

## Chapter 10: Root Locus Analysis

August 4, 2015

# Outline

- 1 Concept of Root Locus
- 2 How To Sketch the Root Locus
  - General Approach
  - Rules of thumb for sketching the root locus
- 3 Design criteria
- 4 Root Locus and MATLAB
  - Root Locus
  - SISOTOOL

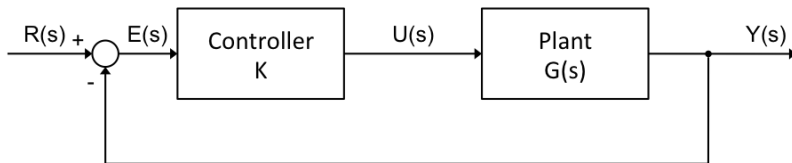
# Concept of the method

In this chapter the Root Locus Method is presented. This technique shows how changes in the system's feedback characteristics and other parameters influence the pole locations. The method permits us to plot the locus of the closed-loop pole locations in the  $s$ -plane as a parameter varies, which will produce a root locus (hence the name of the technique).

It is very important to understand the background of root loci: how they take their shape, why they are useful, ... Therefore, we start this chapter by explaining the concept of the Root Locus Technique. Next we explain how to sketch a root locus. Finally we will give some examples in MATLAB.

## Concept of the technique

We begin with the basic feedback system, shown in the figure below:



The closed-loop transfer function is:

$$H(s) = \frac{Y(s)}{R(s)} = \frac{KG(s)}{1+KG(s)}.$$

## Concept of the technique

Looking at this transfer function, we can conclude that the closed-loop roots depend on the gain  $K$ . We can now plot the locus of all possible roots of the characteristic equation:

$$1 + KG(s) = 0 \quad (1)$$

as  $K$  varies from 0 to  $\infty$ . This results in a graph which can help us in selecting the best value of  $K$ .

Furthermore, by studying the effects of additional poles and zeros, we can determine the consequences of additional dynamics in the loop. We can also extend this technique to examine the effect of other plant-parameter changes in order to achieve the best overall control design.

# Root Locus

## Definition I

The root locus is the set of values of  $s$  for which  $1 + KG(s) = 0$  is satisfied as the real parameter  $K$  varies from 0 to  $+\infty$ . Often  $G(s)$  is the open-loop transfer function of a system; in this case, roots on the locus are closed-loop poles of that system.

## Use of root locus

The root locus method provides a tool not only for selecting the gain but for designing the dynamic compensations as well.

## Concept of the technique

For the further derivation, we assume that the system's open-loop transfer function  $G(s)$  is a rational function whose numerator is  $b(s)$  and whose denominator is  $a(s)$ .  $b(s)$  is a monic polynomial of degree  $m$  and  $a(s)$  is a monic polynomial of degree  $n$ .

### All positive values of $K$

As  $K \rightarrow 0$ , the closed-loop poles of system are the roots of  $a(s) = 0$  or the open-loop poles. As  $K \rightarrow \infty$ , the closed-loop poles of the system are the roots of  $b(s) = 0$  or the open-loop zeros.

## Concept of the technique

If  $H(s)$  has more poles than zeros ( $m < n$ ) we say that  $H(s)$  has zeros at infinity. In this case, the limit of  $H(s)$  as  $s \rightarrow \infty$  is zero. The number of zeros at infinity is  $n - m$ , the number of poles minus the number of zeros, and is the number of branches of the root locus that go to infinity (asymptotes).

### Open-loop vs Closed-loop

The root-locus method can be thought of as a method for inferring properties of the closed-loop system given the open-loop transfer function  $KG(s)$ .



## Concept of the technique

### Root-Locus form

We can express Eq. (1) in different equivalent ways:

$$1 + KG(s) = 0,$$

$$1 + K \frac{b(s)}{a(s)} = 0,$$

$$a(s) + Kb(s) = 0,$$

$$G(s) = -\frac{1}{K}.$$

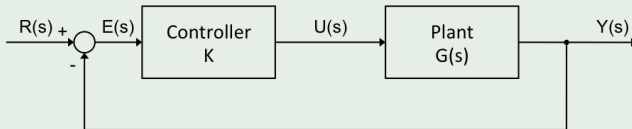
These different presentation are called the root-locus forms.

## Example

**Problem:** a normalized transfer function of a DC motor is:

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

Solve for the locus of roots with respect to the proportional gain  $K$  of the closed-loop system created by feeding back the output as shown in the figure:



Solve by using direct calculations of the root locations.

**Solution:** in terms of our notation

$$\begin{array}{lll} m = 0, & K_G = 1, & b(s) = 1, \\ n = 2, & K_A = K, & a(s) = s(s + 1). \end{array}$$

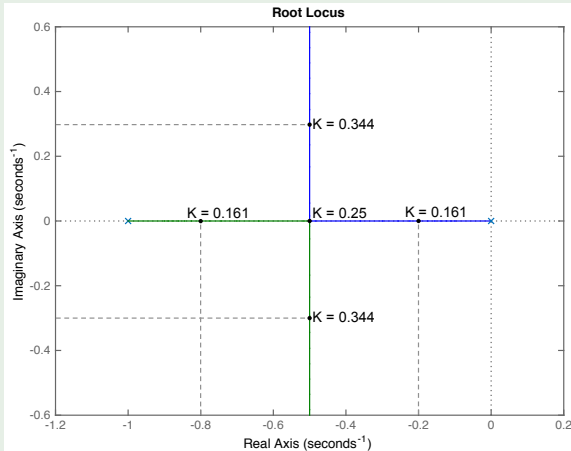
We can use the root-locus form  $a(s) + Kb(s) = 0$  to obtain a quadratic equation of which the roots will produce a graph. In this case, the quadratic equation is:

$$s^2 + s + K = 0 \tag{3}$$

has roots:

$$r_1, r_2 = -\frac{1}{2} \pm \frac{\sqrt{1 - 4K}}{2}. \tag{4}$$

The root locus is shown below



- For  $0 \leq K \leq \frac{1}{4}$ , the roots are real between  $-1$  and  $0$ ;
- At  $K = \frac{1}{4}$  there are two roots at  $-\frac{1}{2}$ ;
- For  $K > \frac{1}{4}$  the roots become complex, with a real part of  $-\frac{1}{2}$  and an imaginary part that increases essentially in proportion to the square root of  $K$ .

# Outline

- 1 Concept of Root Locus
- 2 How To Sketch the Root Locus
  - General Approach
  - Rules of thumb for sketching the root locus
- 3 Design criteria
- 4 Root Locus and MATLAB
  - Root Locus
  - SISOTOOL

# Outline

- 1 Concept of Root Locus
- 2 How To Sketch the Root Locus
  - General Approach
  - Rules of thumb for sketching the root locus
- 3 Design criteria
- 4 Root Locus and MATLAB
  - Root Locus
  - SISOTOOL

## General approach

Now look at the equation:  $G(s) = -\frac{1}{K}$ . If  $K$  is to be real and positive,  $G(s)$  must be real and negative. This means that if we arrange  $G(s)$  in polar form as magnitude and phase,  $G(s)$  must have the opposite phase of  $K$  in order to satisfy the equation above. We can thus define a root locus in terms of this phase condition:

### Definition II

The root locus of  $G(s)$  is the set of points in the s-plane where the phase of  $G(s)$  is  $180^\circ$ .



## General approach

Since the phase remains unchanged if an integral multiple of  $360^\circ$  is added, we can express Definition II as:  $\angle G(s) = 180^\circ + 360^\circ I$ , where  $I$  is any integer. While it is very difficult to solve a high-order polynomial, computing the phase is relatively easy.

The usual case is when  $K$  is real and positive; we call this case the **positive or  $180^\circ$  locus**. When  $K$  is real and negative,  $G(s)$  must be real and positive of  $s$  to be on the locus. Therefore, the phase of  $G(s)$  must be  $0^\circ$ . This special case is called a **negative or  $0^\circ$  locus**.

Although measuring the phase is easy, measuring the phase at every point in the  $s$ -plane is hardly practical. It would be better if there would exist some general guidelines we can use for determining where the root locus is situated.

## Guidelines in practice

We will now sketch the root locus of a system with transfer function:

$$G(s) = \frac{1}{s[(s+4)^2 + 16]} \quad (5)$$

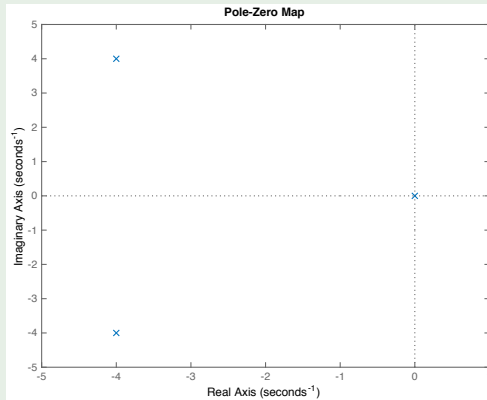
Using Definition II, we can check whether a point  $s_0$  lies on the root locus for some value of  $K$  by checking if the following expression is valid:

$$\angle 1 - \angle s_0 - \angle [(s_0 + 4)^2 + 16] = 180^\circ + 360^\circ l \quad (6)$$

but as already mentioned, it is not practical to do this for every point. So we will now use the general guidelines to sketch the root locus.

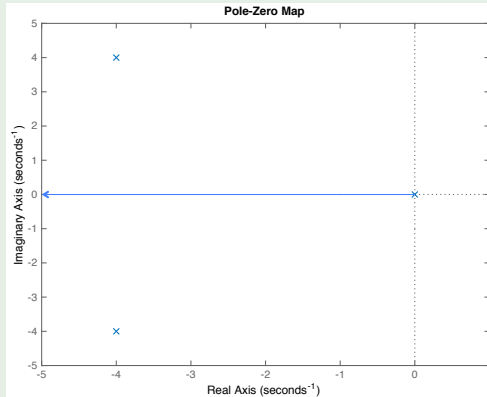
## STEP 1: the open-loop $\times$ 's and $\circ$ 's

Calculate and plot the open-loop poles and zero's:



## STEP 2: locus on real axis

The portion of the real axis to the left of an odd number (counted from the right) of open loop poles and zeros are part of the locus:



## STEP 3: asymptotes for large values of K

As K approaches  $\infty$ , the equation  $G(s) = -\frac{1}{K}$  can be satisfied only if  $G(s) = 0$  where  $G(s) = \frac{b(s)}{a(s)}$ .

We can now substitute this in  $1 + KG(s) = 0$ , resulting in the following equation:

$$1 + K \frac{b(s)}{a(s)} = 1 + K \frac{s^m + b_1 s^{m-1} + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} = 0 \quad (7)$$

Since  $n < m$ ,  $G(s) = 0$  can occur when:

- $b(s) = 0$ ;
- $s \rightarrow \infty$ .

## STEP 3: asymptotes for large values of $K$

We now look closer to the second condition:  $G(s) = 0$  when  $s \rightarrow \infty$ .

For very large values of  $s$ , the highest-order power of  $s$  in Eq.(8) predominates. We can divide  $a(s)$  by  $b(s)$  and match the dominant two terms (highest powers in  $s$ ) to the expansion  $(s - \alpha)^{n-m}$ , resulting in the following approximation:

$$1 + K \frac{1}{(s - \alpha)^{n-m}} \quad (8)$$

We need to find the locus for the asymptotic system (Eq.(8)) and  $\alpha$ .

## STEP 3: asymptotes for large values of $K$

To find the locus, we choose  $s_0 = Re^{j\phi}$  for some large value of  $R$  and some variable  $\phi$ .

Since all poles of Eq.(9) are in the same place, the angle of its transfer function is  $180^\circ$  if all  $n - m$  angles ( $\phi_l$ ) sum to  $180^\circ$ . Therefore,  $\phi_l$  is given by:

$$\phi_l = \frac{180^\circ + 360^\circ l}{n - m}, l = 1, 2, \dots, n - m \quad (9)$$

For our system:  $n - m = 3$ , thus  $\phi_1, \phi_2, \phi_3 = 60^\circ, 180^\circ, 300^\circ$ . All the lines of the asymptotic locus come from  $s_0 = \alpha$ .

## STEP 3: asymptotes for large values of K

To determine  $\alpha$  we use a simple property of polynomials. Suppose we have a monic polynomial with coefficients  $a_i$  and roots  $p_i$  and we equate the polynomial form with the factored form:

$$s^n + a_1 s^{n-1} + a_2 s^{n-2} + \dots + a_n = (s - p_1)(s - p_2) \dots (s - p_n).$$

If we multiply out the factors on the right side of this equation, we can see that the coefficient of  $s^{n-1}$  is  $-p_1 - p_2 - \dots - p_n$ . On the left side of the equation we see that this term is  $a_1$ .

Consequently  $a_1 = -\sum p_i$ , the coefficient of the second-highest term in a monic polynomial is the negative sum of its roots (the poles of  $G(s)$ ). The same can be concluded for  $b_1 = -\sum z_i$ .



## STEP 3: asymptotes for large values of $K$

Applying these results in the closed-loop characteristic polynomial, leads to:

$$s^n + a_1s^{n-1} + \dots + a_n + K(s^m + b_1s^{m-1} + \dots + b_m) = 0.$$

The negative of the sum of the poles is the coefficient of  $s^{n-1}$  and is independent of  $K$  if  $m < n - 1$ . However, since this is the closed-loop characteristic equation, this coefficient is the negative of the sum of the roots of the closed-loop system  $\sum r_i$ , hence

- the center point of the roots does not change with  $K$  if  $m < n - 1$ ;
- $-\sum r_i = -\sum p_i$ .

## STEP 3: asymptotes for large values of $K$

For large values of  $K$ ,  $m$  of the roots  $r_i$  are approximately equal to the zeros  $z_i$  and  $n - m$  of the roots are from the asymptotic  $\frac{1}{(s-\alpha)^{n-m}}$  system, whose poles add up to  $(n - m)\alpha$ .

Combining these results we conclude that the sum of all the roots equals the sum of those roots that go to infinity plus the sum of those roots that go to the zeros of  $G(s)$ :

$$+\sum r_i = +(n - m)\alpha + \sum z_i = +\sum p_i.$$

Solving for  $\alpha$  we get

$$\alpha = \frac{\sum p_i - \sum z_i}{n - m}.$$

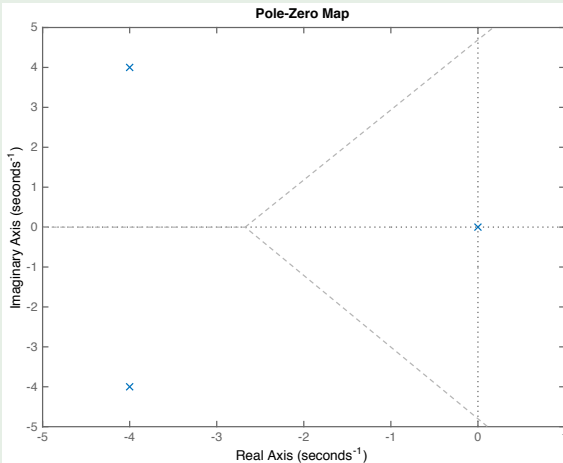
## STEP 3: asymptotes for large values of K

Complex poles and zeros always occur in complex conjugate pairs, consequently in the sums  $\sum p_i$  and  $\sum z_i$  the imaginary parts will always add to zero.

For our system:  $\alpha = \frac{-4-4+0}{3-0} = -2.67$ .

We can now use the results for  $\phi_1, \phi_2, \phi_3$  and  $\alpha$  to draw the asymptotes.

## STEP 3: asymptotes for large values of K

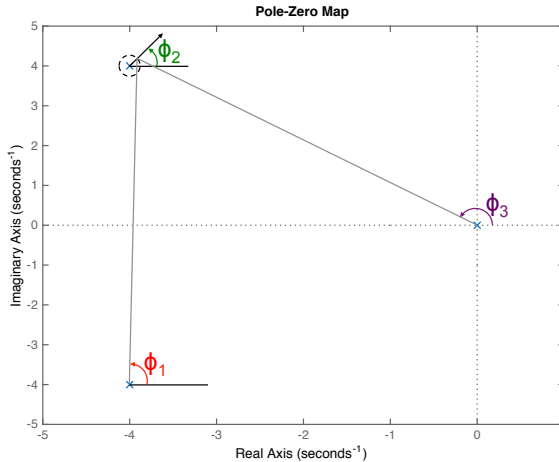


## STEP 4: departure and arrival angles

We know that the locus begins at the  $\times$ 's and that it goes to either  $\circ$ 's or to infinity along the radial asymptotic lines.

We next compute the angle by which a branch of the locus departs from one of the poles. We take a test point  $s_0$  very near pole 2 at  $-4 + 4j$  and compute the angle of  $G(s_0)$  (illustration see next slide). We select the test point close enough to pole 2 that the angles  $\phi_1$  and  $\phi_3$  to the test point can be considered the same as those angles to pole 2:  $\phi_1 = 90^\circ$  and  $\phi_3 = 135^\circ$ .

## STEP 4: departure and arrival angles



## STEP 4: departure and arrival angles

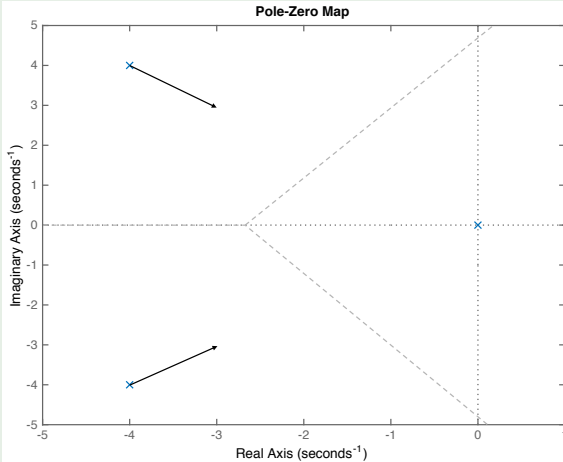
We can calculate  $\phi_2$  from the angle condition:

$$-90^\circ - \phi_2 - 135^\circ = +180^\circ + 360^\circ l$$

where  $l$  is chosen so that  $-180^\circ < \phi_2 < +180^\circ$ . If we take  $l = -1$ , then  $\phi_2 = -45^\circ$ .

By the complex conjugate symmetry of the plots, the angle of departure of the locus near pole 1 at  $-4 - 4j$  will be  $+45^\circ$ . The angle of departure from pole 3 at the origin is  $180^\circ$ .

## STEP 4: departure and arrival angles



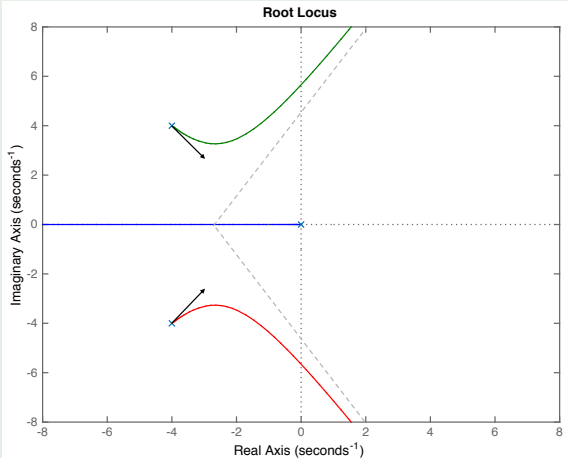


## STEP 5: location of multiple roots and their arrival and departure angles

- Two locus segments coming together at any point in the s-plane will always approach at a relative angle of  $180^\circ$  and then break away with a  $90^\circ$  change in direction;
- Three locus segments coming together will always approach at  $120^\circ$  angles that are rotated  $60^\circ$  relative to the arrival angles.

In our example, this step is not applied. But the first case (with two locus segments) can be seen in the first example from this chapter.

## STEP 6: Complete the sketch



# Outline

- 1 Concept of Root Locus
- 2 How To Sketch the Root Locus
  - General Approach
  - Rules of thumb for sketching the root locus
- 3 Design criteria
- 4 Root Locus and MATLAB
  - Root Locus
  - SISOTOOL

## Overview of the guidelines

Drawing the root locus yourself can be done by following the next steps:

- 1 Mark poles with  $\times$  and zeros with  $\circ$ ;
- 2 Draw the locus on the real axis to the left of an odd number of real poles plus zeros;
- 3 Draw the asymptotes, centered at  $\alpha$  and leaving at angles  $\phi_l$ , where:

$$n - m = \# \text{asymptotes}$$

$$\alpha = \frac{\sum p_i - \sum z_i}{n - m}$$

$$\phi_l = \frac{180^\circ + 360^\circ(l - 1)}{n - m}, l = 1, 2, \dots, n - m.$$

## Overview of the guidelines

- 4 Compute locus departure angles from the poles and arrival angles at the zeros:

$$q\phi_{dep} = \sum\psi_i - \sum\phi_i - 180^\circ - 360^\circ l,$$
$$q\psi_{arr} = \sum\phi_i - \sum\psi_i + 180^\circ + 360^\circ l,$$

where  $q$  is the multiplicity of the pole or zero and  $l$  takes on  $q$  integer values so that the angles are between  $\pm 180^\circ$ .  $\sum\phi_i$  is the sum of the angles of the (remaining) poles and  $\sum\psi_i$  is the sum of the angles of the (remaining) zeros.

## Overview of the guidelines

- 5 Use the results from the study of multiple roots to help in sketching how locus segments come together and break away: two segments come together at  $180^\circ$  and break away at  $\pm 90^\circ$ , three locus segments approach each other at relative angles of  $120^\circ$  and depart at angles rotated by  $60^\circ$ .
- 6 Complete the locus, using the facts developed in the previous steps and making reference to the illustrative loci for guidance. The locus branches start at poles and at zeros or infinity.

If further refinement is required at the stability boundary, you can also estimate the points where the root locus crosses the imaginary axis. This is done by using Routh's stability criterion. We don't discuss it in this course, but you can find an explanation for it in the additional material of this chapter.

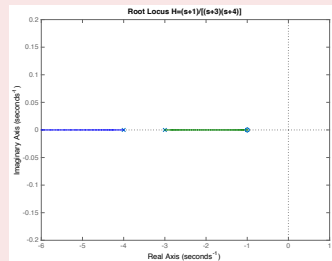
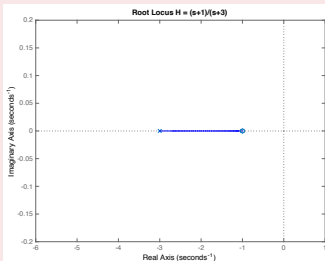
## Important to remember!

While you are sketching your root locus, keep the following things in mind:

- 1 The root locus has  $\max(\# \text{poles of } G(s), \# \text{ zeros of } G(s))$  branches;
- 2 Complex roots of  $1 + KG(s)$  (closed loop poles) always occur in conjugate pairs;
- 3 A branch will never cross over itself;
- 4 Branches leave and enter the real axis at  $90^\circ$  (this is not a 100% general rule, but it is true in most cases);

## Important to remember!

- 5 Each branch starts at an open loop pole of  $G(s)$  (for  $K = 0$ ) and ends at a closed loop zero of  $G(s)$  (for  $K \rightarrow \infty$ ). If  $\# \text{poles of } G(s) \neq \# \text{zeros of } G(s)$  the extra branches go to/come from a zero/pole in infinity.

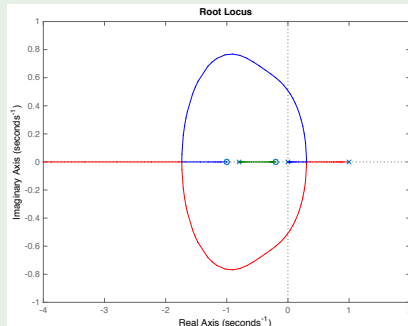




# Example 1

## Example

Try to understand why the root locus of the following system,  $H(s) = \frac{(s+0.2)(s+1)}{s(s+0.8)(s-1)}$ , takes the shape as shown in the figure.

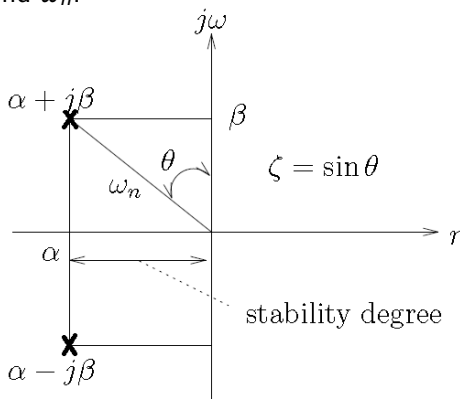


# Outline

- 1 Concept of Root Locus
- 2 How To Sketch the Root Locus
  - General Approach
  - Rules of thumb for sketching the root locus
- 3 Design criteria
- 4 Root Locus and MATLAB
  - Root Locus
  - SISOTOOL

## Second order system

Recall from chapter 5 that the dynamic behavior of the second-order system can be described in terms of only two parameters  $\zeta$  and  $\omega_n$ .



## Root locus method: design criteria

If you are able to express your design criteria in terms of the poles ( $\alpha + j\beta$ , with  $\alpha < 0$  and  $\beta > 0$ ), you can find out if there are any  $K$ -values appropriate and, more importantly, which  $K$ -values.

### Criteria

- The damping ratio:  $\zeta = \frac{\beta}{|\alpha + j\beta|}$  ( $0 \leq \zeta \leq 1$ );
- The natural frequency:  $\omega_n = -\frac{\alpha}{\zeta} = \frac{\beta}{\sqrt{1-\zeta^2}}$ ;
- The rise time:  $t_r \cong \frac{1.8}{\omega_n}$ ;
- The settling time:  $t_s = \frac{4.6}{\zeta\omega_n}$ ;
- The peak time:  $t_p = \frac{\pi}{\omega_d}$ , with  $\omega_d = \omega_n\sqrt{1-\zeta^2}$ ;
- The overshoot:  $M_p = e^{-\pi\frac{\zeta}{\sqrt{1-\zeta^2}}}$ .

## Example design criteria

### Example

We will now apply a design criteria on a system from a previous example which we discussed in the first subsection:

$$G(s) = \frac{1}{s(s+1)}. \quad (10)$$

We can now compute  $K$  at the point where the locus crosses  $\zeta = 0.5$ : we know that if  $\zeta = 0.5$ , then  $\theta = 30^\circ$  and the magnitude of the imaginary part of the root is  $\sqrt{3}$  times the magnitude of the real part. The magnitude of the real part is  $\frac{1}{2}$  and thus, we have:

$$\frac{\sqrt{4K-1}}{2} = \frac{\sqrt{3}}{2} \quad (11)$$

and therefore,  $K = 1$ .

## Root locus method: design criteria

The possible design criteria for second order systems are very useful since they represent physically important measures in terms of the poles.

### Second order approximation

For most of the time only one or two poles dominate the behavior of the system. These are called the dominant poles. Consequently many systems behave more or less as if they are of second order.

- A single pole at position  $-a$  results in a  $e^{-at}$  term in the output. So after some time, the poles with the largest real part will dominate the behavior;
- A single pole at position  $me^{j\omega}$  results in a  $m^k e^{j\omega k}$  term in the output. So after some time the poles with the largest modulus will dominate.

# Outline

- 1 Concept of Root Locus
- 2 How To Sketch the Root Locus
  - General Approach
  - Rules of thumb for sketching the root locus
- 3 Design criteria
- 4 Root Locus and MATLAB
  - Root Locus
  - SISOTOOL

# Outline

- 1 Concept of Root Locus
- 2 How To Sketch the Root Locus
  - General Approach
  - Rules of thumb for sketching the root locus
- 3 Design criteria
- 4 Root Locus and MATLAB
  - Root Locus
  - SISOTOOL



## MATLAB Commands

MATLAB provides a function to plot the root locus and the zero-pole map of a system:

- root locus: **"rlocus(sys)"**
- zero pole: **"pzmap(sys)"**

Most of the time you want to use the plot of the root locus to decide which value of  $K$  you need to satisfy the requirements. To visualize these requirement, you can use the command **"sgrid(requirement1, requirement2,...)"**. This will result in a grid that envisions the requirements.

You can also just include a grid (**"grid"**) in your root locus plot which can help you locate the poles.

## MATLAB Commands

You can choose the desired poles on the locus in MATLAB by using the "**rlocfind(sys)**"-command. This will allow you to select a point on your root locus plot. MATLAB will tell you the precise point you selected, will tell you what the gain is at that point and will tell you the poles of the system with that gain.

MATLAB can also calculate the closed-loop transfer function. You can do this by using the "**feedback(K\*sys,1)**" command. You only need to specify the value of  $K$  if you haven't yet used the "**rlocfind**" command.

## Example

We will now design a controller using MATLAB, the system has the following transfer function:  $H(s) = \frac{s+7}{s(s+5)(s+15)(s+20)}$  and must meet the following design criteria:

- overshoot must be less than 5%;
- rise time = 1s

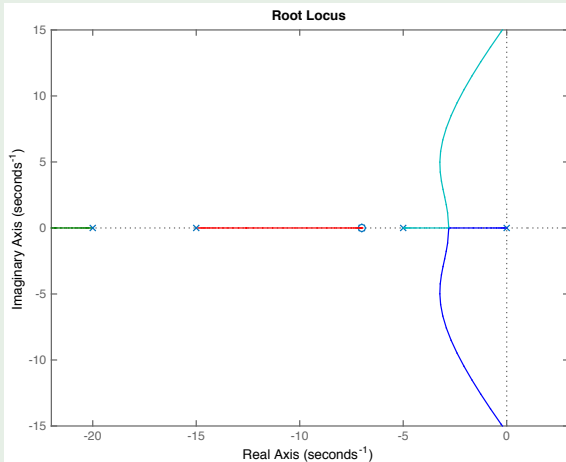
MATLAB commands:

`s = tf('s')`

`H = (s+7)/(s*(s+5)*(s+15)*(s+20))` we define the system H by its transfer function

`rlocus(H)` we command MATLAB to draw the root locus  
`axis([-22 3 15 15])` we adjust the axis

Which results in the following plot:



Next we translate the design criteria into requirements for  $\zeta$  and  $\omega_n$ :

- $\zeta = 0.7$ ;
- $\omega_n = 1.8$ .

We can visualize these requirement using the **"sgrid(requirement1, requirement2,...)"**-command.

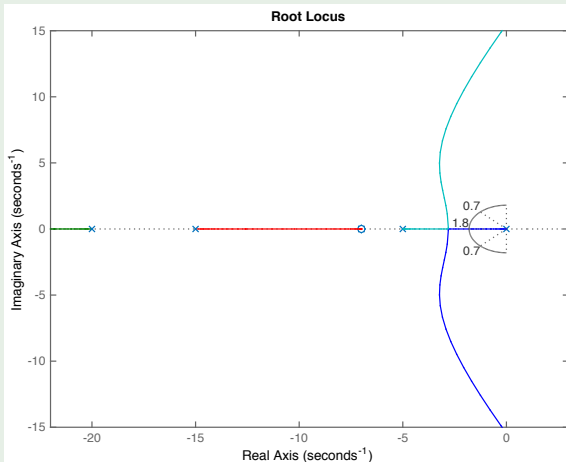
MATLAB commands:

Zeta = 0.7

Wn = 1.8

sgrid(Zeta, Wn)

Which results in the following plot:



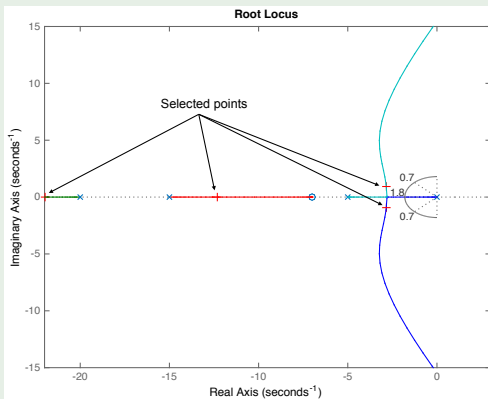
Going back to our problem, to make the overshoot less than 5%, the poles have to be in between the two white dotted lines, and to make the rise time shorter than 1 second, the poles have to be outside of the white dotted semicircle.

From the plot above we see that there is part of the root locus inside the desired region. So in this case, we need only a proportional controller to move the poles to the desired region. You can select a point on the root locus plot that meets the requirement by using the "**rlocfind(sys)**"-command.

MATLAB command:

$[k, poles] = rlocfind(H)$

In the next figure, the  $+$ -sings indicate the selected points. MATLAB will return the gain and the poles of the system you have created by selecting these points.





# Outline

- 1 Concept of Root Locus
- 2 How To Sketch the Root Locus
  - General Approach
  - Rules of thumb for sketching the root locus
- 3 Design criteria
- 4 Root Locus and MATLAB
  - Root Locus
  - SISOTOOL

## MATLAB Commands

Another way to visualize the root locus, is by using the interactive MATLAB GUI called sisotool. The SISO design tool is an interactive graphical user interface that facilitates the design of compensator for single-input, single-output (SISO) feedback loops.

First you need to define your system in MATLAB and then you use the sisotool function on your system: **"sisotool(sys)"**. An interactive user face will pop up in which you can select you plant, the plots you would like to see and many other options.

## MATLAB Commands

### Example

We can also design the same controller as done above but instead of using separate commands, we will now make use of the sisotool.

Same transfer function:  $H(s) = \frac{s+7}{s(s+5)(s+15)(s+20)}$

and the same design criteria:

- overshoot must be less than 5%;
- rise time = 1s

MATLAB commands: `s = tf('s')`

`plant = (s+7)/(s*(s+5)*(s+15)*(s+20))` we define the system H by its transfer function

`sisotool(plant)` we summon the sisotool, an interactive user face will pop up

## MATLAB Commands

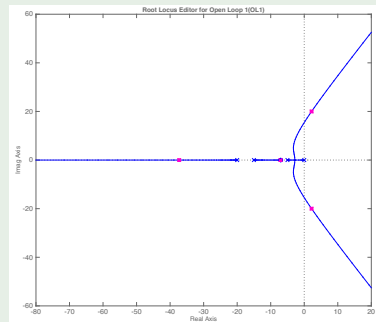
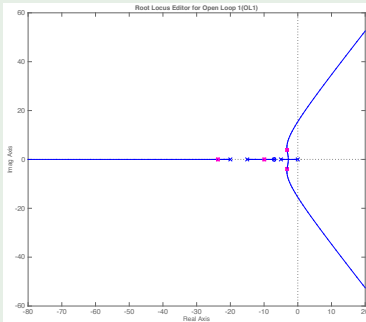
In the **Control and Estimation Tools Manager** you can select the tab labeled **Graphical Tuning**. In this tab:

- turn **Plot 2** off;
- make sure **Plot 1** is the root locus.

Now you can click the button labeled **Show Design Plot** and then a tunable plot will appear. In this root locus plot, you can drag the closed-loop poles along the locus to adjust the loop gain.

# MATLAB Commands

The following figures are an example of the result you get when you move the closed-loop poles.



## MATLAB Commands

We can take the design criteria into account directly in the plot. This is done by right-clicking and selecting **Design Requirements, New**. Now you can specify the different design criteria: settling time, percent overshoot, damping ratio, natural frequency, region constant.

In our example we have 2 design criteria which we have already translated into requirements for  $\zeta$  and  $\omega_n$ :

- $\zeta = 0.7$ ;
- $\omega_n = 1.8$ .

On the plot, any area which is still white, is an acceptable region for the poles.

## MATLAB Commands

If we complete the specification of the requirements and adjust the axis for a better view, we obtain the following figure:

