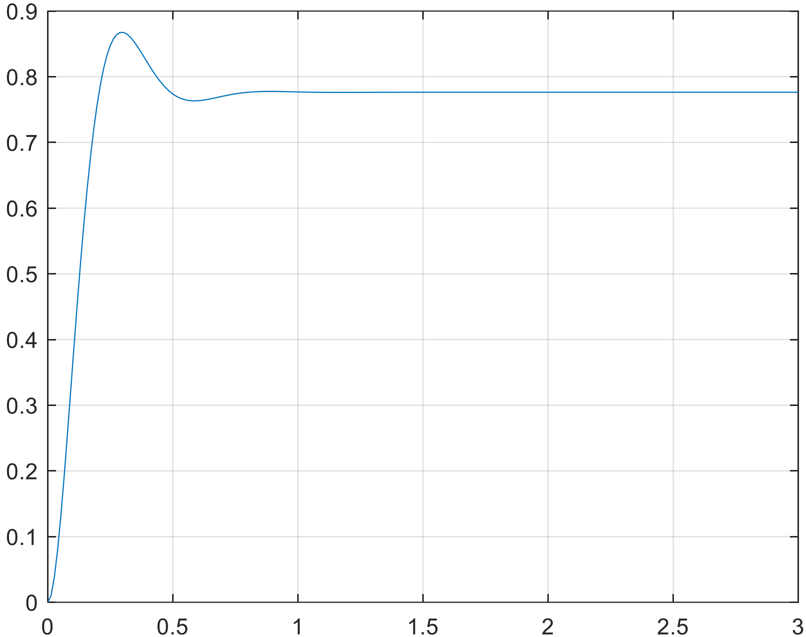
In the first part set

$$C(s) = K(s + a)$$

Then (for example) setting $a = 0.2$ and aiming for a damping ratio 0.55 we obtain from the root locus: K=62.5.

Below are the root locus and step response


Root Locus
System: untitled1
Gain: 62.5
Pole: -7.04 + 10.7i
Damping: 0.55
Overshoot (%): 12.6
Frequency (rad/s): 12.8
Imaginary Axis (seconds^-1)
Real Axis (seconds^-1)
15
10
5
0
-5
-10
-15
-12
-10
-8
-6
-4
-2
0
2


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3

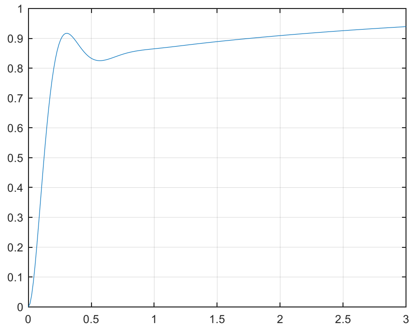
This looks quite good except for the steady-state, so now introduce integral control (to make the system Type 1) in the form

$$C(s) = \frac{K(s+a)(s+b)}{s}$$

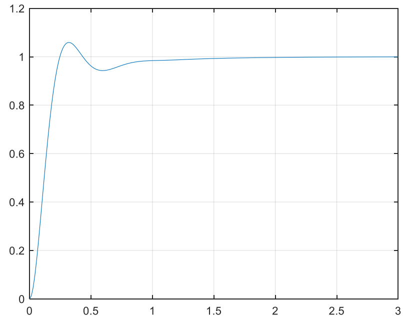
This doesn't quite look like PID, but it is … see Problem 9. Here b should be small, so it does not affect the root locus (we add a pole and zero close together).

The step response with b=0.5 is shown below on the left, the settling is maybe a little too slow, hence we try again with b=2 on the right, which seems to work very well.

It is possible to use root locus properties (angle sum) to force the closed-loop system to have a pre-defined pole – see Problem 10 below.


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3

b=0.5


1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
3

b=2

Tutorial Problems continued

8. In PID example 1, use the PI form of Ziegler Nichols. How do the results compare with the PID step response?
9. In PID example 2, find the PID gains is standard format. Also plot the root locus with and without the term $(s+b)/s$ with b=0.5 or 2.
10. In PID example 2, find the value of a which makes the root locus go through the point $s=-5+5j$. To do this, find the angle sum at this point for G(s). Then $\angle G(s)+\phi=\pi$ gives an equation which can be used to determine the unknown point a. [ANS. a=8.44]
11. Carrying on from question 10, use root locus to find the required gain K.
12. Carrying on from question 11, complete the PID controller design finding a suitable value for b. Plot the step response and find the values of K_p, K_d, K_i.