Lecture 6 The Roots of a Parameterized Polynomial (i.e. a Root Locus)

QUESTION: What is a parameterized polynomial?

ANSWER: It is a polynomial that includes a parameter that can be varied.

Example 1. Consider $p(s) = s^3 + 2s^2 + 25s + K$.

Write a *Matlab* code that will plot the roots of $p(s)$ for K=0:0.1:100. Then use the data cursor to find the values of the purely imaginary roots when they hit the imaginary axis. Finally, substitute one of those purely imaginary values into $p(s)$, and solve it for the corresponding K value.

<u>*Solution*</u>: [See code @ 1(e).]

$$p(s = i5) = (i5)^3 + 2(i5)^2 + 25(i5) + K = 0 \Rightarrow \quad K = 50$$

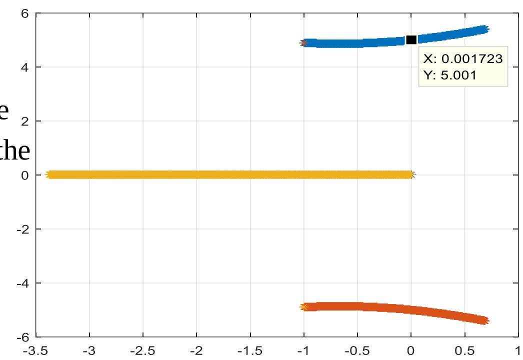


Figure 1 Roots of $p(s)$ as a function of K.

```
%PROBLEM 1(e):
K=0:.1:100;   %LET K VARY FROM 0 TO 100
n=length(K);
rp = zeros(n,3);
for k=1:n
    rp(k,:)=roots([1 2 25 K(k)]); %FIND THE ROOTS FOR A GIVEN K
end
RE=real(rp);  %BY DEFAULT, THEY ARE ASSUMED TO BE COMPLEX-VALUED
IM=imag(rp);
figure(1)
plot(RE,IM,'*')  %PLOT THE PARAMETERIZED COLLECTION (OR LOCUS) OF ROOTS
grid
```

Note that the value for K that corresponds to purely imaginary roots is critical, since for that value the system is marginally (or neutrally) stable. This is because the polynomial of interest is the system characteristic polynomial. From the above plot, we see that the system will be unstable for all values of K greater than $K_{cr} = 50$. □

Example 2. Consider $G_p(s) = \dfrac{1}{s^2 + 2s + 25}$ and $G_c(s) = \dfrac{K(s+5)}{s}$ in a unity feedback configuration. Then the closed loop transfer function is $W(s) \overset{\Delta}{=} \dfrac{Y(s)}{Y_r(s)} = \dfrac{G_p(s)G_c(s)}{1 + G_p(s)G_c(s)G_s(s)}$. The closed loop system poles are the s-values that make the denominator zero:

$$1 + G_p(s)G_c(s)G_s(s) = 0 = 1 + \frac{K(s+5)}{s(s^2 + 2s + 25)} \quad \Rightarrow \quad p(s) \overset{\Delta}{=} s^3 + 2s^2 + (K+25)s + 5K = 0 \cdot$$

Rather than writing our own code to find the root locus for $p(s)$, we will use Matlab's 'rlocus' command:

Gp=tf(1,[1 2 25]); Gc=tf([1 5],[1 0]); rlocus(Gc*Gp)

Note that the rlocus <u>argument</u> is the *open loop transfer function* $G_p(s)G_c(s)G_s(s)$ with K set to 1.0. With this argument, the 'rlocus' code computes the s-values that satisfy $1 + KG_p(s)G_c(s)G_s(s) = 0$ for a range of K-values. □

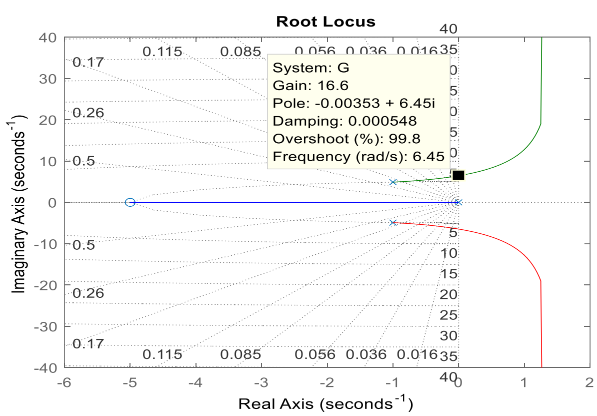


Figure 2 Closed loop root locus.

The root locus is a useful tool for the method of controller design via ***pole-placement***. The designer first determines the desired closed loop specifications. These are then translated into related closed loop pole locations. Having these, the designer then uses the root locus ***angle criterion*** to arrive at the controller pole/zero locations. Finally, the required root locus gain, K, is determined using the root locus ***magnitude criterion***. We will develop these criteria in the next lecture.

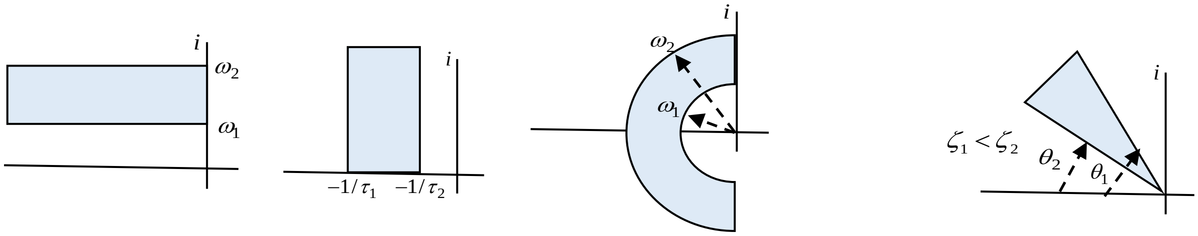
We will now review the geometry of complex poles and their relation to typical specifications.

GEOMETRY 101 ☺:

Consider the polynomial $p(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$ having complex-conjugate roots. From the quadratic formula, the roots of $p(s)$ are:

$$s_{1.2} = -\zeta\omega_n \pm i\,\omega_d \quad where \quad \omega_d \overset{\Delta}{=} \omega_n\sqrt{1-\zeta^2} \text{ for } 0 \le \zeta < 1. \tag{1}$$

The plots below illustrate the geometry associated with (1).



(i) $\omega_1 \le \omega_d \le \omega_2$ **(ii)** $\tau_1 \le \tau \le \tau_2$ **(iii)** $\omega_1 \le \omega_n \le \omega_2$ **(iv)** $\zeta_1 \le \zeta \le \zeta_2$

Figure 3. The plots show areas of constant (i) ω_d , (ii) τ , (iii) ζ , and (iv) ω_n .

We will now review how the shaded regions in Figure 1 are arrived at in relation to $s_1 = -\zeta\omega_n + i\,\omega_d$.

(i) $\omega_1 \le \omega_d \le \omega_2$: Since ω_d is the imaginary part of s_1 it follows that lines of constant ω_d are horizontal.

(ii) $\tau_1 \le \tau \le \tau_2$: Since $\tau = 1/\zeta\omega_n$ and $-\zeta\omega_n$ is the real part of s_1 it follows that lines of constant τ are vertical.

(iii) $\omega_1 \le \omega_n \le \omega_2$: Since $|s_1| = \sqrt{(-\zeta\omega_n)^2 + \omega_d^2} = \omega_n$, we have $s_{1.2} = -\zeta\omega_n \pm i\,\omega_d \quad where \quad \omega_d \overset{\Delta}{=} \omega_n\sqrt{1-\zeta^2}$ it follows that a line of constant ω_n is a circle of radius ω_n .

(iv) $\zeta_1 \le \zeta \le \zeta_2$: From (ii) and (iii) we have $\cos\theta = \zeta\omega_n / \omega_n = \zeta$, or $\zeta = \cos^{-1}\theta$. Hence, a line of constant ζ is radial.

Example 4. Suppose that we require that system poles adhere to specifications: (S1) $0.7 < \zeta < 0.9$ and (S2) $0.1 < \tau < 0.2$. The pole region associated with these specifications is shown at right. □

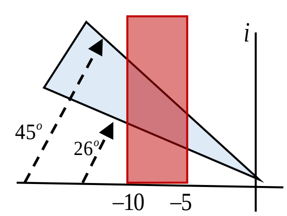


Figure 4. CL pole design region.

Example 5. ***A Simple Example of Pole-Placement Design***

For a plant $G_p(s) = \dfrac{2}{s(s+1)}$, design a controller that will satisfy (S1) $\zeta = 0.707 = 1/\sqrt{2}$ and (S2) $\tau = 0.5$.

(a) Since $\zeta\omega_n = 1/\tau = 2 \Rightarrow \omega_n = 2\sqrt{2} = 2.828$, we have $\omega_d = \omega_n\sqrt{1-\zeta^2} = 2$. Hence, the required closed loop poles are $s_{1,2} = -2 \pm i2$.

(b) For $G_c(s) = K$ use a root locus plot to show that this controller cannot place the CL poles $s_{1,2} = -2 \pm i2$.

Gp=tf(2,[1 1 0]); rlocus(Gp) give the plot at right.

Clearly, the vertical locus never comes close to $s_{1,2} = -2 \pm i2$.

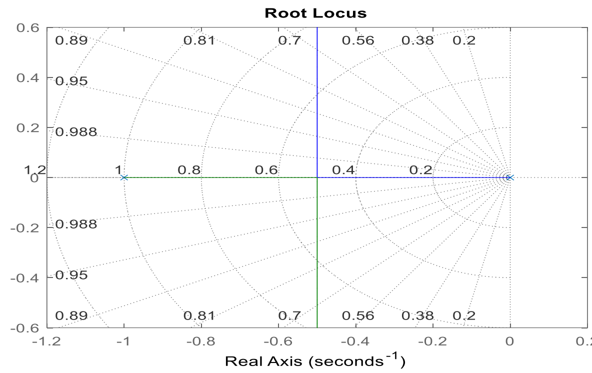


Figure 5. Root locus for $KG_p(s) = \dfrac{2K}{s(s+1)}$.

The root locus angle criterion states that $s_1 = -2 + i2$ will be on the CL root locus if the following condition is satisfied:

$$\sum_k \phi_z(k) - \sum_k \phi_p(k) = -180^o . \qquad (2)$$

where $\phi_z(k)$ is the angle from the kth open loop zero to s_1, and where $\phi_p(k)$ is the angle from the kth open loop zero to s_1.

(c) Use (2) to prove that no controller of the form $G_c(s) = K_p$ can result in $s_1 = -2 + i2$.

$$\sum_k \phi_z(k) - \sum_k \phi_p(k) = 0^o - (135^o + 116.56^o) = 251.56^o = -71.56^o - 180^o \cdot$$

Hence, the angle criterion id ‘off’ by -71.56^o. So, not only have we proved that the controller $G_c(s) = K_p$ will not work. We have also determined how much angle the controller must <u>add</u> to the root locus.

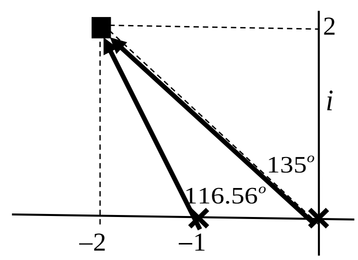


Figure 6. Root locus angles for $G_c(s) = K_p$.

(d) In view of (b), consider the PD controller $G_c(s) = K(s+\alpha)$. From (b) we must have $\phi_{z_1} = 71.56^o$. Hence, $\phi_{z_1} \cong \tan 71^o = 2/(\alpha - 2)$ results in $\alpha = \mathbf{2.67}$. Hence:

$$G_c(s) = K(s + 2.67).$$

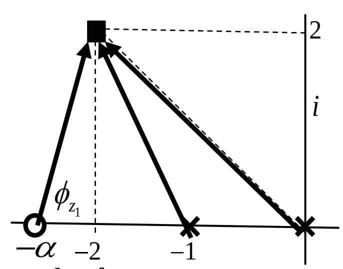


Figure 7. Root locus angles for $G_c(s) = K(s + 2.67)$.

Before determining the required value for K , let’s obtain the root locus plot.

Gc=tf([1 2.67],1); rlocus(Gc*Gp) grid

The root locus plot at right confirms that for $K=1.5$ the closed loop transfer function will have poles at $s_{1,2}=-2\pm i2$.

The advantage of obtaining this plot is that we can avoid the use of the root locus magnitude criterion by simply using the data cursor. ☺

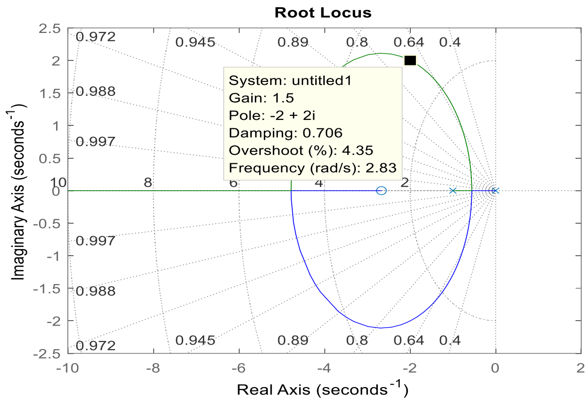


Figure 8. Root locus for $G_c(s)=K(s+2.67)$.

(e) Use the 'feedback' command to obtain the closed loop transfer function $W(s)=\dfrac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}$.

Gc=1.5*Gc; W=feedback(Gc*Gp,1) W = (3 s + 8.01)/(s^2 + 4 s + 8.01)

(f) Use the 'zpkdata' command to verify that the specifications have been satisfied.

[z,p,k]=zpkdata(W,'v')

z =-2.6700 p = **-2.0000 +/- 2.0025i** k =3

(g) Overlay the step responses for $W(s)=\dfrac{3s+8}{s^2+4s+8}$ and $W^*(s)=\dfrac{8}{s^2+4s+8}$. Then comment.

[n,d]=tfdata(W,'v') n = 0 3.0 8.01 d = 1.00 4.00 8.01
WW=tf(n(3),d) WW = 8.01 / (s^2 + 4 s + 8.01)
step(W)
hold on
step(WW)
title('Step Responses')
legend('W','WW')
grid

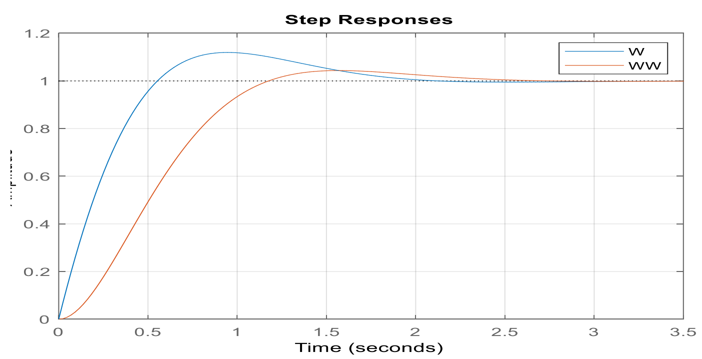


Figure 9. Step responses for $W(s)$ and $W^*(s)$.

Comment: A main reason for choosing $\zeta=0.707=1/\sqrt{2}$ is that the associated poles will yield minimal overshoot. What we see in Figure 9 is that the actual overshoot is notably greater than what was expected due to the zero in $W(s)$. This illustrates a major limitation of the controller pole-placement design method. It specifies the closed loop poles, but ignores closed loop zeros. □