Cascade Compensation Design with Complex Values

*Jacob A. Wilroy1*

Department of Aerospace Engineering and Mechanics

The University of Alabama, Tuscaloosa, AL 35487

# Nomenclature

*a* = constant

*e* = position error

*G(s)* = plant transfer function

*H(s)* = cascade compensator transfer function

= closed loop transfer function

= pole of transfer function

= gain

= position error constant

*r* = reference input

*s* = transfer function variable

*u* = plant input

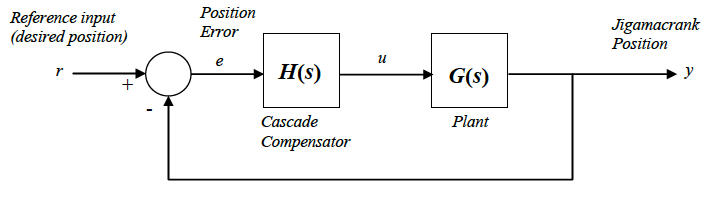
*y* = plant output

= zero of transfer function

**I. Introduction**

For this assignment, two unique poles were given to each student for use in a predetermined plant equation, which is given by Eqn. 1. Using the unitary feedback loop shown by Fig. 1, and the plant equation, the student needed to design cascade compensation that would change the response of the function to meet given criteria when given a step input. Depending on the poles assigned, the plant function could be unstable, have large, undamped oscillations, or have other characteristics. These characteristics needed to be determined so that compensation could be implemented to achieve the requested output.

|  |  |
| --- | --- |
|  | (1) |

**Figure 1. Block diagram of closed loop system**

In order to design compensation for the system, design criteria are needed. When given a reference input *r*, the output *y* needed to settle to within 10% of the desired value in 1 second, and to within 1% in 5 seconds. The output also needed to have a rise time of less than 0.5 seconds and avoid excessive oscillation.

**II. Compensator Design Process**

For the plant function, the two poles given were

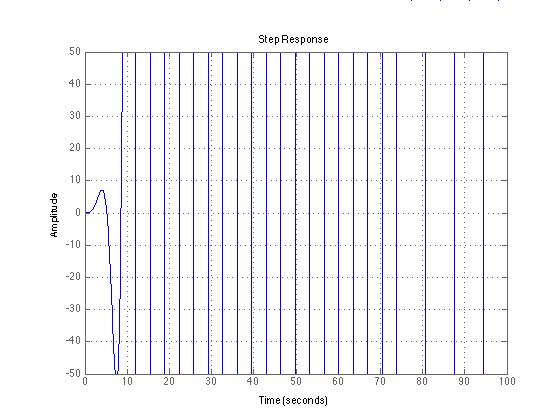
|  |  |
| --- | --- |
|  | (2) |

When these two poles were inserted into the given plant function and then into a unitary feedback loop, the closed loop transfer function became

|  |  |
| --- | --- |
|  | (3) |

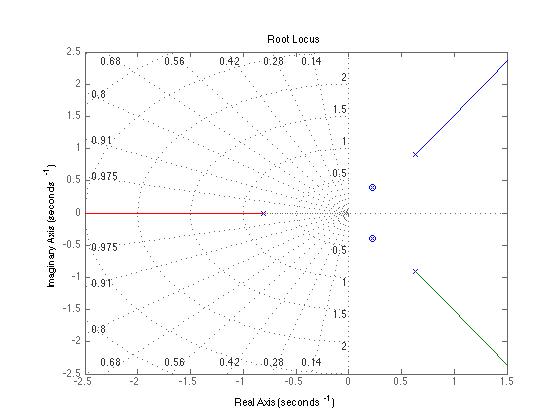
where *H* is equal to one because a cascade compensator has not yet been designed.

The response of the step input is presented by Fig. 2.



**Figure 2.** **Step input response reveals the unstable characteristics of the plant function within the closed loop transfer function.**

As can be seen, the output does not meet the design criteria and the function is very unstable. This is where the compensation design process came in. Examining the given poles for the plant function, one notices that they are a complex conjugate while also values that exist on the right half plane (RHP). Existence on this side of the plane leads to an unstable function, which yields the above step response. When examining the root locus plot given by Fig. 3, one notices that the root loci of the two poles on the RHP go to positive infinity. This leads to the instability in Fig. 2 and needs to be corrected so that the design criteria can be met. This can be done by using compensation to attract the poles from the RHP over to the left half plane (LHP).



**Figure 3. Root locus plot of the closed loop transfer function without compensation**

Since zeros attract poles, two zeros could simply be placed on the real axis to attract the root loci of the poles. This, however, violates causality yielding a transfer function whose highest polynomial is in the numerator rather than the denominator. For this reason, either a lead or lag compensator (or a combination of those) needs to be used to achieve the response required to meet the design criteria. Using Matlab’s “sisotool” function, one can graphically places poles and zeroes on a root locus plot and modify the compensator values to achieve the desired output. Since there is a complex pair of poles on the RHP, a complex pair of zeros on the LHP is needed to attract the root loci as the gain is increased, but this cannot be done without adding a complex pair of poles as well. To keep the new poles from attracting the newly placed zeros, the poles need to be placed at a distance far enough from the imaginary axis to limit their effect on the zeros. Upon doing the calculation to determine the departure angle from the poles trying to be attracted from the RHP, one will notice that zeros placed at a location of approximately -100 Hz on the real axis do not have much of an effect on the departure angle. Keeping in mind that the goal is to place zeros close to the imaginary axis to attract the unstable poles, and the fact that Matlab’s “sisotool” is a very useful piece of software, one can place poles and zeros on the root locus plot of the “sisotool” and alter the values to attract the root loci. This has to be done very carefully as the transfer functions are very sensitive (small changes in pole/zero location cause dramatic changes of root loci). After much experimentation, the following transfer function was created for a compensator

|  |  |
| --- | --- |
|  | (4) |

with a gain

|  |  |
| --- | --- |
|  | (5) |

where the zeros of the compensator are

|  |  |
| --- | --- |
|  | (6) |

and the poles are

|  |  |
| --- | --- |
|  | (7) |

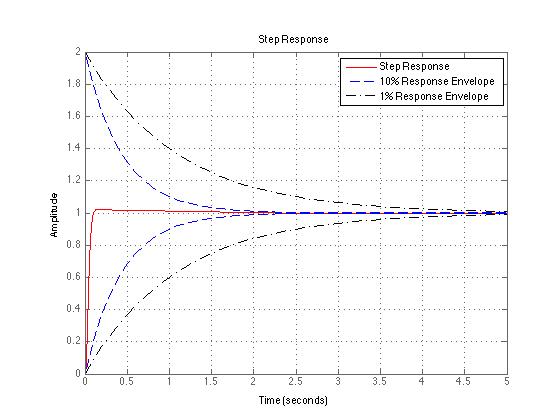
Since the poles are larger than the zeros, this is called a lead filter.

**III. Results**

After designing lead compensation including gain and implementing it into the closed loop transfer function, a very long transfer function is formed. This function is included in appendix A, but its results are displayed here. In order to determine whether the response curve fits the criteria, envelopes needed to be developed. Since the desired final position is one, the following equation can be used with the given criteria,

|  |  |
| --- | --- |
|  | (8) |

By solving for *a* in the above equation using the criteria, one gets the envelopes that have been implemented in the Figs. 4 and 5. Below is the step response plot of the closed loop transfer function including compensation and gain. The response clearly has a rise time of less than 0.5 seconds, and has settled to within 10% of the desired value in 1 second.

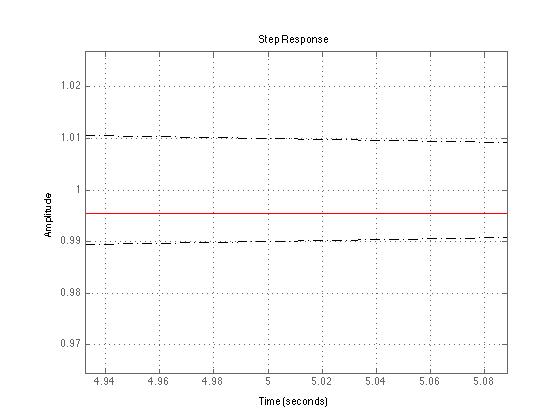


**Figure 4. Step response plot of the closed loop transfer function including lead compensation and gain. The plot also includes the design criteria envelopes.**

Figure 5 shows the same plot as Fig. 4, but focuses on the response near 5 seconds. The response has settled to within 1% of the desired value in less than 5 seconds, so it meets the design criterion. However, the response seems to have settled on a value slightly smaller than 1. This is due to the fact that the type of transfer function that is created from the control loop is type zero. The amount of error of this response, since it is a step input, is determined by the step error constant. For type zero systems, the steady state error is given by

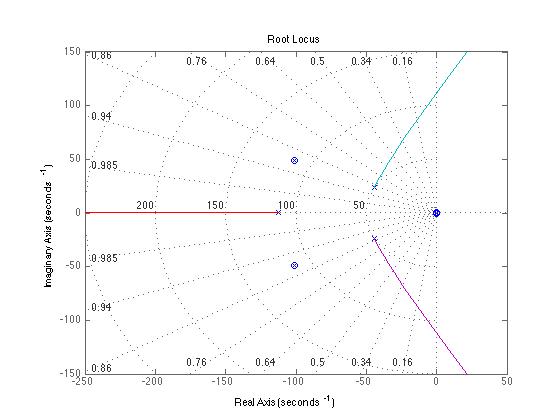
|  |  |
| --- | --- |
|  | (9) |

where is the position error constant. Because this is a type zero system, the position error is nonzero and finite. The error could be decreased by changing the type of system, but this is not necessary since the design criteria have been met.

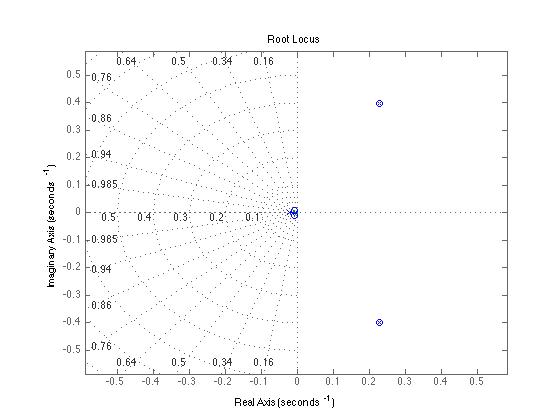


**Figure 5. Step response plot of the closed loop transfer function focusing on five seconds. The plot also includes the five seconds design criterion envelope.**

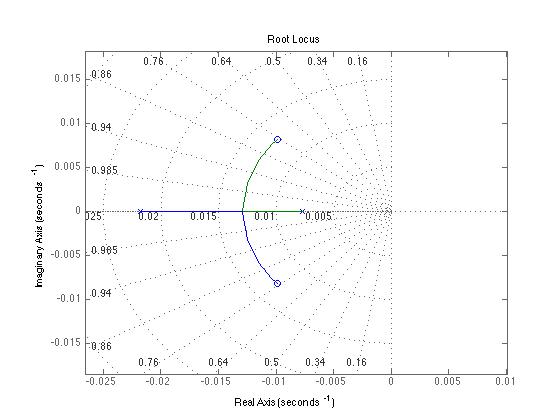
Below is a root locus plot of the closed loop transfer function. At first glance one notices that there is a complex pair of poles located in the LHP whose root loci travel into the RHP towards infinity, but this does not create an unstable function. The poles in question were those located in the RHP. The cascade compensation has now placed zeros directly on top of the poles creating a stable function.



**Figure 6. Root locus plot of the closed loop transfer function including cascade compensation and gain.**



**Figure 7. Same plot as above, but focusing more on the origin.**



**Figure 8. Root locus plot of the closed loop transfer function again, but with an even greater focus on the activity near the origin.**

**Appendix**

Matlab code written to plot root locus and step function:

clear; clc;

s = tf('s');

pone = 0.2273+0.3982i; ptwo = 0.2273-0.3982i;

G = 1/(s\*(s-pone)\*(s-ptwo))

K=0.0037746

Hlead = ((1+1.2e2\*s+(78\*s)^2)/(1+0.016\*s+(0.0089\*s)^2));

H = Hlead\*K

CLTF = (G\*H)/(1+G\*H)

figure(1)

rlocus(CLTF)

grid on

figure(2)

grid on

step(CLTF,'r')

axis([0 5 0 2])

hold on; grid on

t = 0:0.1:10;

y1 = 1+exp(-2.303\*t); y2 = 1-exp(-2.303\*t); y3 = 1+exp(-0.9210\*t); y4 = 1-exp(-0.9210\*t);

plot(t,y1,'--b')

plot(t,y3,'-.k')

plot(t,y2,'--b')

plot(t,y4,'-.k')

legend('Step Response','10% Response Envelope','1% Response Envelope')

Output printed to the screen when code is run:

Warning: The numerator or denominator of this transfer function has complex-valued coefficients.

> In warning at 26

In tf.tf>tf.tf at 348

In C:\Program Files\MATLAB\R2010b\toolbox\shared\controllib\engine\+ltipack\matchType.p>matchType at 106

In InputOutputModel.plus at 57

In InputOutputModel.InputOutputModel>InputOutputModel.minus at 324

In AEM468Assignment2 at 4

Warning: The numerator or denominator of this transfer function has complex-valued coefficients.

> In warning at 26

In tf.tf>tf.tf at 348

In C:\Program Files\MATLAB\R2010b\toolbox\shared\controllib\engine\+ltipack\matchType.p>matchType at 106

In InputOutputModel.plus at 57

In InputOutputModel.InputOutputModel>InputOutputModel.minus at 324

In AEM468Assignment2 at 4

Transfer function:

1

---------------------------

s^3 - 0.4546 s^2 + 0.2102 s

K =

0.0038

Transfer function:

22.96 s^2 + 0.453 s + 0.003775

------------------------------

7.921e-005 s^2 + 0.016 s + 1

Transfer function:

0.001819 s^7 + 0.3666 s^6 + 22.81 s^5 - 9.913 s^4 + 4.627 s^3 + 0.09352 s^2

+ 0.0007935 s

----------------------------------------------------------------------------

6.274e-009 s^10 + 2.529e-006 s^9 + 0.0004121 s^8 + 0.03344 s^7 + 1.338 s^6

+ 21.92 s^5 - 9.292 s^4 + 4.437 s^3 + 0.1377 s^2 + 0.0007935 s