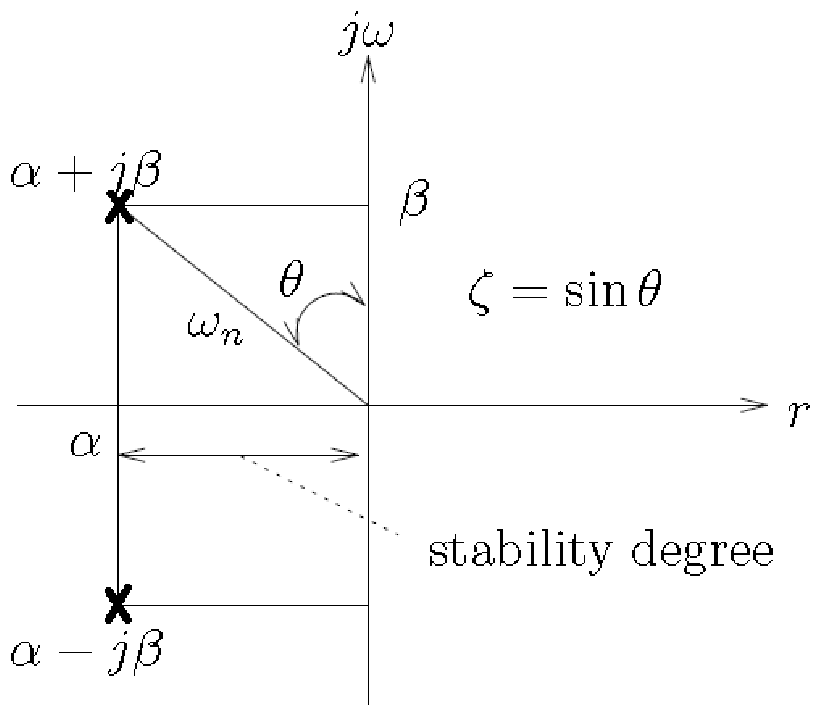
Outline

1. Concept of Root Locus
2. How To Sketch the Root Locus
    - General Approach
    - Summary of the rules 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 ζ and ω_n, where $\alpha = -\zeta\omega_n$ and $\beta = \omega_n\sqrt{1-\zeta^2}$.



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 some appropriate K-values 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: $M_p = e^{-\pi\frac{\zeta}{\sqrt{1-\zeta^2}}}$.

Root locus method: design criteria

Example

Let's go back to the problem of the DC motor discussed in the first subsection, whose transfer function is:

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

For this particular case, we are interested in finding a gain K for which the poles of the closed-loop system has a damping ratio equal to 0.5.

We want to design a proportional controller K that leads the closed-loop systems to have damping-ratio $= 0.5$.

Root locus method: design criteria

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. Previously, we already found the following equation and its roots:

$$s^2 + s + K = 0$$
$$r_1, r_2 = -\tfrac{1}{2} \pm \tfrac{\sqrt{1-4K}}{2}$$

The magnitude of the real part is $\frac{1}{2}$ and thus, we have:

$$\frac{\sqrt{4K-1}}{2} = \frac{\sqrt{3}}{2} \tag{11}$$

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
   - Summary of the rules 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
   - Summary of the rules 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 have not 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:

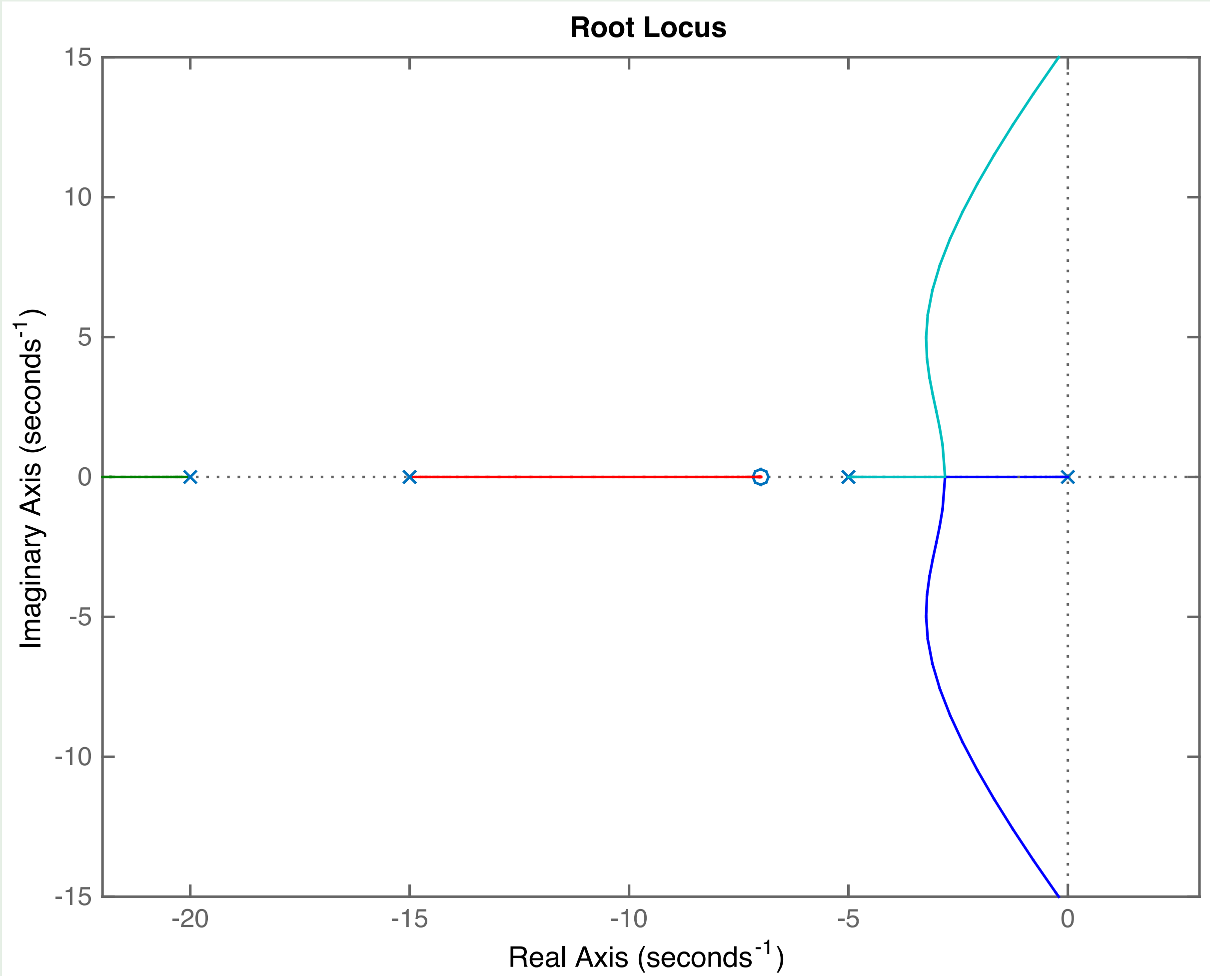
s = tf ('s')

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:



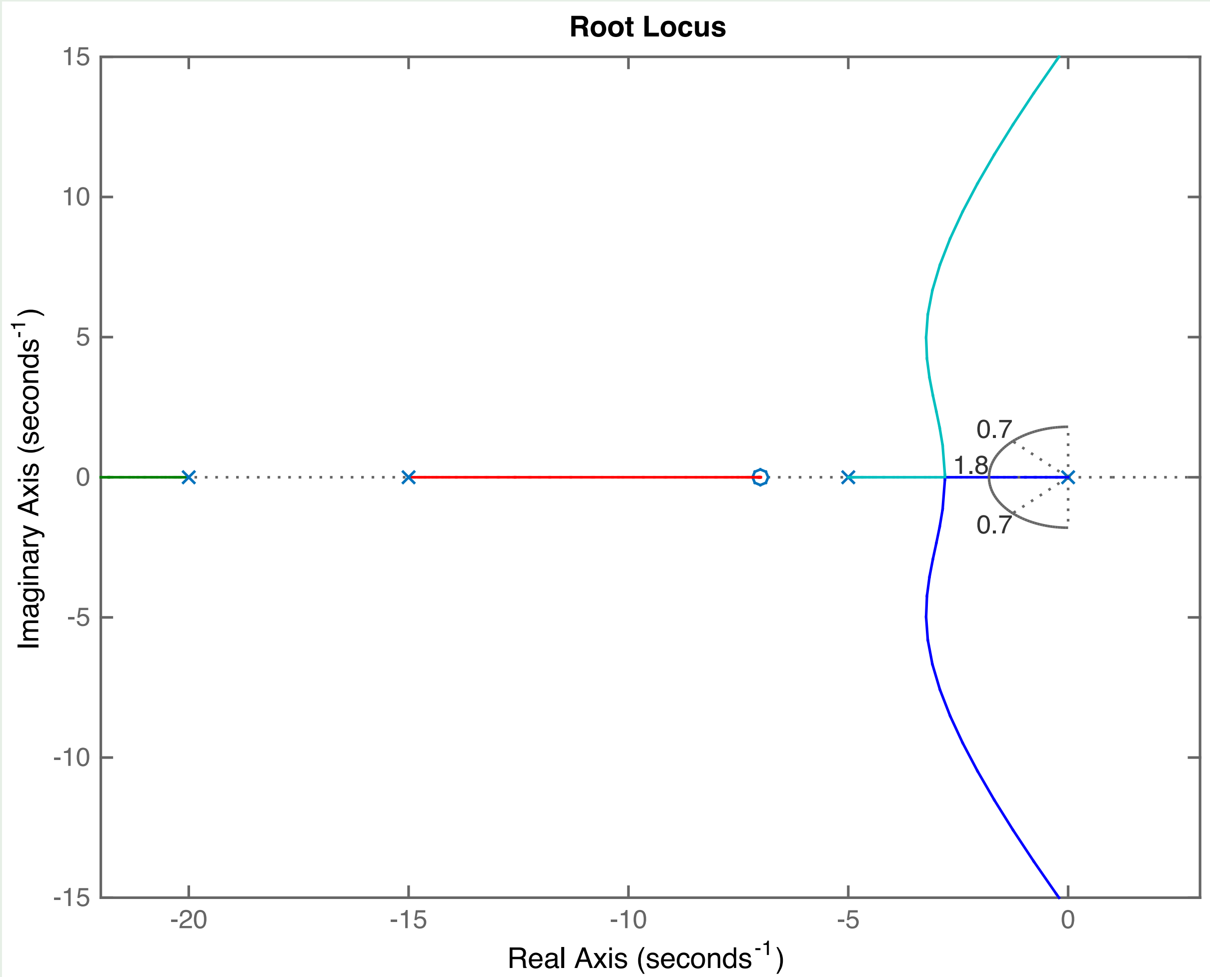
Next we translate the design criteria into requirements for ζ and ω_n:

- $M_p = e^{-\pi \frac{\zeta}{\sqrt{1-\zeta^2}}} < 0.05 \rightarrow \zeta \leq 0.7$;
- $t_r \cong \frac{1.8}{\omega_n} \leq 1s \rightarrow \omega_n \geq 1.8.$

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:

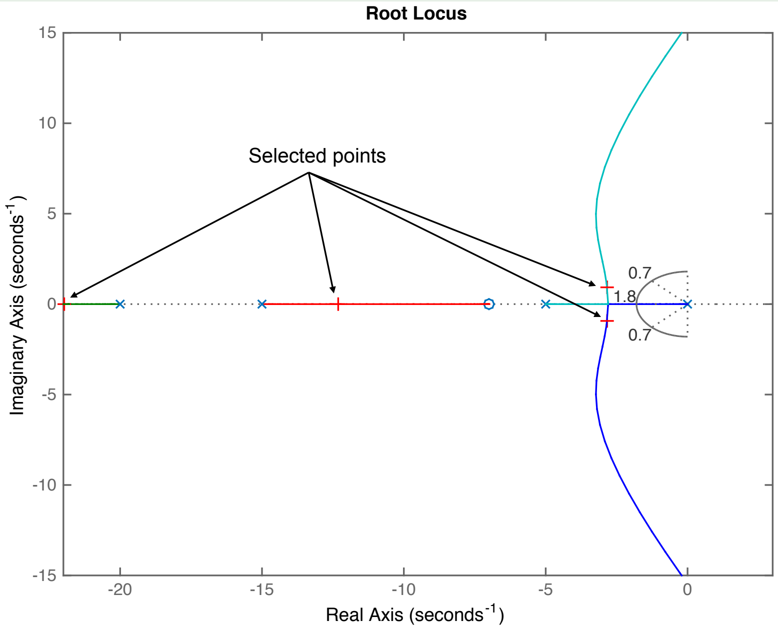


Going back to our problem, to make the overshoot less than 5%, the poles have to be in between the two black dotted lines, and to make the rise time shorter than 1 second, the poles have to be outside of the black semicircle.

From the previous plot 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.



MATLAB will return the numerical result for the point you selected. E.g.:

- Selected point: $-12.4325 + 0.0000j$;
- Gain: $K = 330.4944$;
- Poles: $-21.9209 + 0.0000j$, $-12.4325 + 0.0000j$, $-2.8233 + 0.7196j$, $-2.8233 - 0.7196j$.

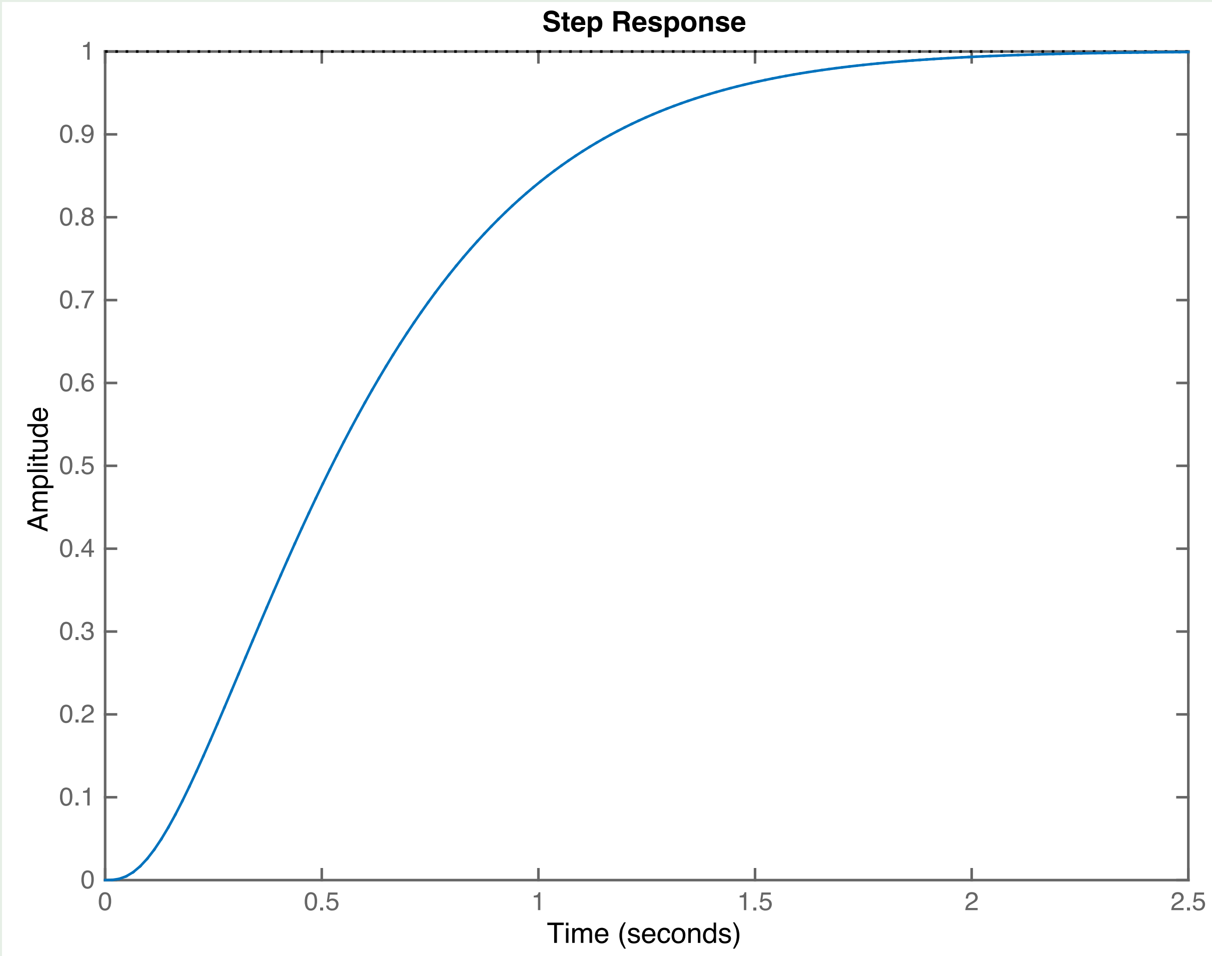
To corroborate that the design specifications have been met, you should plot the step response of the closed-loop system you designed. You can do that with MATLAB by using the command **step(sys)**.

First we need to specify the closed-loop system in MATLAB. Based on the previous results, we choose a gain of $K = 350$, next we use the **feedback**-command.

MATLAB commands:

```
K = 350
Hcl = feedback(K*H,1)
```

Finally we plot the step-response:



Outline

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 execute the sisotool. An interactive user face will pop up

MATLAB Commands

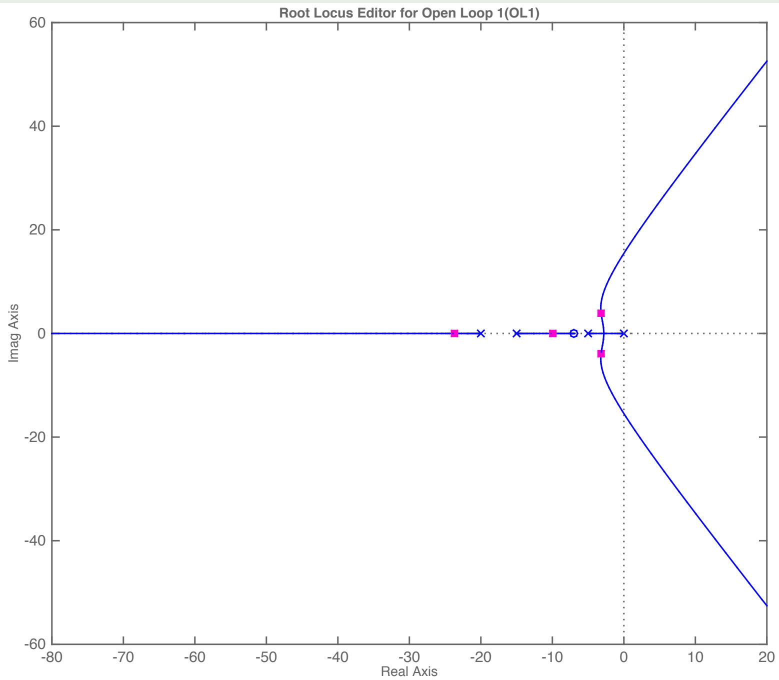
In the **Control and Estimation Tools Manager** you can select the tab labeled **Graphical Tuning**. In this tab:

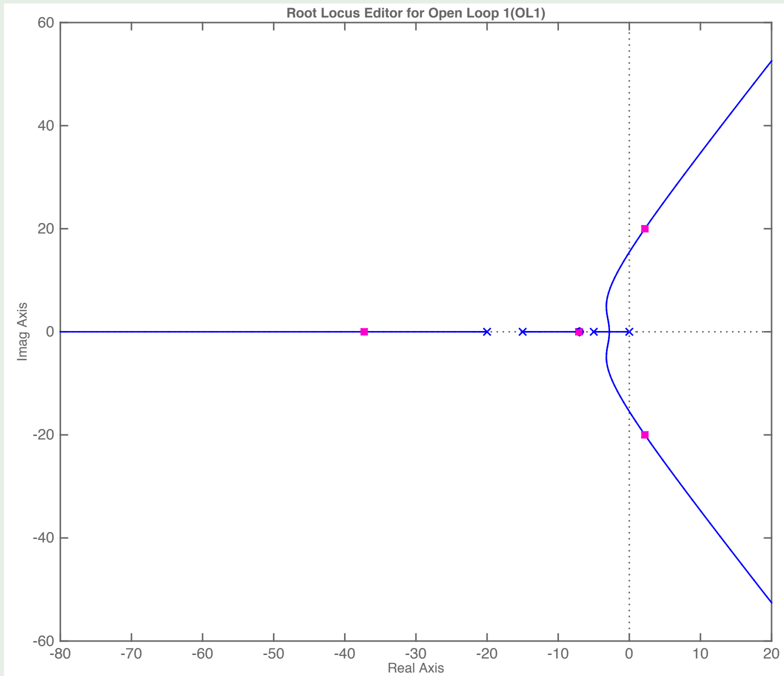
- turn **Plot 2** off;
- make sure **Plot 1** is the root locus.

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 ζ and ω_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: