

Design of Closed-Loop Control of the Jigamacrank of a Widget

Amber Deja

The University of Alabama Undergraduate Aerospace Engineering, Tuscaloosa, AL, 35401

A company, Fly by Night Industries, Inc., is working toward introducing a new line of widgets. This widget is operated by selecting the position of the widget's jigamacrank. This command will then cause the actuator to move the jigamacrank to the selected position. For this assignment, a jigamacrank must be designed. The position of the jigamacrank must have a rise time of less than 0.5 seconds, avoid excessive oscillation, settle to within 10% of the desired position in 1 second, and settle to within 1% of the desired position within 5 seconds. The compensation design process and results are documented below.

Nomenclature

a = zero of cascade compensator transfer function

b = pole of cascade compensator transfer function

$CLTF$ = closed loop transfer function

$G(s)$ = plant transfer function

$H(s)$ = cascade compensator transfer function

K = gain of closed loop transfer function

p_1 = pole of plant transfer function

p_2 = second pole of plant transfer function

s = transfer function variable

t = time

I. Introduction

The transfer function between the input to widget's jigamacrank position selector and the actual position of the jigamacrank is shown below in Eq. (1).

$$G(s) = \frac{1}{(s)(s + p_1)(s + p_2)} \quad (1)$$

Figure 1 displays the plant embedded in a unitary feedback system. The part of the feedback system that is designed in this assignment is $H(s)$.

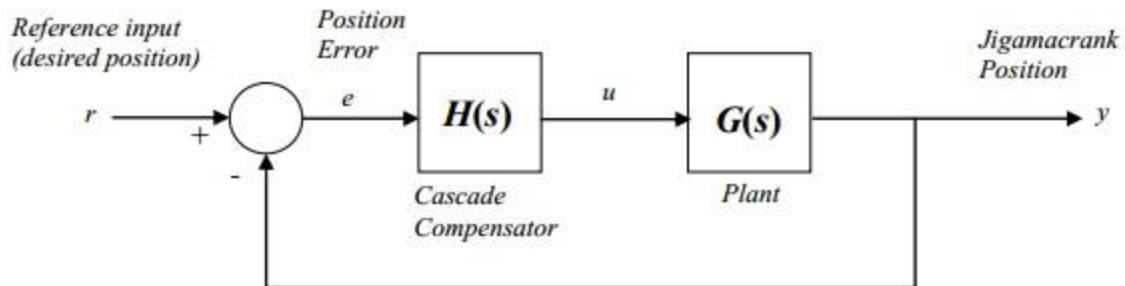


Fig. 1 Block Diagram of Closed-Loop System

The poles (p_1 and p_2) of the plant transfer function, $G(s)$, are given as $0.1228 \pm 0.3350i$ making the plant transfer function for this specific widget that of Eq. (2).

$$G(s) = \frac{1}{s * (s - (0.1228 + 0.3350j)) * (s - (0.1228 - 0.3350j))} \quad (2)$$

This is a complex, unstable pole pair. They are unstable because the poles lie in the right half of the s-plane, as shown by the root locus diagram of the closed loop transfer function without compensation ($H(s) = 1$) in Fig. 2. The closed loop transfer function is shown in Eq. (3).

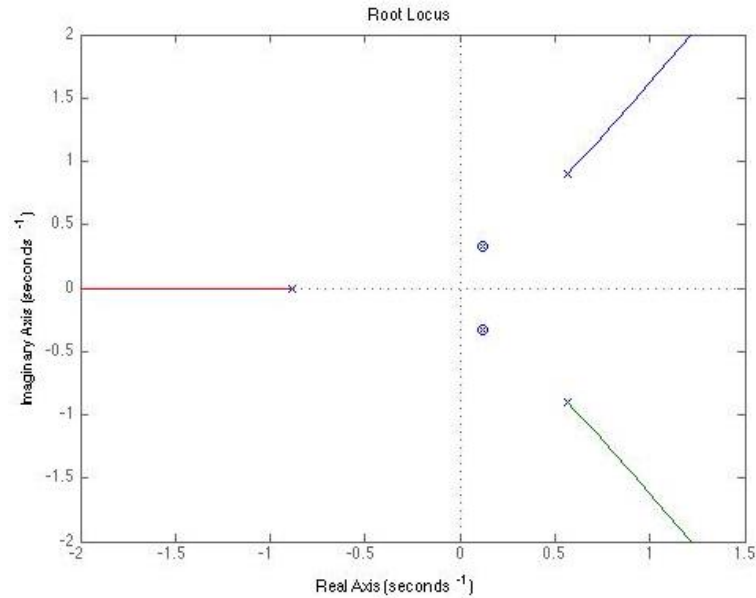


Fig. 2 Root Locus Diagram of Closed Loop Transfer Function without Compensation

$$CLTF = \frac{G(s) * H(s)}{1 + G(s) * H(s)} \quad (3)$$

The step function for the closed loop transfer function without compensation is shown in Fig. 3. As predicted, the step response is unstable due to the poles lying in the right half of the s-plane. The amplitude does not settle and none of the design constraints are met.

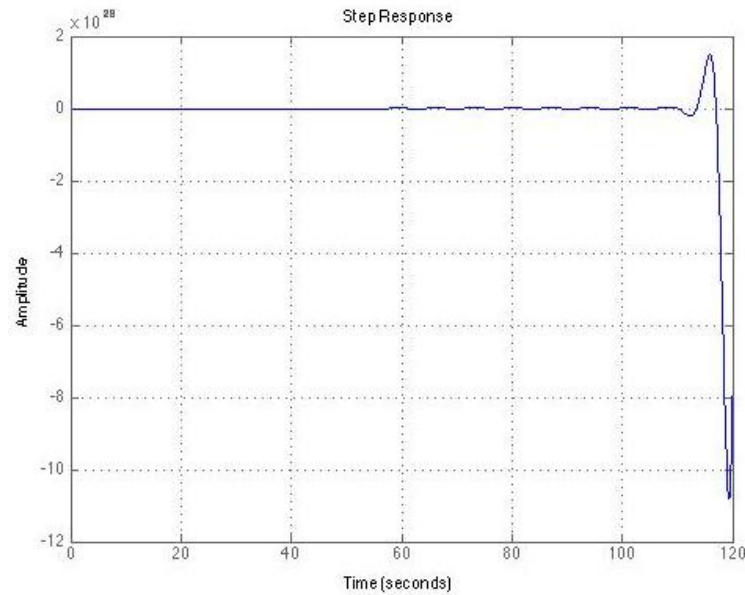


Fig. 3 Step Response of Closed Loop Transfer Function without Compensation

In order to stabilize the system and meet the design constraints, compensation must be added to the system.

II. Compensation Design Process

The following is a description of the sequential steps it took to stabilize the system and meet the design constraints. The first concern was stabilizing the system. Lead filters were tried as a starting point. These are compensators with transfer functions that appear as Eq. (4).

$$H_{lead} = \frac{s + a}{s + b} \text{ where } b > a \quad (4)$$

This type of compensator increases the stability of the system by attracting the unstable poles to the left hand side of the s-plane. Several combinations of lead filters with real poles and zeros were tried with no success at stabilizing the system. The next step was to add a complex pole and a complex zero. Zeros attract poles, so in order to attract the unstable poles to the left half of the

s-plane, a complex zero was added near the imaginary axis. A complex pole was also placed farther away from the imaginary axis than the zero. When calculating the departure angle of the unstable poles in the right half plane, a pole that is in the left half plane and far away from the imaginary axis will have little effect on the departure angle of those unstable poles. The complex pole was placed at $-2 \pm 0.3i$ and the complex zero was placed at $-10 \pm 0.15i$. The compensator transfer function is shown in Eq. (5).

$$H(s) = \frac{1 + 0.98s + (0.49s)^2}{1 + 0.2s + (0.1s)^2} \quad (5)$$

When implementing this compensator, the root loci of the unstable system were pulled slightly toward the stable left half plane, but not enough to become stable. This is shown below in Fig. 4.

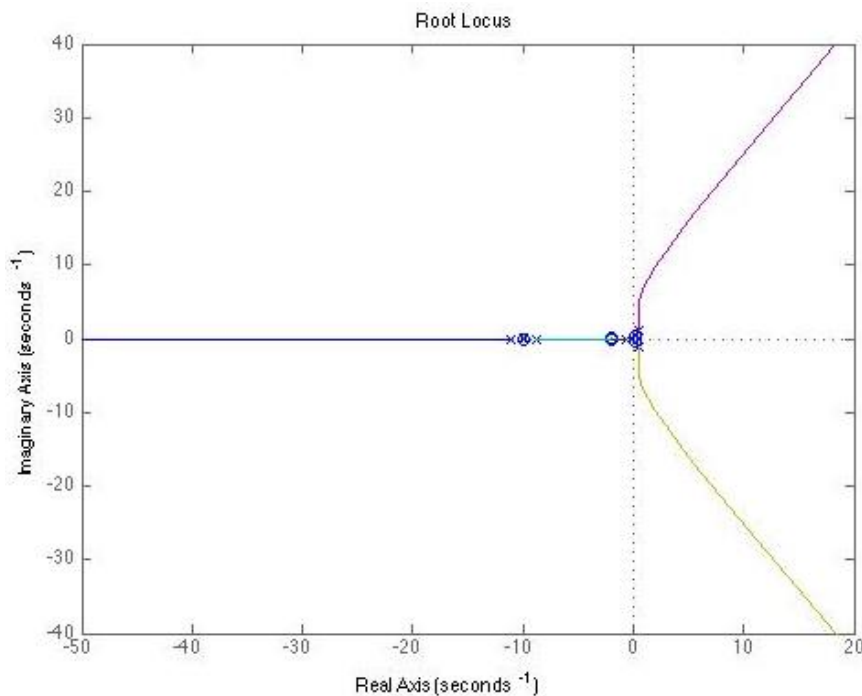


Fig. 4 Root Locus Diagram of Closed-Loop Transfer Function with Complex Pole and Zero Compensator Added

The step response, shown in Fig. 5, is clearly not stable. The amplitude still does not settle and the design constraints are not met.

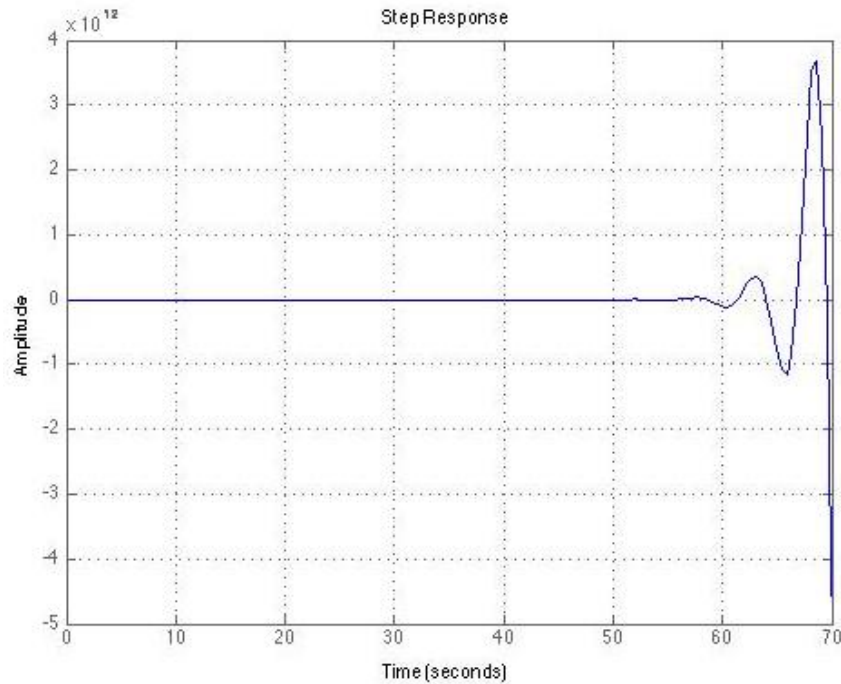


Fig. 5 Step Response of Closed-Loop Transfer Function with Complex Pole and Zero Compensator Added

To increase the stability of the system, the complex zero was moved closer to the imaginary axis, along the real axis. The pole was also moved further away from the imaginary axis, along the real axis. The complex zero is now at $-0.5 \pm 0.3i$. The complex pole is now at $-100 \pm 0.15i$. The new compensator transfer function is shown in Eq. (6).

$$H(s) = \frac{1 + 2.9s + (1.7s)^2}{1 + 0.02s + (0.01s)^2} \quad (6)$$

As shown by the step response in Fig.6, the system is now stable. However, the design constraints for rise time and settling time are not yet met.

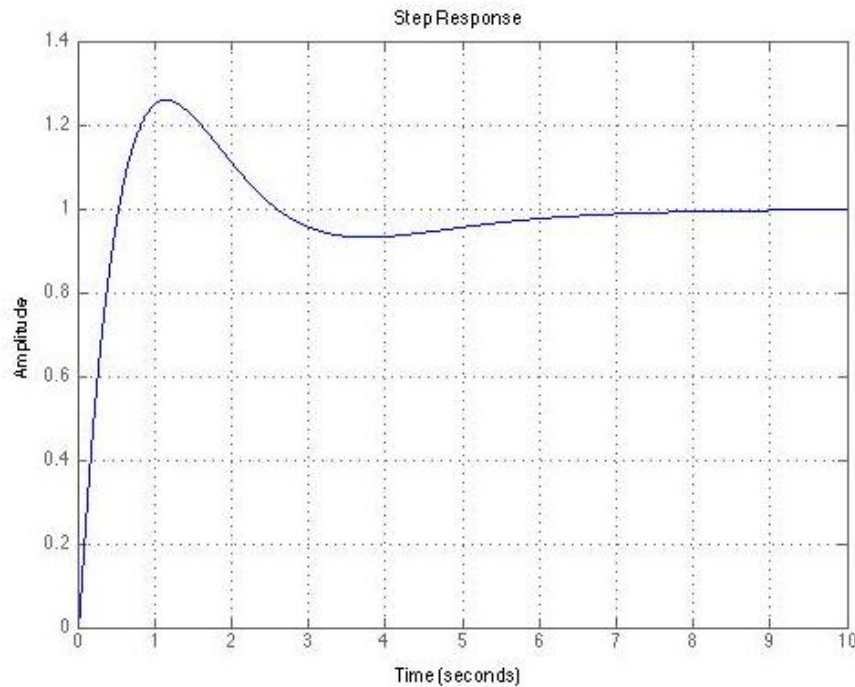


Fig. 6 Step Response of Closed Loop Transfer Function with Compensator Zero Moved Closer to Imaginary Axis and Compensator Pole Moved Farther from Imaginary Axis

Next, to decrease the rise time of the step response, a gain of 5 was added to the system. The new compensator transfer function is shown in Eq. (7).

$$H(s) = 5 * \frac{1 + 2.9s + (1.7s)^2}{1 + 0.02s + (0.01s)^2} \quad (7)$$

Figure 7 shows the step response of the closed loop transfer function with added gain. The rise time has decreased and the settling time has decreased.

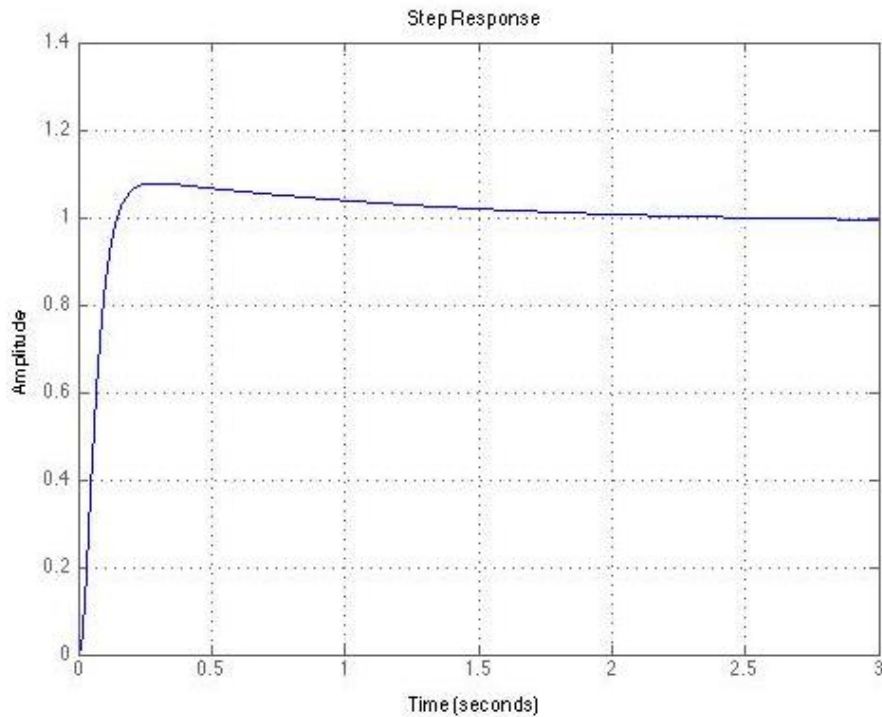


Fig. 7 Step Response of Closed Loop Transfer Function with Compensator and Gain

Finally, to decrease the settling time, the complex zero was moved closer to the real axis. The zero is now at $-0.5 \pm 0.05i$. The final compensator transfer function is shown in Eq. (8).

$$H(s) = 5 * \frac{1 + 4s + (2s)^2}{1 + 0.02s + (0.01s)^2} \quad (8)$$

III. Results

The design constraints of this assignment specify that the rise time of the step response must be less than 0.5 seconds, the step response must settle to within 10% of the desired position in 1 second, and settle to within 1% of the desired position within 5 seconds. In order to ensure that the response falls within these constraints, exponential envelopes were superimposed onto the step response graph. These envelopes were found by solving Eqs. (9), (10), (11), and (12) for alpha.

$$0.9 = 1 - e^{-\alpha(1)} \quad (9)$$

$$1.1 = 1 + e^{-\alpha(1)} \quad (10)$$

$$0.99 = 1 - e^{-\alpha(5)} \quad (11)$$

$$1.01 = 1 + e^{-\alpha(5)} \quad (12)$$

The following exponential envelopes, Eq. (13) and Eq. (14), were then superimposed onto the step function.

$$1 \pm e^{-2.302585*t} \quad (13)$$

$$1 \pm e^{-0.921034*t} \quad (14)$$

Figure 8 displays the final step response. The response is within the settling time design constraints. Using the MATLAB function “stepinfo”, the rise time, which is defined as the time it takes for the response to go from 10% to 90% of the steady state value, is 0.0629 seconds. This is also within the design constraints.

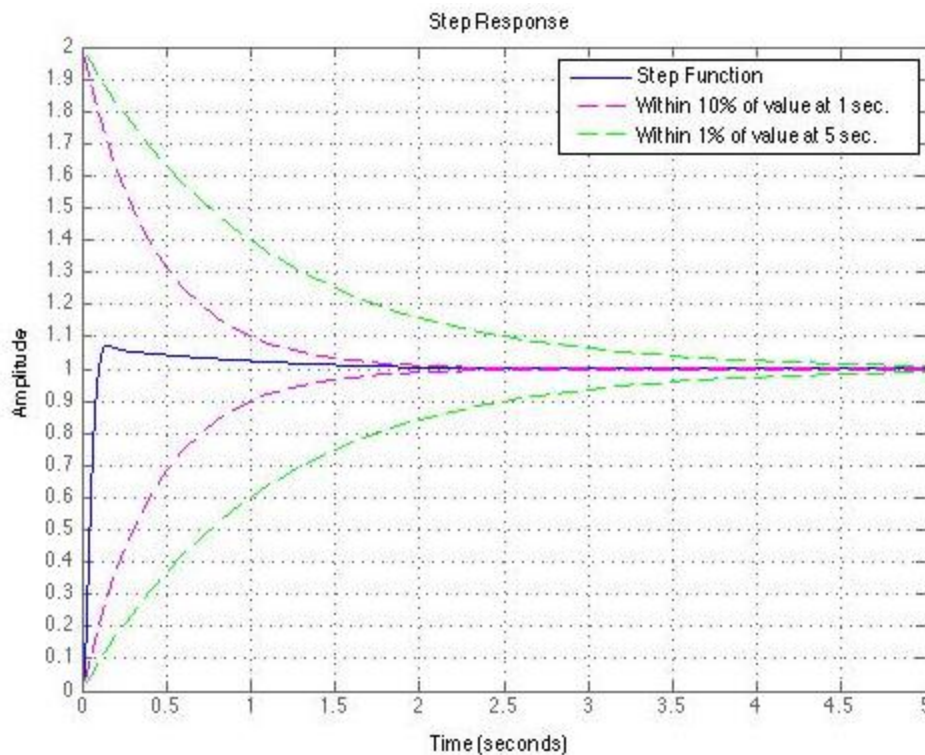


Fig. 8 Step Response with Final Compensation and Superimposed Design Constraint Envelopes

Figure 9 displays an enlarged view of the response at 5 seconds to show that the design constraints are met.

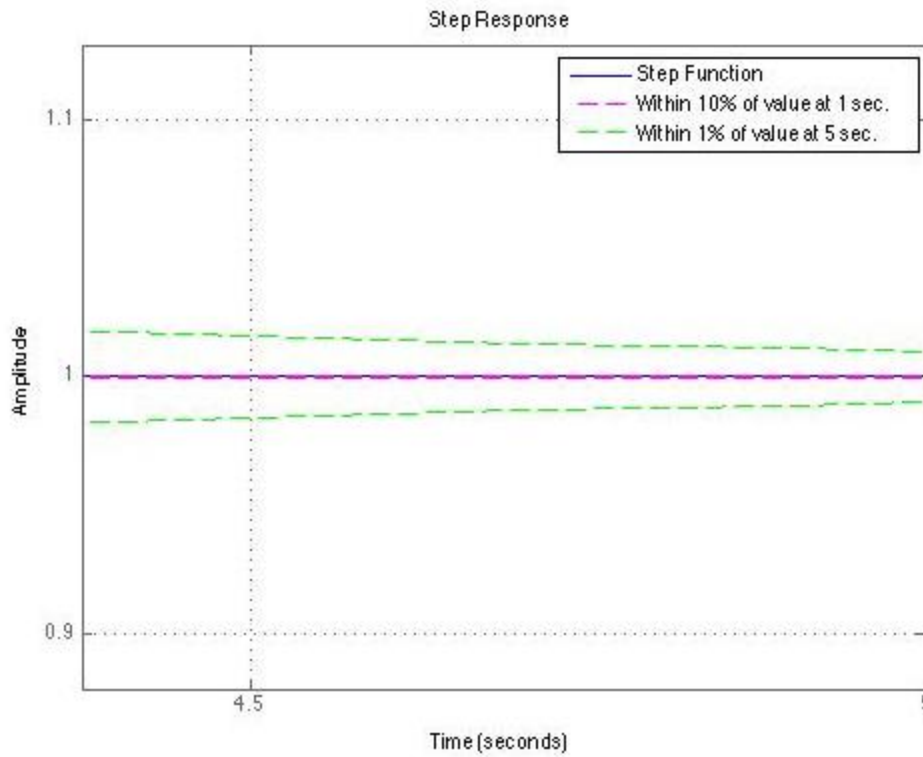


Fig. 9 Enlarged View of Step Response at 5 Seconds

Finally, Fig. 10 shows that the step response is settling on the desired value of 1.

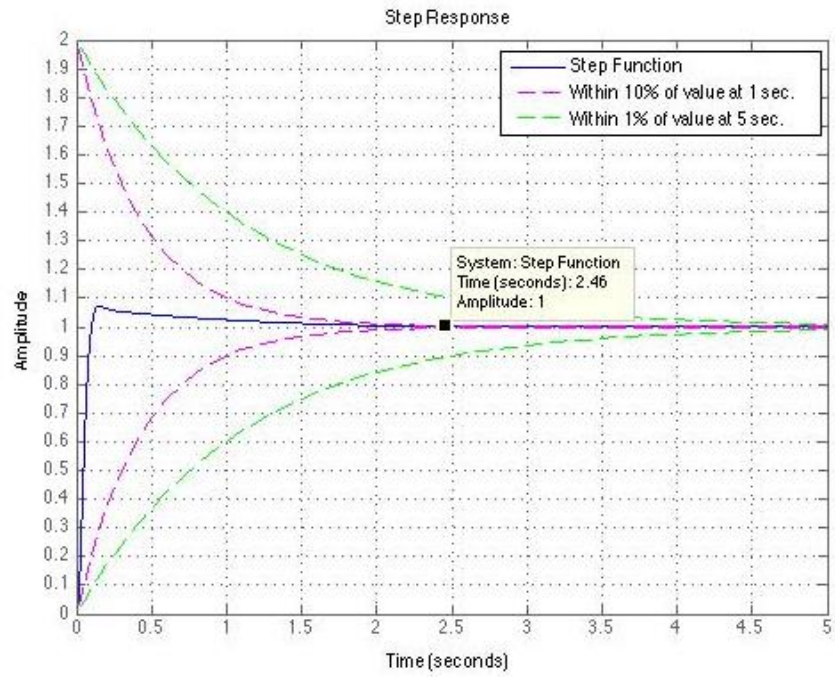


Fig. 10 Step Response Settles on Desired Value of 1

Appendix A – MATLAB Code

```
clear all
s=tf('s');

pole1=0.1228+0.3350i;
pole2=0.1228-0.3350i;

%Original Transfer Function
G=1/(s*(s-pole1)*(s-pole2));
H=1;
TF=(G*H)/(1+G*H);
figure(1)
rlocus(TF)
figure(2)
step(TF)
grid on

% 1) Add complex pole and zero
% G=1/(s*(s-pole1)*(s-pole2));
% H=(1+0.98*s+(0.49*s)^2)/(1+0.2*s+(0.1*s)^2)
% TF=(G*H)/(1+G*H)
% figure(3)
% rlocus(TF)
% figure(4)
% step(TF)
% grid on

% 2) Move zero closer to the imaginary axis
% G=1/(s*(s-pole1)*(s-pole2));
% H=(1+1.8*s+(0.96*s)^2)/(1+0.2*s+(0.1*s)^2)
% TF=(G*H)/(1+G*H)
% figure(3)
% subplot(2,1,1)
% rlocus(TF)
% subplot(2,1,2)
% step(TF)

% % 3) Move zero closer to the imaginary axis
% G=1/(s*(s-pole1)*(s-pole2));
% H=(1+2.4*s+(1.3*s)^2)/(1+0.2*s+(0.1*s)^2)
% TF=(G*H)/(1+G*H)
% figure(4)
% subplot(2,1,1)
% rlocus(TF)
% subplot(2,1,2)
% step(TF)
```

```

% 4) Move zero closer to the imaginary axis
% G=1/(s*(s-pole1)*(s-pole2));
% H=(1+2.9*s+(1.7*s)^2)/(1+0.2*s+(0.1*s)^2)
% TF=(G*H)/(1+G*H)
% figure(5)
% rlocus(TF)
% figure(6)
% step(TF)
% grid on

% 5) Move zero closer to the imaginary axis
% G=1/(s*(s-pole1)*(s-pole2));
% H=(1+3.1*s+(2.8*s)^2)/(1+0.2*s+(0.1*s)^2)
% TF=(G*H)/(1+G*H)
% figure(6)
% subplot(2,1,1)
% rlocus(TF)
% subplot(2,1,2)
% step(TF)

% 6) Go back to step 4 zero and move pole further away from imaginary axis
% along real axis
% G=1/(s*(s-pole1)*(s-pole2));
% H=(1+2.9*s+(1.7*s)^2)/(1+0.04*s+(0.02*s)^2)
% TF=(G*H)/(1+G*H)
% figure(7)
% subplot(2,1,1)
% rlocus(TF)
% subplot(2,1,2)
% step(TF)

% 7) Move pole further away from imaginary axis along real axis
% G=1/(s*(s-pole1)*(s-pole2));
% H=(1+2.9*s+(1.7*s)^2)/(1+0.02*s+(0.01*s)^2)
% TF=(G*H)/(1+G*H)
% figure(7)
% rlocus(TF)
% figure(8)
% step(TF)
% grid on

% 8) Step 7 did not change much, increase gain
% G=1/(s*(s-pole1)*(s-pole2));
% H=5*(1+2.9*s+(1.7*s)^2)/(1+0.02*s+(0.01*s)^2)
% TF=(G*H)/(1+G*H)
% figure(9)
% rlocus(TF)
% figure(10)
% step(TF)
% grid on

```

```

% 9) Move zero closer to the real axis
G=1/(s*(s-pole1)*(s-pole2));
H=5*(1+4*s+(2*s)^2)/(1+0.02*s+(0.01*s)^2);
TF=(G*H)/(1+G*H)
figure(11)
rlocus(TF)
figure(12)
step(TF)
grid on
axis([0 5 0 2])
set(gca,'Xtick',0:0.5:5)
set(gca,'Ytick',0:0.1:2)

hold on

tmat=[0:0.1:5];
env=zeros(50,1);
env2=zeros(50,1);
env3=zeros(50,1);
env4=zeros(50,1);
x=1
for t=0:0.1:5
    env(x,1)=1-exp(-2.302585*t);
    env2(x,1)=1+exp(-2.302585*t);
    env3(x,1)=1-exp(-0.921034*t);
    env4(x,1)=1+exp(-0.921034*t);
    x=x+1;
end

hold on
plot(tmat,env2,'--m')
plot(tmat,env3,'--g')
plot(tmat,env,'--m')
plot(tmat,env4,'--g')
legend('Step Function','Within 10% of value at 1 sec.','Within 1% of value at
5 sec.')

```