

# EENG307: Designing Controllers Using Bode Plots, Part 1\*

Lecture 32

Elenya Grant, Kathryn Johnson, and Hisham Sager<sup>†</sup>

Fall 2022

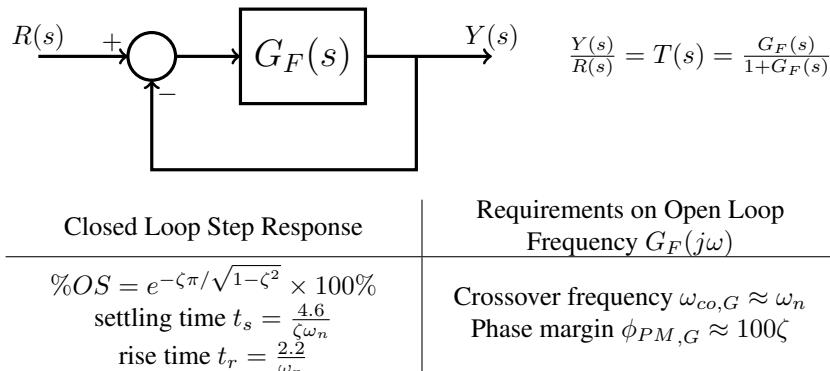
## Contents

<b>1 Frequency Domain Specifications</b>	<b>1</b>
<b>2 Analysis of Proportional Control</b>	<b>3</b>
<b>3 Lecture Highlights</b>	<b>7</b>

## 1 Frequency Domain Specifications

Previously we saw how to translate time domain specifications to frequency domain specifications for a unity gain feedback system

### Transient Response Specifications



We can also add requirements on the open loop frequency response due to closed loop steady state error specifications. Recall our investigations of steady state error due to a reference input

System Type	Steady State Error to a...		Error Constants
	Step Input $r(t) = Au(t)$	Ramp Input $r(t) = Atu(t)$	
0	$e_{ss} = \frac{A}{1+K_s}$	$e_{ss} = \infty$	$K_s = \lim_{s \rightarrow 0} G_F(s) = G_F(0)$
1	$e_{ss} = 0$	$e_{ss} = \frac{A}{K_v}$	$K_v = \lim_{s \rightarrow 0} sG_F(s)$

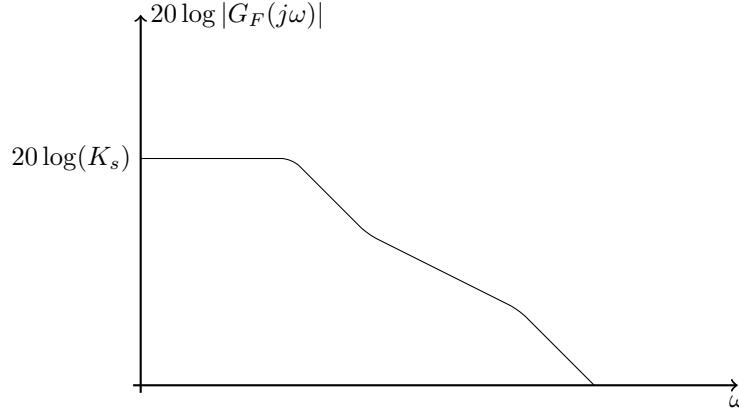
These requirements can be summarized in the plot of the ideal *open loop* frequency response for  $G_F(s)$  for two cases.

\* This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

<sup>†</sup> Developed and edited by Tyrone Vincent and Kathryn Johnson, Colorado School of Mines, with contributions from Salman Mohagheghi, Chris Coulston, Kevin Moore, CSM and Matt Kupiliuk, University of Alaska, Anchorage

### Open Loop Frequency Response Specifications When $K_s = \text{finite}$ and $K_v = 0$

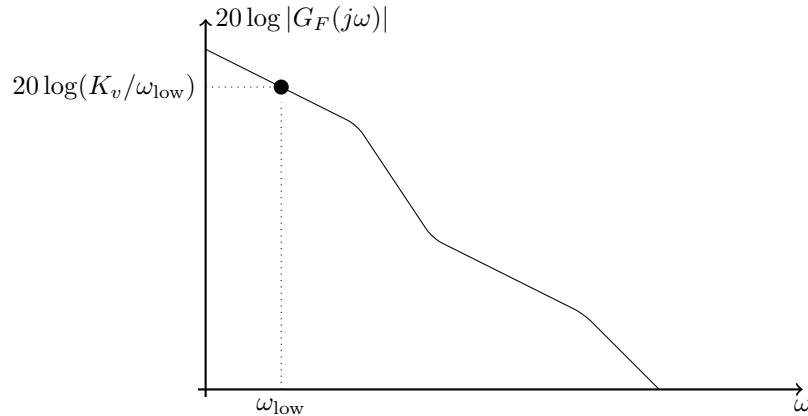
In the first case (Type 0 system), the magnitude Bode plot has a low-frequency slope of 0 dB/dec, as shown in the figure below.



For this Type 0 system,  $K_s = G_F(0)$  is finite, so when the reference is a step the steady state error is  $e_{ss} = \frac{1}{1+K_s}$ . When the reference is a ramp, the steady state error  $e_{ss} = \infty$ .

### Open Loop Frequency Response Specifications When $K_s = \infty$ $K_v = \text{finite}$

In the second case (Type 1) system, the magnitude Bode plot has a low-frequency slope of  $-20\text{dB/dec}$  because of the integrator in  $G_F(s)$  (i.e., the term that makes the system Type 1).



In this Type 1 case,  $K_v = \lim_{s \rightarrow 0} sG_F(0)$  is finite, so when the reference is a step the steady state error is  $e_{ss} = 0$ . When the reference is a ramp, the steady state error is  $e_{ss} = \frac{1}{K_v}$ .

The important features of the open loop frequency response are

- Low frequency gain: either a large gain or a gain that is increasing as  $\omega \rightarrow 0$  (i.e., moving from right to left on the magnitude Bode plot) can decrease the steady-state error
- Crossover at desired bandwidth (with slope of -20dB/dec) to achieve the desired closed-loop natural frequency
- Phase high at crossover for good phase margin to achieve the desired closed-loop phase margin

## 2 Analysis of Proportional Control

One way to design a control system to meet step response specifications is to choose a controller  $C(s)$  so that  $G_F(s) = C(s)G(s)$  has the desired crossover frequency and phase margin. In this section, we try to do this for the case of a proportional control  $C(s) = K$ .

Consider the following system:

$$G(s) = \frac{5}{s(s+1)(s+10)}$$

We would like to design a controller so that the closed-loop, unity feedback system (i.e., our standard configuration) meets these specifications:

- Zero steady state error to a step input
- Rise time of 1.1 s
- $\%OS \leq 16.5\%$ .

The first step is to translate these requirements into the frequency domain. It is usually easiest to start with % overshoot. The required closed loop damping ratio is found via

$$\zeta_T = \frac{-\ln(.165)}{\sqrt{\pi^2 + \ln(.165)^2}} \geq .50$$

The rise-time specification translates to a requirement on  $\omega_n$  via

$$t_r = \frac{2.2}{\omega_n} = 1.1$$

$$\omega_n = \frac{2.2}{1.1} = 2$$

Thus the desired  $\omega_{n,T} = 2$  rad/s.

Finally, the steady state error specification requires a Type 1 loop gain. Since  $G(s)$  already has an integrator, an integrator does not need to be added within the controller, so using proportional control  $C(s) = K$  should be adequate.

These requirements on the closed loop poles are then translated to requirements on the loop gain (see Lecture 31 for derivations):

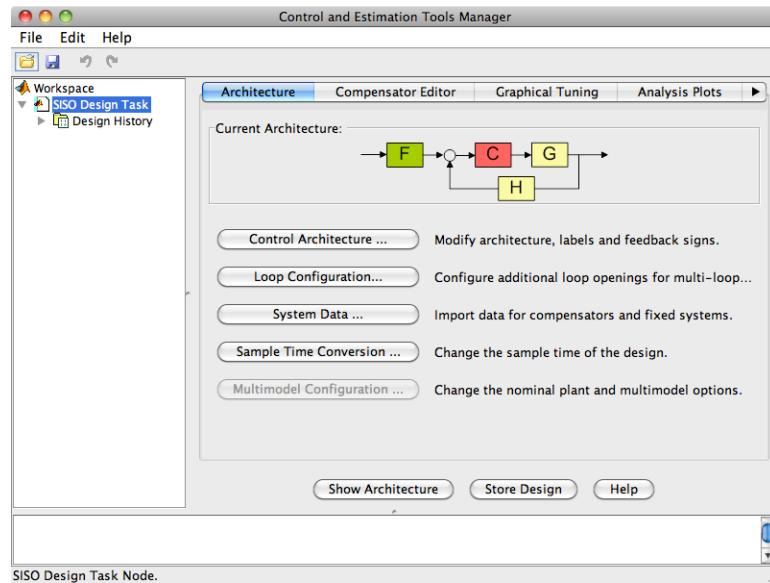
- We can use our approximate relationship  $\phi_{PM,G} = 100 \times \zeta_T$ , so the desired phase margin is  $50^\circ$  or greater.
- We choose  $\omega_{co,G} = \omega_{n,T} = 2$  rad/s

To examine the design, we will use a control design tool in MATLAB. First, let's enter in the system transfer function

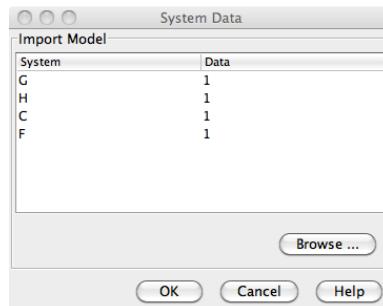
```
>> s=tf([1 0],1)
>> G= 5/(s*(s+1)*(s+10))
```

The design tool is initialized by typing `sisotool` at the command prompt. Depending on which version of Matlab you are using, `sisotool` may look somewhat different (and is now also called Control System Designer), but the big picture concepts are consistent across most versions.

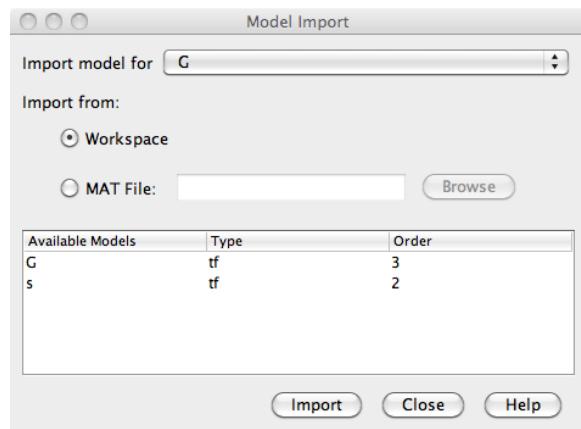
```
>> sisotool
```



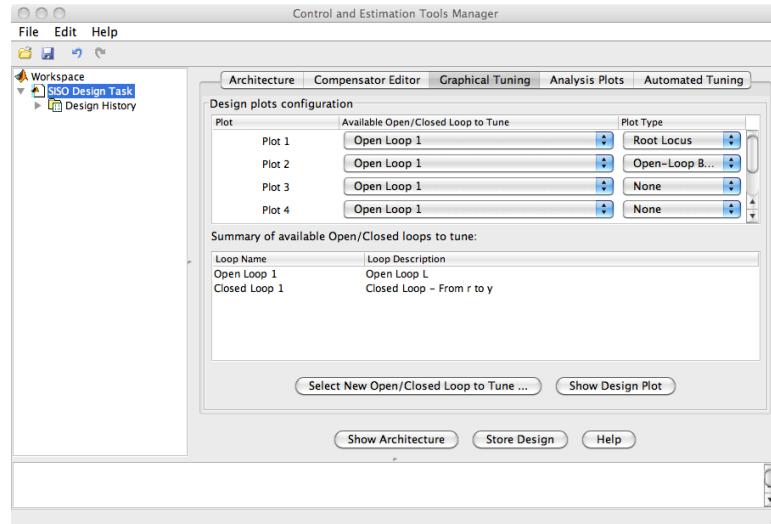
Click on “System Data” to load the system transfer function. It opens up the following window



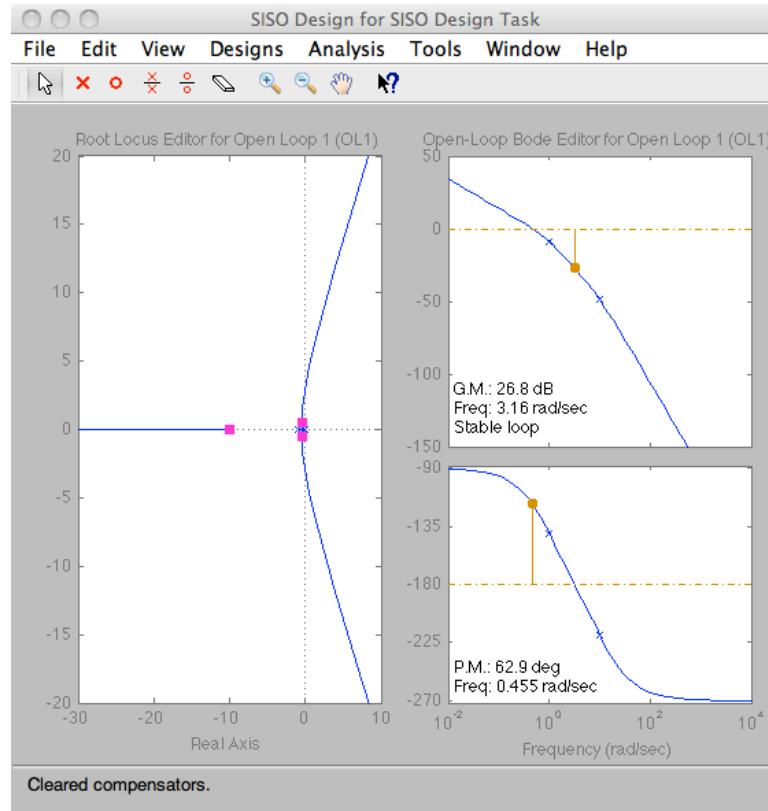
Click on “Browse ...” to open up yet another window



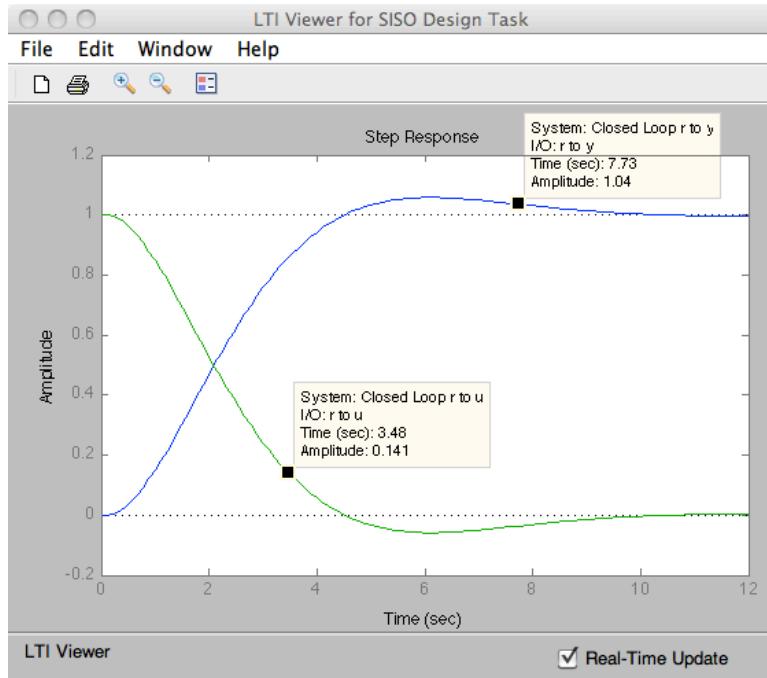
Click on “G” to choose the transfer function you entered at the command line. Click “Import”, and “Close”, and then click on “OK” on the System Data window. In the Control and Estimation Tools Manager window, click on “Graphical Tuning”. In the row “Plot 3”, choose “Plot Type” as None, so that the window looks like the following:



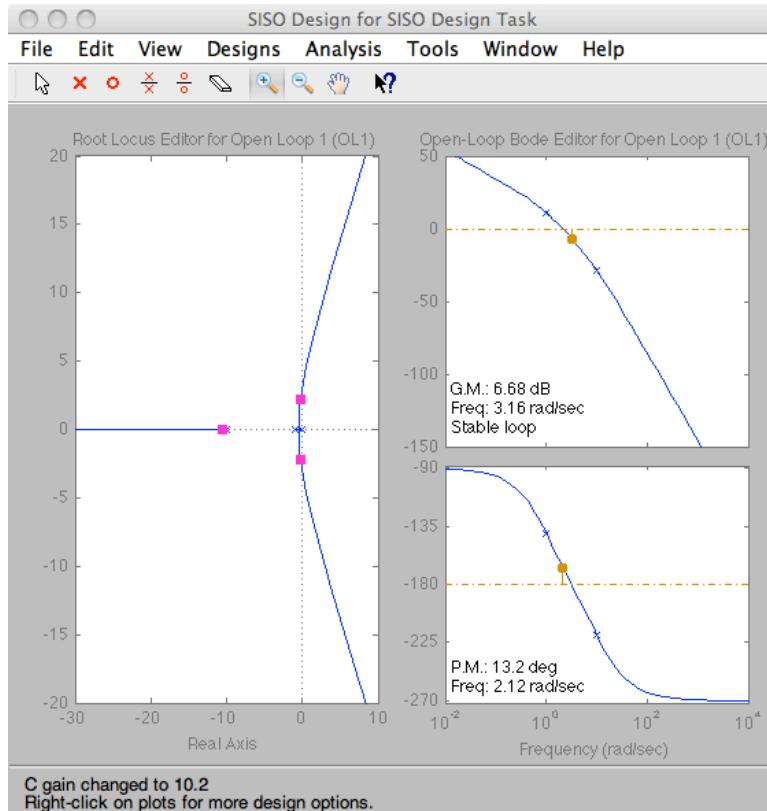
Click on “Show Design Plot”. This opens up the window below, where on the left we see the root locus of  $G(s)$ , and on the right is the frequency response of  $G(s)$ .



In the menu of the SISO Design for SISO Task, click on “Analysis -> Response to Step Command”. This opens the window below. The blue line shows the *closed loop* step response, while the green line shows the actuator command (i.e. the signal  $u$  which is input to  $G(s)$  in the block diagram). This is confirmed by clicking on the plots to see the legend information.

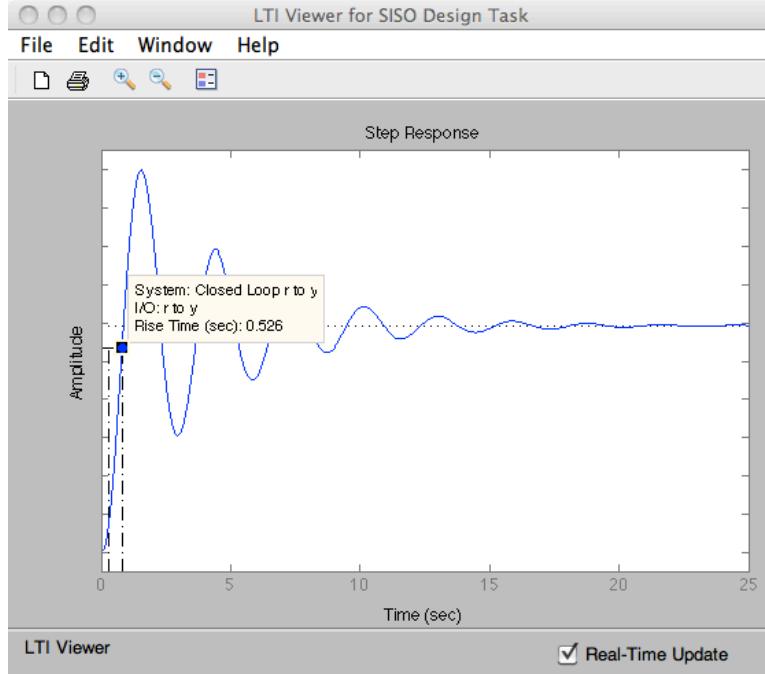


This step response does not meet our design specifications, because the rise time is greater than 0.9 seconds. This is reflected in the Bode plot, as the crossover frequency is 0.455 rad/s (as seen in phase plot, where it also says that the phase margin is 62.9 degrees) instead of 2 rad/s, where we want it. We can increase the crossover frequency by increasing the gain. This can be done in the SISO Design for SISO Design Task window by placing the cursor over the magnitude Bode plot until a hand appears. At this point, a left click will allow you to grab the magnitude Bode plot and shift it up and down. Shift it up until the crossover frequency is 2 rad/s, as shown in the following figure



Notice that the crossover frequency is labeled in the *phase Bode plot* (in this image, the text Freq: 2.12 rad/sec) because it's the crossover frequency that is associated with the phase margin  $\phi_{PM}$ .

With this adjustment in the crossover frequency, we can check on the closed loop step response. Right click on the LTI Viewer for SISO Design Task window, select “Systems”, and de-select the “Closed loop r to u response”. Then right click again, select “Characteristics -> Rise Time”. Click on the blue dot that indicates the rise time to get the plot below



This meets our rise time spec, but now we have a large overshoot! The crossover frequency is 2 rad/second, but since the phase at this frequency is close to  $-180^\circ$ , the phase margin is very low, and the step response is accordingly very oscillatory. This problem can be addressed by using additional terms (poles and/or zeros) in  $C(s)$  to alter the crossover frequency without reducing the phase margin, which will be shown in Lecture 34.

### 3 Lecture Highlights

The primary takeaways from this article include

1. This article gives an example of designing a proportional controller using frequency response techniques and Matlab’s *sisotool*.
2. Please see the Lecture 31 for the derivations of the frequency response control techniques used.