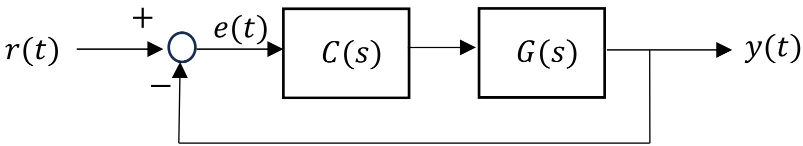
Week 7 Slides



For a closed-loop control system we know that

- Controller gains affect steady-state error through the position, velocity and acceleration error constants
- These error constants are found from the open-loop transfer function $C(s)G(s)$
- Controller gains affect pole locations for the closed-loop transfer function

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

- These pole locations come from solving the characteristic equation

$$1 + C(s)G(s) = 0$$

This week we aim to use $C(s)$ to shift poles so they are fast enough (real part) and relatively stable (angle θ) using the root locus method.

Example 1 – introducing the idea of a **root-locus plot** by using the Matlab commands **feedback** and **pole**.

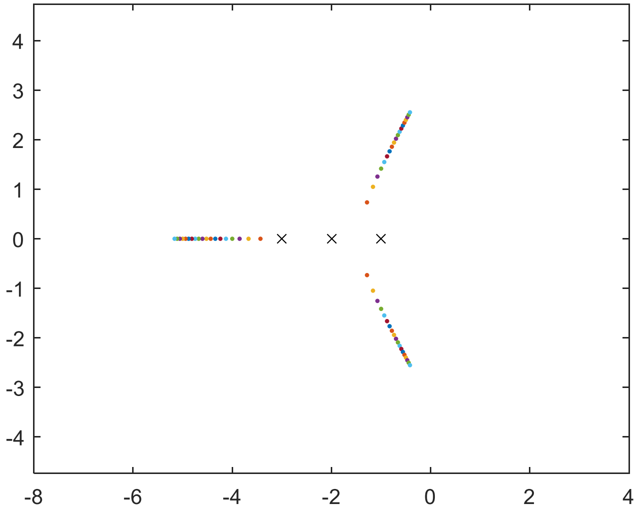
The following code here uses proportional control, gain = K, with a third-order plant

$$G(s) = \frac{6}{(s+1)(s+2)(s+3)}$$

Applying feedback and plotting the poles for different values of K gives a 'map' of the closed-loop system.

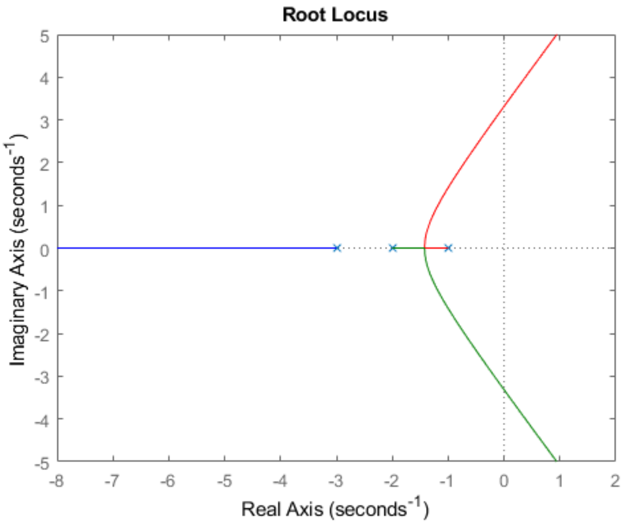
Starting with K very small (0.001) we plot X and then increase until K=5

```
%% example 1
clear, close all
s=tf('s');
N=6;
D=(s+1)*(s+2)*(s+3);
G=N/D;
KK=[0.001:0.25:5];
figure
for i=1:length(KK)
    K=KK(i);
    T=feedback(K*G,1);
    p=pole(T);
    if i==1
    plot(real(p),imag(p),'kX')
    else
    plot(real(p),imag(p),'.')
    end
    xlim([-8,4])
    axis('equal')
    hold on
end
```

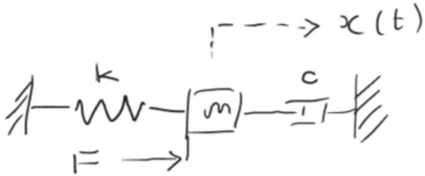


What happens when K increases more?

The same pattern is drawn by Matlab, using **rlocus(G)**. It's the same information as above but with nicer curves!



Example 2 – With a very simple example we can find the closed-loop poles without Matlab (We used a similar example previously)



$$m\ddot{x} + c\dot{x} + kx = F \ , \ \ G(s) = \frac{1}{s^2 + s + 10}$$

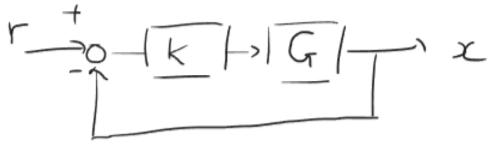
The open-loop characteristic equation is quadratic

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

with solutions

$$s = -0.5 \pm j\sqrt{9.75}$$

In closed-loop



$$T(s) = \frac{X(s)}{R(s)} = \frac{KG}{1 + KG}$$

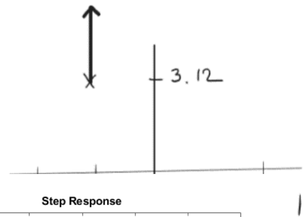
Then

$$1 + K\frac{1}{s^2 + s + 10} = 0 \;\; \rightarrow \;\; s^2 + s + 10 + K = 0$$

and

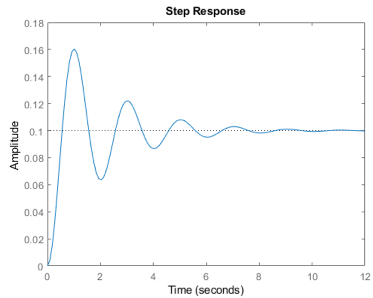
$$s = -0.5 \pm j\sqrt{9.75 + K}$$

So if K increased from zero to large values, the poles move vertically up the complex plane, **starting at the open-loop poles**.



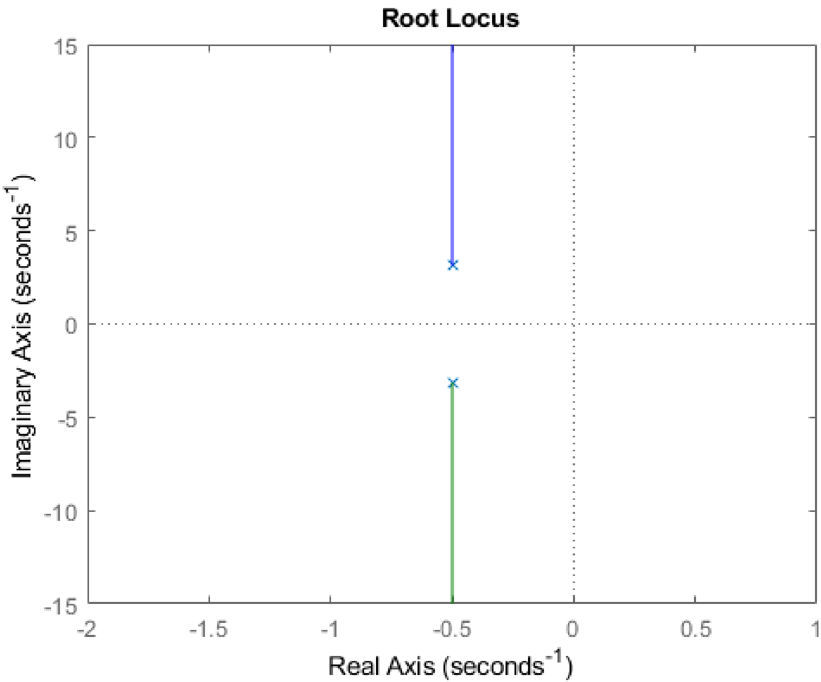
The controller will not be very successful because the poles are angled too close to the imaginary axis.

e.g. using $\zeta = \sigma/|p|$, for K=0, $\zeta = 0.5/\sqrt{0.5^2 + 3.12^2}$ = 0.158 less damping than we would like (e.g. $\zeta = 0.55$).

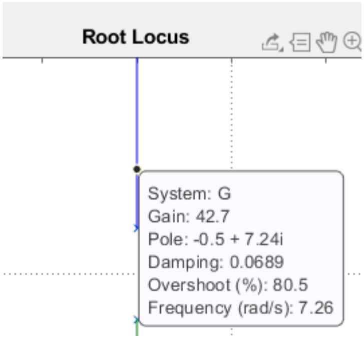


Increasing K increases $|p|$ but not σ so ζ will just decrease. Below is the open-loop step-response

Now use the Matlab **rlocus(G)**


Root Locus
15
10
5
0
-5
-10
-15
-2
-1.5
-1
-0.5
0
0.5
1
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

Note: hovering the mouse over the root locus gives useful information!



As we know, the system needs more damping, motivating a PD controller

$$C(s) = K_p + K_d s$$

equivalent to

$$F(t) = K_p e(t) + K_d \dot{e}(t)$$

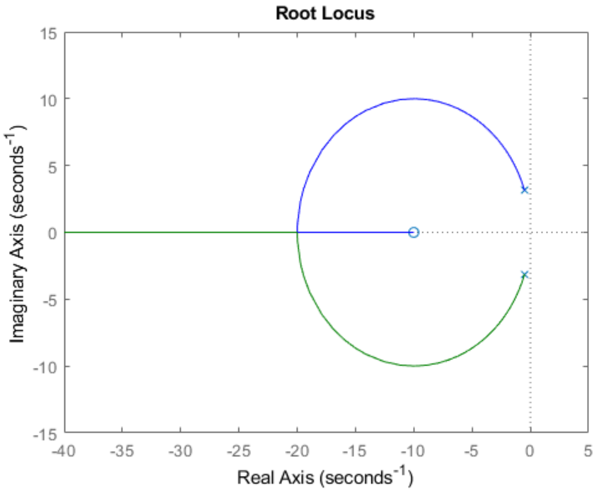
Let's assume $K_d = 0.1K_p$, so $C(s) = K(1 + 0.1s)$

For the root locus use

$$\frac{1 + 0.1s}{s^2 + s + 10}$$

The addition of a zero at -10 makes a huge difference!

The root locus is pushed over to the left and the whole shape changes. We can now find a large gain with good damping.

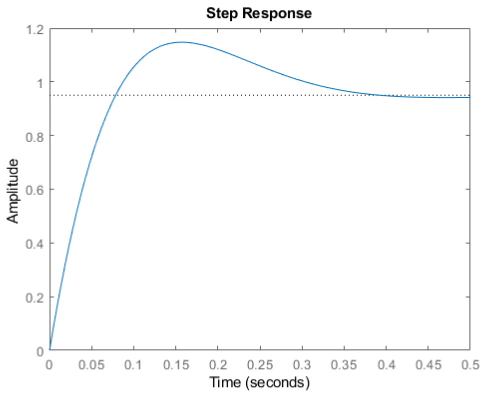


```
K=190;T=feedback(K*C*G,1);p=pole(T);
step(T)
```

```
p: 2x1 complex double =

 -10.0000 +10.0000i
 -10.0000 -10.0000i
```

For this gain, $\theta = 45°$ and $\zeta = 0.707$



Overshoot is a little higher than predicted just from the poles (why?)

Because of the high gain the steady-state error is small.

Root locus gives us confidence in the controller design and allows us to set a suitable gain –better than trial and error simulation!

Example 3 (using Matlab). Given the plant transfer function

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

a) Find the open-loop poles and zeros and decide if the plant is stable
b) Plot the open-loop step response to confirm
c) Plot the root locus assuming a proportional feedback controller. Find the gain at which the closed-loop system becomes unstable. Why is proportional control not a good method for controlling this plant?
d) Decide on two ‘rules’ of how root locus plots ‘behave’:
(i) where do they start? (ii) where do they end? Also, what symmetry do you always see?

Tutorials Problems

For the system of Example 3, using Matlab …

1. Assume a PD control of the form $K(1 + 0.1s)$ and find the new root locus.
2. Find the value of K which gives a damping ratio 0.55 on the complex branches of the plot
3. For the value of K found in question 2, find the step response and comment on the result.
4. How can you adjust the gain to reduce the overshoot in the closed-loop step response? Test out your idea.

In all the examples, the root locus **starts at open-loop poles and either go to infinity or end at open loop poles**.

We can also check that points on the root locus only sit on the real axis when there are an **odd number of real poles/zeros to its right**.

We only <u>observed</u> these properties so far, but it is essential to see why they are always true. To do so, we should recall some **basics of complex numbers**: complex numbers as vectors, products and quotients, argument and modulus, complex conjugates, roots of s, and especially the roots of -1.

Gain and phase for points on the root locus

On the root locus

$$KG(s) = -1$$

So if we know test point s_0 is on the root locus, the required gain can be found from taking the modulus $K = 1/|G(s_0)|$.

The angle (argument) is really useful, since it eliminates K. Then

$$\angle G = \arg(G) = \pi \ \ (\text{or } 3\pi, \text{or } 5\pi, \text{or } -\pi \text{ etc.})$$

i.e.

$$\angle G = \pi \ \ (+2k\pi)$$

for any integer k, $k = 0, \pm 1, \pm 2, \ldots$

If G is written in zpk form, e.g.

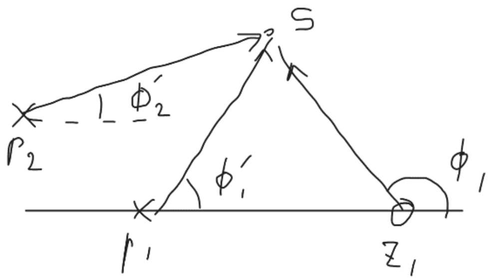
$$G(s) = k\frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)(s - p_3)}$$

then for any test point on the root locus we get the **angle sum rule**:

$$\angle G = \phi_1 + \phi_2 - \phi_1' - \phi_2' - \phi_3' = \pi \qquad (+2k\pi)$$

Here $\phi_1 = \arg(s - z_1)$ is the angle of the 'vector' from z_1 to s etc.

Use ϕ for zeros and ϕ' for poles.



In general there are m zeros and n poles

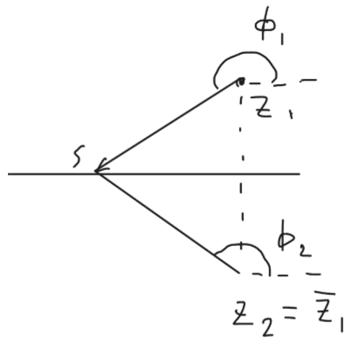
$$G(s) = k\,\frac{(s-z_1)(s-z_2)\,\dots(s-z_m)}{(s-p_1)(s-p_2)\,\dots\,(s-p_n)}$$

Test points on the real axis

It the test point s is on the real axis then complex poles and zeros have no effect on the angle sum, e.g. in the figure its obvious that $\phi_1+\phi_2=360^\circ=2\pi$, or, to be more mathematical

$$(s-z_1)=re^{j\phi_1}$$

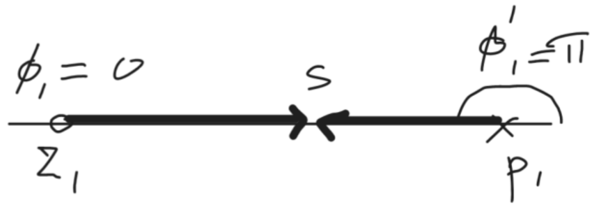
$$(s-\bar{z}_1)=(\bar{s}-\bar{z}_1)=re^{-j\phi_1}$$



and the arguments sum to zero. So complex poles and zeros can be ignored in the angle sum.

For poles and zeros on the real axis, **those to the left** of the test point are zero, so again can be ignored.

Those to the right each contribute π to the angle sum …



Thus only real poles and zeros to the right of the test point ‘get to vote’ and must sum to π or 3π etc. This confirms what we saw in the examples:

real segments of the root locus have odd numbers of poles + zeros to the right.

Start points ($K = 0$) **and end points** ($K \to \infty$). Writing

$$G(s) = \frac{N(s)}{D(s)}$$

the characteristic equation of the closed-loop system, with constant gain included, is

$$K\frac{N(s)}{D(s)} = -1$$

or

$$KN(s) + D(s) = 0$$

The as $K \to 0$

$$D(s) = 0 \text{ [pole of G]}$$

and the root locus starts at an open-loop pole.

For $K \to \infty$ it helps to re-write the characteristic equation as

$$N(s) + K^{-1}D(s) = 0$$

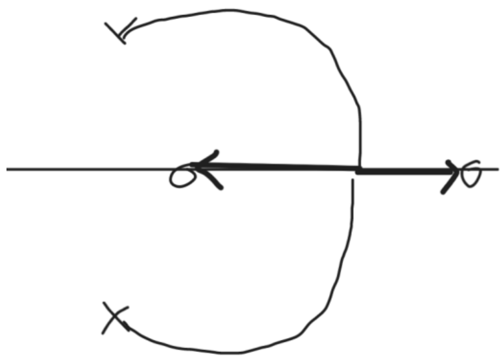
For very large K this means

$$N(s) = 0 \text{ [zero of G]}$$

Simply knowing start points, end points and the real axis rule is often enough to do a rough sketch of the root locus. Another simple feature is **symmetry about the real axis**.

This happens because the coefficients in the characteristic equation are real. Then, if a point s satisfies the characteristic equation $KG(s) = -1$, so does its complex conjugate, $KG(\bar{s}) = -1$.

This means the root locus is symmetric about the real axis.



One more really important feature is to do with asymptotes – when the root locus goes off towards infinity with increasing K. We have seen these a few times already.

Example 4 – Find the asymptotes as $K \to \infty$ for the transfer function

$$G(s) = \frac{s+1}{(s+2)(s+3)(s+4)}$$

There seem to be three poles but only one zero.

In fact there are two 'zeros at infinity': G tends to zero as $|s|$ becomes large). For large values of s

$$G(s) \cong \frac{s}{s^3} = \frac{1}{s^2}$$

Write $s = re^{j\phi}$

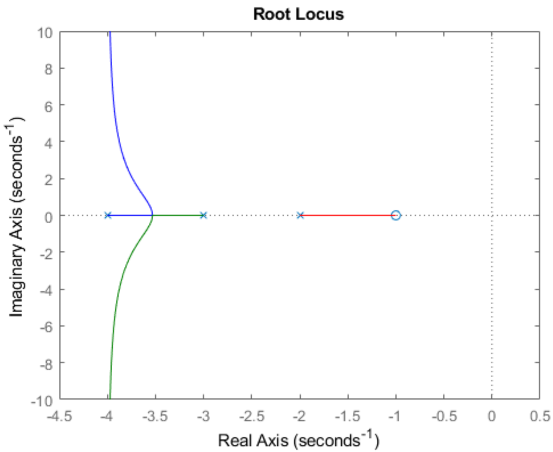
$$\frac{1}{s^2} = r^{-2}e^{-2j\phi}$$

So for large values of $|s|$ on the asymptote, the argument is -2ϕ

Since the argument must equal π on the root locus

$$-2\phi = \pi\ (\pm 2\pi, \pm 4\pi, \dots)$$

There are two different solutions, $\phi = \pi/2$ or $\phi = -\pi/2$.



We have the asymptote angles correct but it's also useful to find where the asymptotes cross the real axis (at $s = a$)

Recall the denominator order = n, the numerator order = m and we normally have $n \geq m$

The general rule is

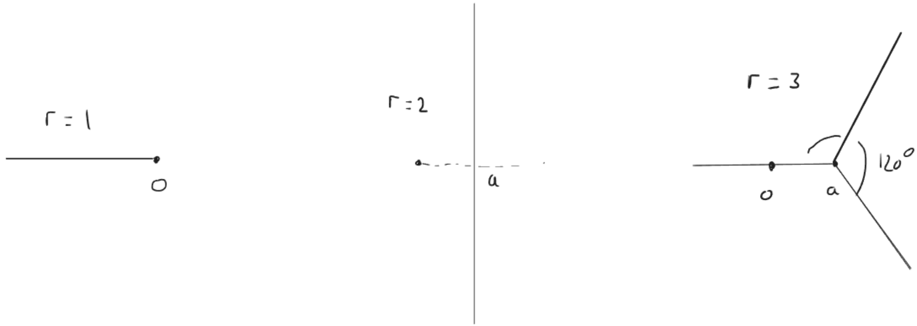
$$a = \frac{\Sigma p_i - \Sigma z_i}{n - m}$$

and $n - m$ is the number of asymptotes. In this example

$$a = \frac{(-2 - 3 - 4) - (-1)}{2} = -\frac{8}{2} = -4$$

This agrees with the plot. In case there are no zeros, the asymptotes meet at the centroid of the open-loop poles.

And if there are $r = n - m$ asymptotes the same logic shows the angles will be at $(\pi + 2\pi k)/r$.



Summary (so far for root locus)

- root locus is a graphical way to understand the effect of changing the overall loop gain
- it's helpful to understand some basic 'rules' that explain a root locus plots – e.g. the effect of adding a zero to the controller
- root locus plot helps us choose controller gains
- Matlab is useful, via **rlocus** (or **rltool**)
- there is always symmetry about the real axis
- There are three 'rules' found so far, for
  (1) start and end points,
  (2) real axis
  (3) asymptotes.

Tutorial Questions (continued)

5. Use Matlab to draw the root locus plot for

$$G(s) = \frac{(s+2)(s+3)}{s(s+1+j)(s+1-j)(s+4)}$$

   and check the various rules mentioned above.

6. Using the rules you know already, sketch the root locus of the plant

$$G(s) = \frac{1}{s(s+1)(s+2)}$$

   (note that there are three zeros at infinity so the asymptote angles are found from $(\pi + 2\pi k)/3$)

7. For the previous question, use the plot to find the gain at which the system just becomes unstable, and check this gain by plotting the closed-loop step response. From the plot, estimate the frequency of the sustained oscillation.

8. The condition for crossing the imaginary axis is

$$1 + KG(j\omega) = 0$$

   This gives two equations (real and imaginary parts) which you can solve for ω and K. Check these agree with the values found from the root locus plot and the step response.
9. Find the root-locus asymptotes for the following plant

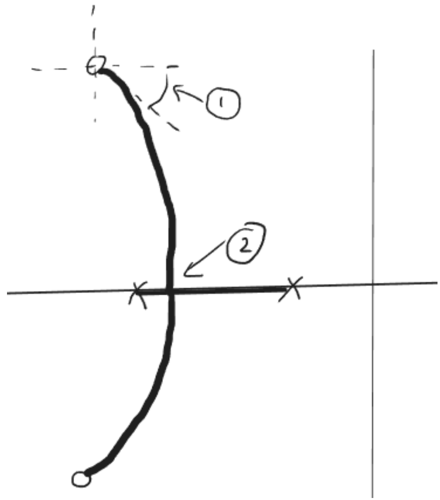
$$G(s) = \frac{1}{s(s+2)(s+1+j)(s+1+j)}$$

   and then check your result by plotting the root locus in Matlab.

Further methods – 'forensic' techniques giving detailed information

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

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

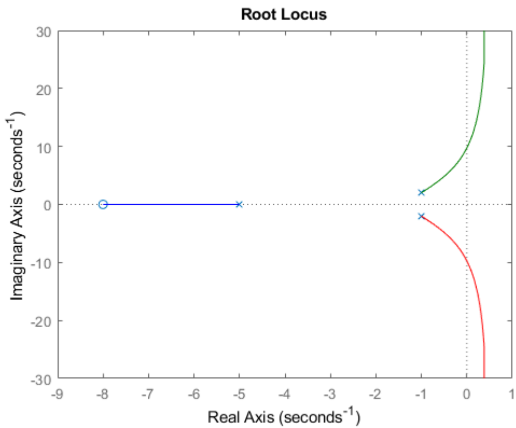


Angle of departure

For the transfer function of Example 3

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

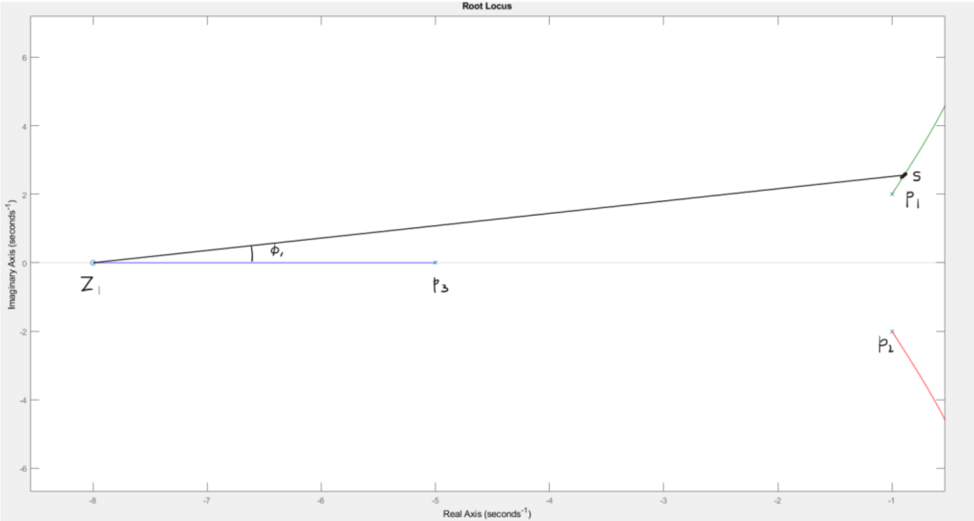
and with proportional control we obtained the root locus shown



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

Apply the angle 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) clearly $\phi'_2 = \pi/2$ (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

$$\phi_1' = -280.6^\circ$$

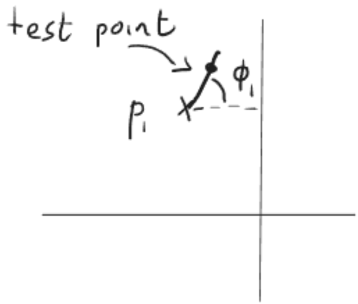
or adding 360°

$$\phi_1' = 79.4^\circ$$

This is the phase angle for departing

$p_1 = -1 + 2j$

This may not look correct on the previous root locus plot, but the axes don't have equal scales. Use **axis 'equal'** in Matlab to avoid this. problem.



Breakaway points (break out from the real axis)

Consider (again) the second order system with PD controller from Example 2.

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

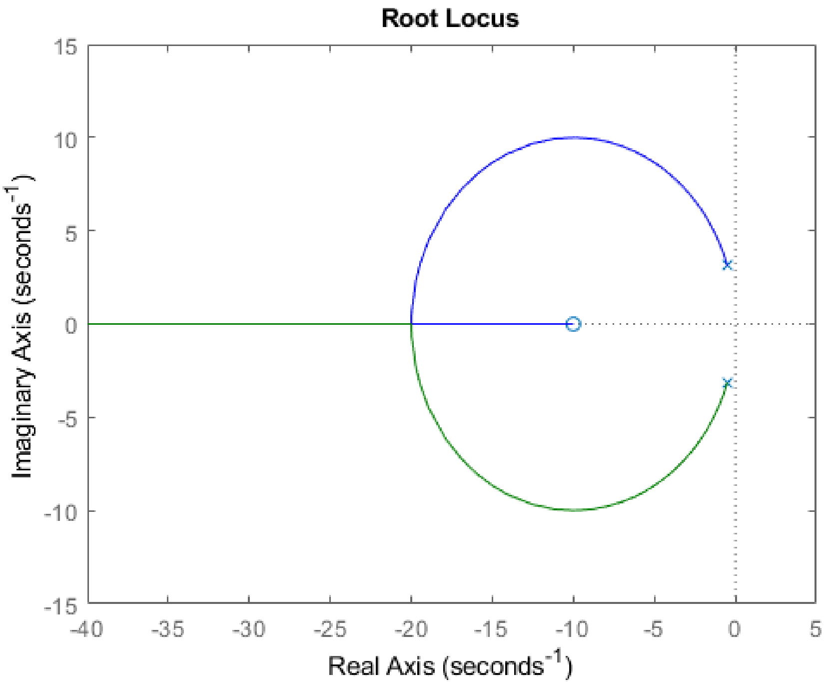
The 'rule' for a breakaway point is

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

So

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

This is easily solved to give $s = 0$ or $s = -20$; clearly it is the second solution we need. The root locus confirms this …


Root Locus
15
10
5
0
-5
-10
-15
-40
-35
-30
-25
-20
-15
-10
-5
0
5
Real Axis (seconds^-1)
Imaginary Axis (seconds^-1)

Summary of the Root Locus 'rules'

- RL branches start at poles of $KG(s)$ and end 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

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

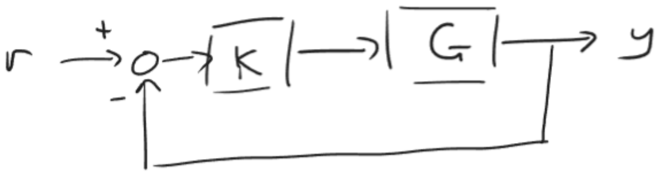
- angles of departure are found from the angle sum

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

- breakaway points satisfy $N'(s)D(s) = N(s)D'(s)$ and depart the real axis at $\pm 90°$.

Example 4 root locus when K is not a simple gain

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



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

But suppose we want to vary a different parameter - we approach the problem by **rewriting the closed-loop characteristic equation** to mimic the standard case:

$$D(s) + KN(s) = 0$$

In Example 2 we had

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

and introduced a PD controller. Let's now vary the derivative gain as an independent parameter:

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

Now the proportional gain is fixed.

The characteristic equation becomes

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

so

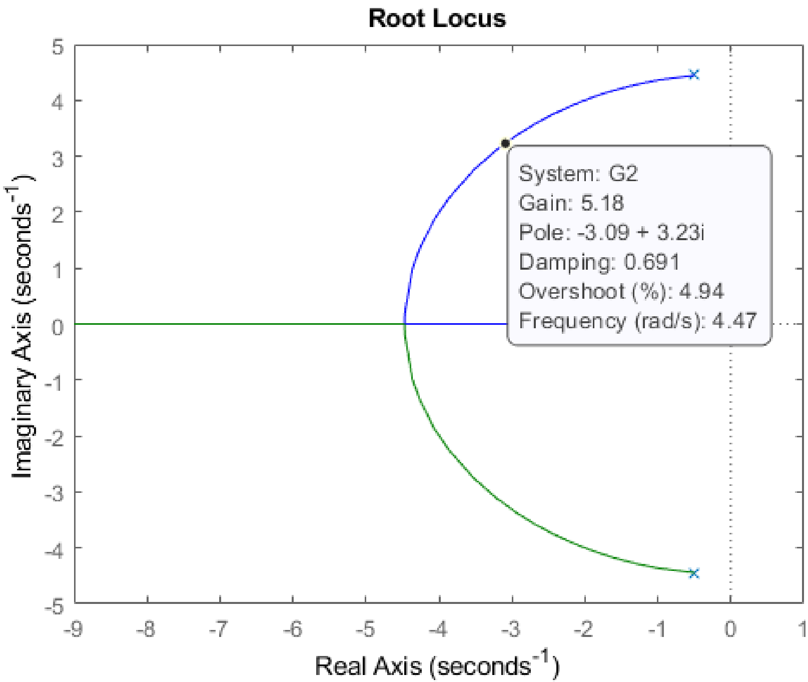
$$s^2 + s + 10 + 10 + Ks = 0$$

$$K\,s + (s^2 + s + 20) = 0$$

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

```
%% Example 4
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 in doing this, but it’s needed after when checking the step response


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)
```