Control
Lab Two

Prepared By

Ahmed Aly Gamal El-Din El-Ghannam

Electronics and Communications - Level 4
ID: 19015292

November 2023

Contents

Listings

1 Introduction

This report showcases the results of simulating and controlling a satellite-tracking antenna system as shown in figure 1 using both Simulink and MatLAB. All the simulation files and MatLAB codes used to produce this result can be found in the lab's Github Repository.

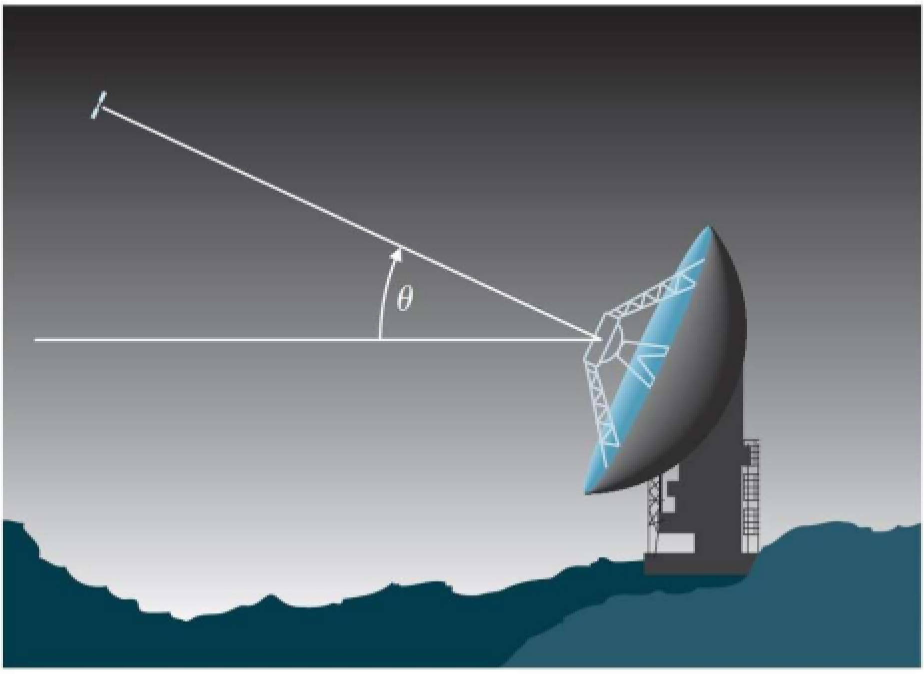

Figure 1: Satellite-Tracking Antenna System

2 Evaluating the Closed-loop Transfer Function

The equation of motion is given as:

$$\mathrm{J}\ddot{\theta} + \mathrm{B}\dot{\theta} = \mathrm{T_c} \tag{1}$$

where $\mathrm{J} = 600000 kg.m^2$, $\mathrm{B} = 20000 N.m.sec$, and the applied torque $\mathrm{T_c}$ is defined as:

$$\mathrm{T_c} = \mathrm{K}(\theta_\mathrm{r} - \theta) \tag{2}$$

and K is the feedback gain.

By equating equation 1 and 2, therefore:

$$\mathrm{J}\ddot{\theta} + \mathrm{B}\dot{\theta} = \mathrm{K}(\theta_r\, \theta) \tag{3}$$

By applying Laplace transform to equation 3:

$$\mathrm{JS^2}\theta + \mathrm{BS}\theta = \mathrm{K}\theta_\mathrm{r} - \mathrm{K}\theta \tag{4}$$

Through mathematical manipulation, the transfer function—where θ is the output and θ_r is the input—can be represented as follows:

$$\mathrm{H(s)} = \frac{\theta}{\theta_\mathrm{r}} = \frac{\mathrm{K}}{\mathrm{JS^2 + BS + K}} \tag{5}$$

Using the characteristic equation,

$$\mathrm{JS^2 + BS + K} = 0 \tag{6}$$

both ω_n and ζ can be deduced:

$$\omega_\mathrm{n} = \sqrt{\frac{\mathrm{K}}{\mathrm{J}}} \tag{7}$$

$$\zeta = \frac{\mathrm{B}}{2\sqrt{\mathrm{KJ}}} \tag{8}$$

3 State-Space Representation Using MATLAB

By entering equation 6 on MATLAB with K = 1 as shown in code snippet 5, the state-space representation of the system is as shown in figure 3.

```
%% Prepreparations
clear
close all
clc

%% System Parameters
J = 600e3;
B = 20e3;
K = 1;

%% Transfer Function (Input: theta_r --- Output: theta)
TF_thetaOverThetar = tf([0 0 K], [J B K]);

%% Converting the Transfer Function to the State-Space Model
SS_thetaOverThetar = ss(TF_thetaOverThetar)
size(SS_thetaOverThetar)
```

Code Snippet 1: State-Space Representation of the System System

```
SS_thetaOverThetar =

  A = 
              x1         x2
   x1   -0.03333  -0.001707
   x2  0.0009766          0

  B = 
            u1
   x1  0.03125
   x2        0

  C = 
            x1       x2
   y1        0  0.05461

  D = 
       u1
   y1   0

Continuous-time state-space model.

State-space model with 1 outputs, 1 inputs, and 2 states.
```

Figure 2: State-Space Representation of the System System

4 Finding Different Values of K Using MATLAB

4.1 Maximum Value for K

In order to find the maximum value for K to keep the closed-loop system stable, the characteristic equation is differentiated and equated to 0. Code snippet 2 demonstrates this and figure 3 shows said value.

```
%% Maximum Value for K to Have a Stable Closed-Loop System
% Characteristic Eqn: JS^2 + BS + K = 0
syms K S;
K = -(J * (S^2)) - (B * S)

S = solve(diff(K, S)==0)
K_max = subs(K)
```

Code Snippet 2: State-Space Representation of the System System

```
K =

- 600000*S^2 - 20000*S


S =

-1/60


K_max =

500/3
```

Figure 3: Maximum Value of K

4.2 For $M_p < 10\%$

By looping over multiple values of K, the maximum value for K to produce an Overshoot less than 10% can be found. Code snippet 3 demonstrates this and figure 3 shows said value.

```
%% Finding Value for K to Have Mp < 10%
SS_variables = stepinfo(SS_thetaOverThetar)

for K = 1:0.1:1006
        TF_thetaOverThetar = tf([0 0 K], [J B K]);
        SS_thetaOverThetar = ss(TF_thetaOverThetar);
        SS_variables = stepinfo(SS_thetaOverThetar);
        if (SS_variables.Overshoot <= 10)
            K_Mp = K;
        end
end
```

Code Snippet 3: Maximum Value for K for $M_p < 10\%$

```
K_Mp =

  476.9200
```

Figure 4: Maximum Value for K for $M_p < 10\%$

4.3 For $t_r < 80s$

By looping over multiple values of K, the maximum value for K to produce a rise time less than 80 seconds can be found: all values of K more than or equal 381 satisfy this condition. Code snippet 4 demonstrates this and figure 5 shows said threshold.

```
%% Finding Value for K to Have tr < 80 sec
for K = 1006:-0.1:1
        TF_thetaOverThetar = tf([0 0 K], [J B K]);
        SS_thetaOverThetar = ss(TF_thetaOverThetar);
        SS_variables = stepinfo(SS_thetaOverThetar);
        if (SS_variables.RiseTime <= 80)
            K_tr = K;
        end
end
```

Code Snippet 4: Values of K for $t_r < 80s$

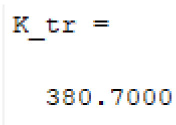

Figure 5: Values of K for $t_r < 80s$

5 System Plots Using MATLAB

In this section, the following plots will graphed at different values of K. In addition, the steady state error will be calculated as well. Code snippet 5 uses the function present in code snippet 6 to accomplish this goal.

```
%% Step Response + Zeros/Poles + Steady State Error
% @ K = 200
SS_errorAtK__200 = FN_plotSS_ZP_SSerror(200, J, B)
% @ K = 400
SS_errorAtK__400 = FN_plotSS_ZP_SSerror(400, J, B)
% @ K = 1000
SS_errorAtK__1000 = FN_plotSS_ZP_SSerror(1000, J, B)
% @ K = 2000
SS_errorAtK__2000 = FN_plotSS_ZP_SSerror(2000, J, B)
```

Code Snippet 5: System Plots at Different Values of K

```
function SS_error = FN_plotSS_ZP_SSerror(K, J, B)
    TF_thetaOverThetar = tf([0 0 K], [J B K]);
    figure
    step(TF_thetaOverThetar)
    figure
    pzplot(TF_thetaOverThetar)
    grid on
    [y, t] = step(TF_thetaOverThetar);
    SS_error = abs(1 - y(end));
end
```

Code Snippet 6: Function to Graph System Plots at Different Values of K

5.1 Step Response

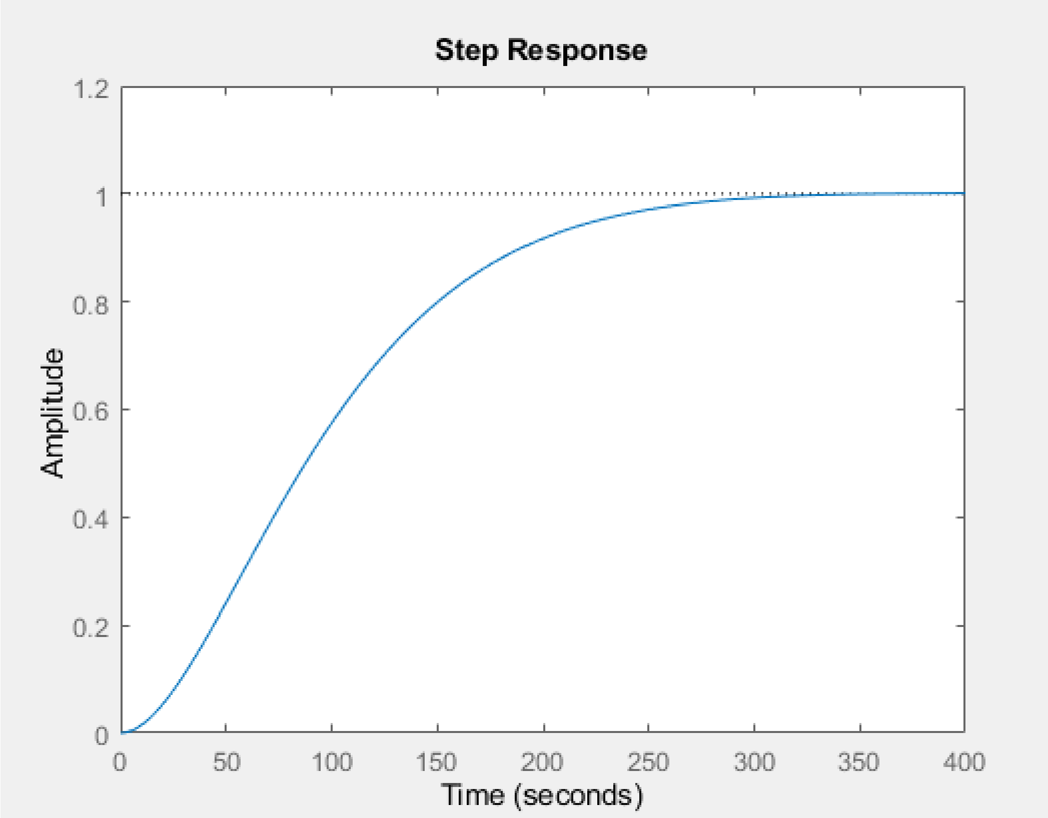

Figure 6: Step Response at K = 200

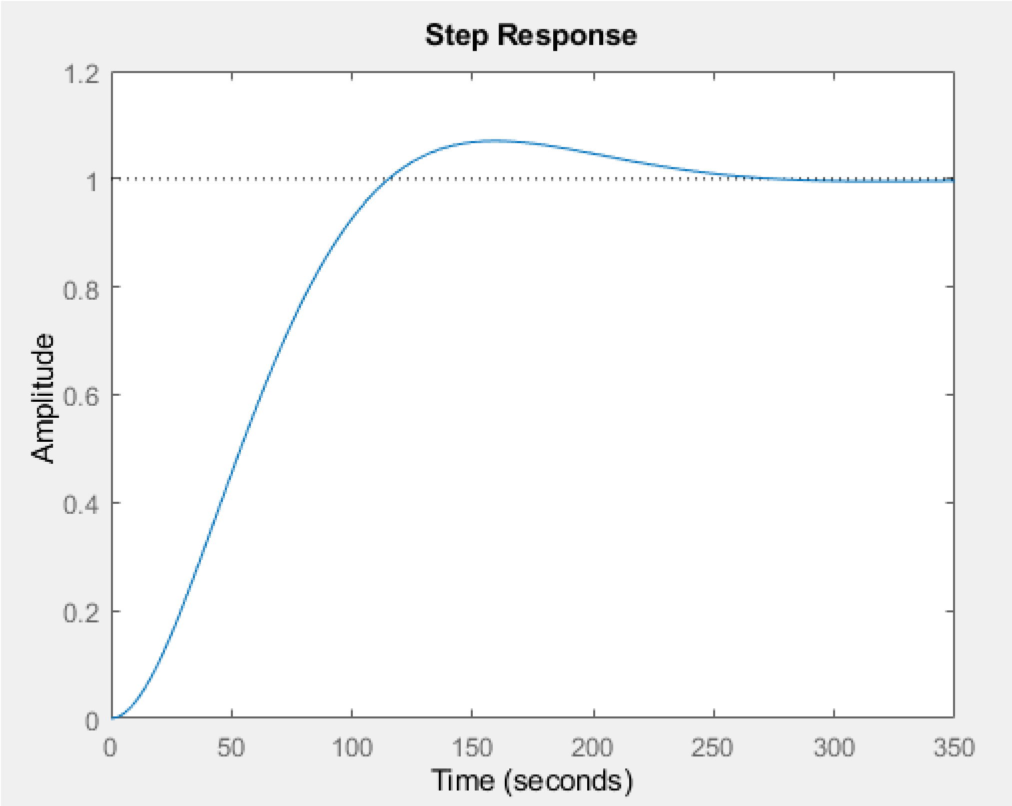

Figure 7: Step Response at K = 400

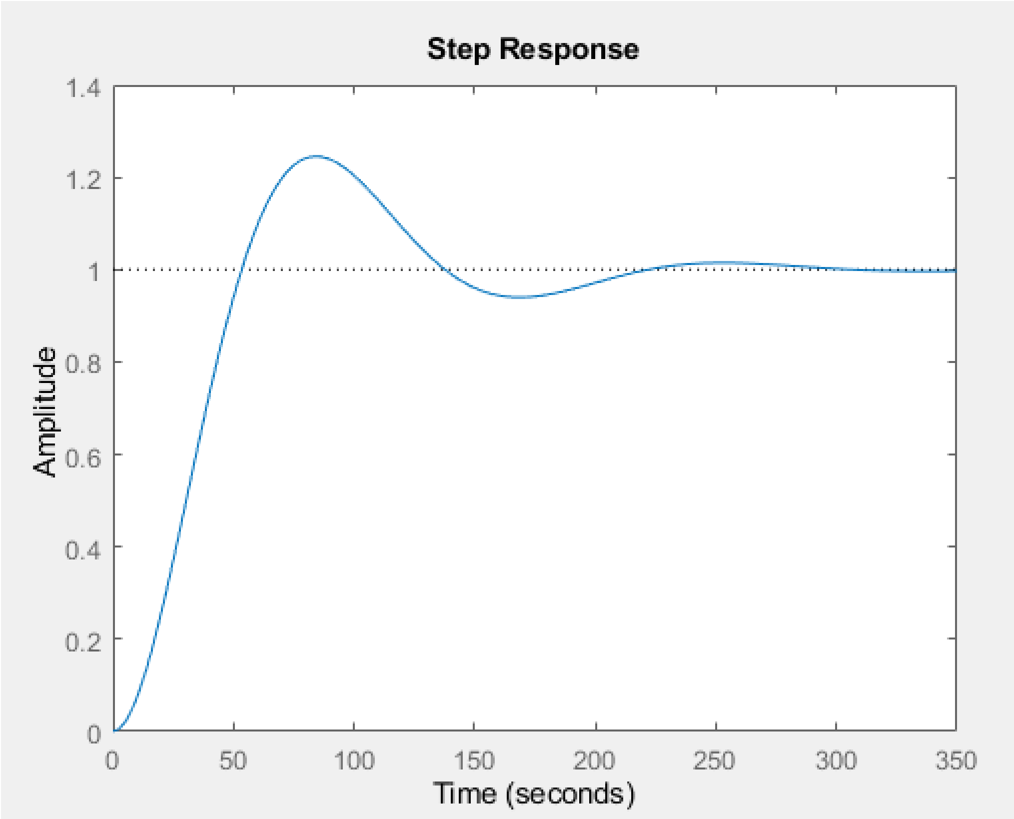

Figure 8: Step Response at K = 1000

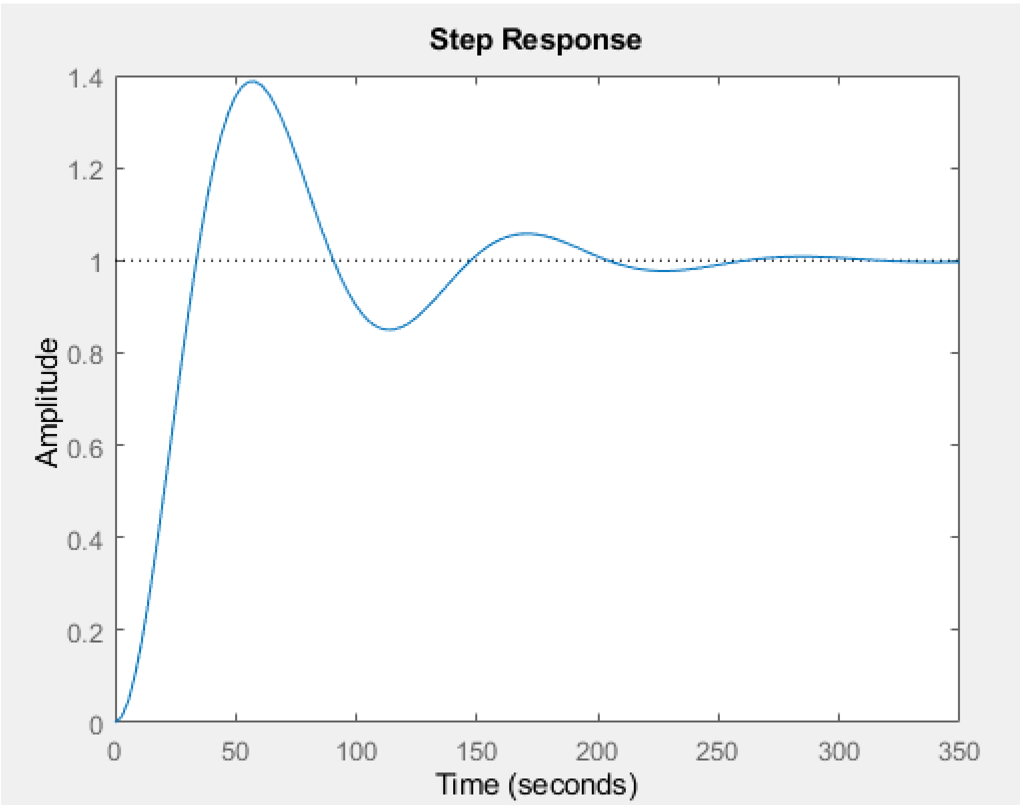

Figure 9: Step Response at K = 2000

5.2 Pole–Zero Map

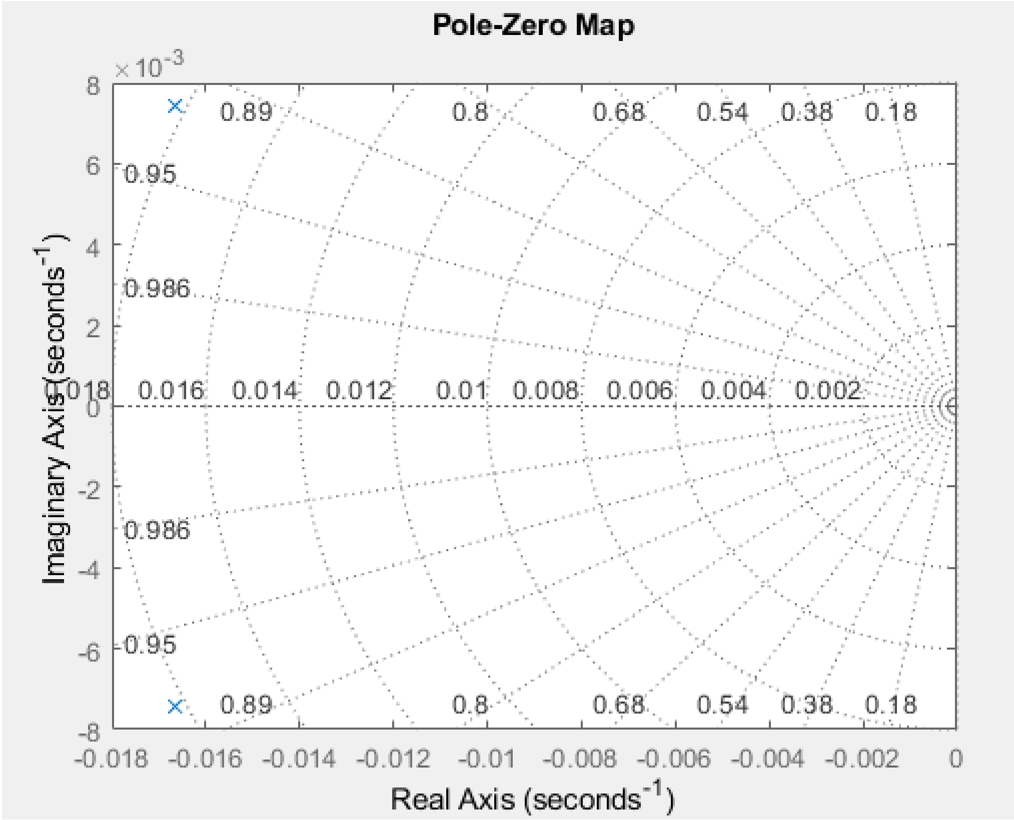

Figure 10: Pole–Zero Map at K = 200

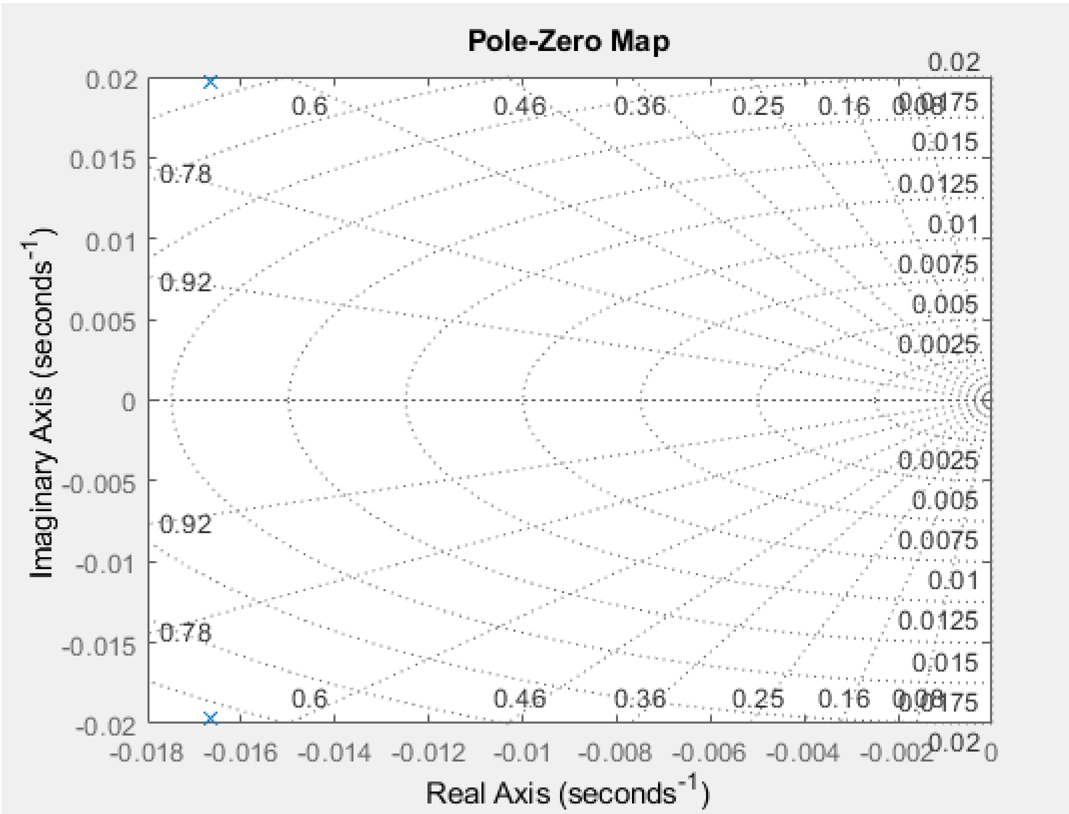

Figure 11: Pole–Zero Map at K = 400

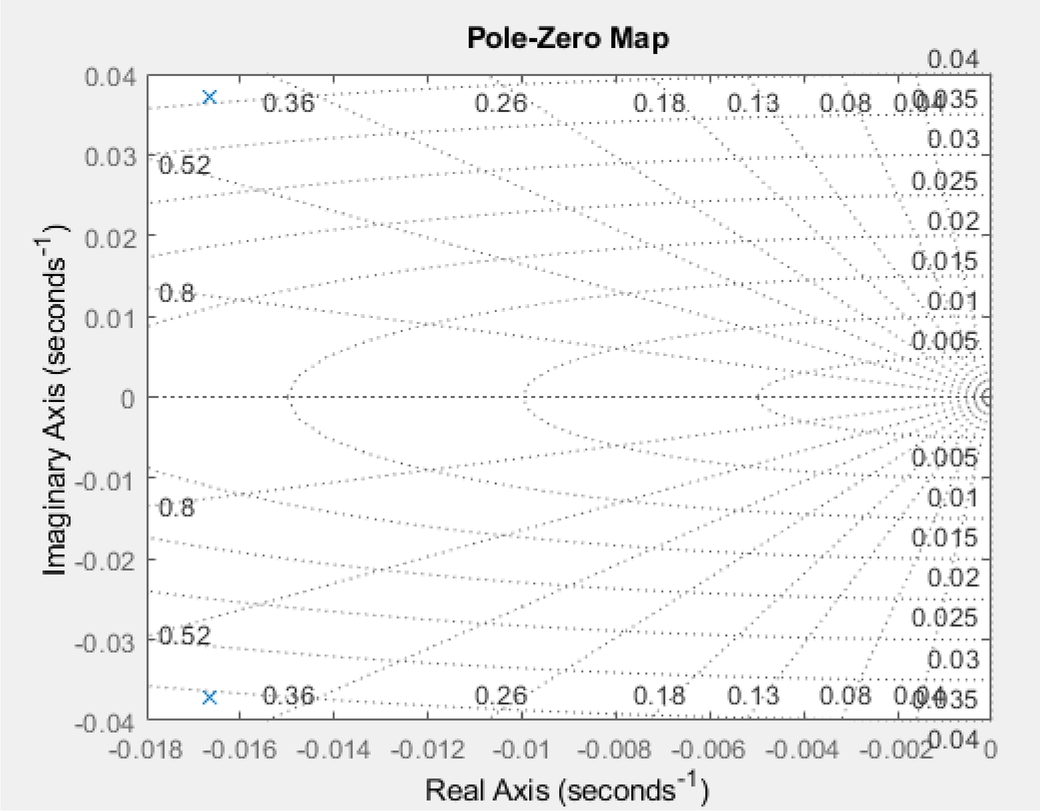

Figure 12: Pole–Zero Map at K = 1000

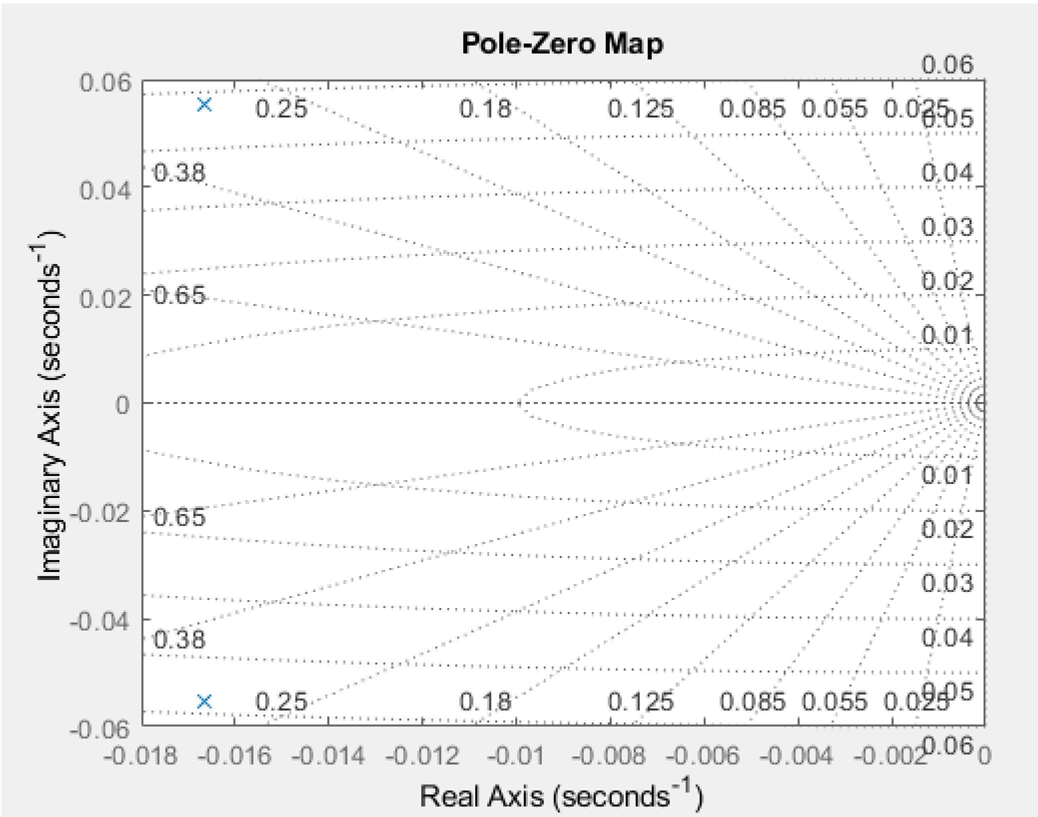

Figure 13: Pole–Zero Map at K = 2000

5.3 Steady State Errors

```
SS_errorAtK__200 =

   5.1400e-04


SS_errorAtK__400 =

    0.0049


SS_errorAtK__1000 =

    0.0033


SS_errorAtK__2000 =

   8.7153e-04
```

Figure 14: Steady State Errors at Different Values of K

6 Adding Poles and Zeros on Simulink

6.1 Adding Poles

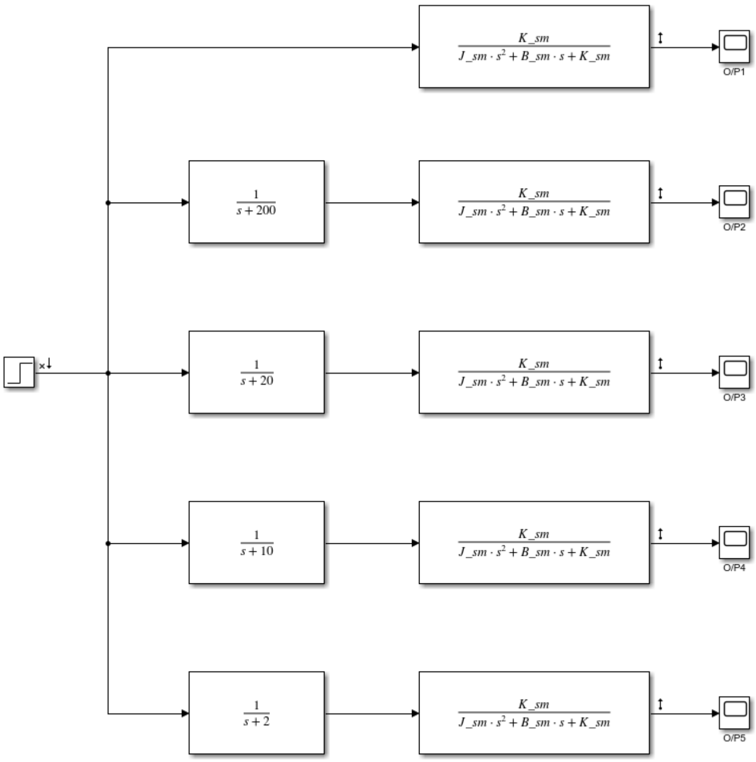

Figure 15: Adding Poles of Different Values to Existing System

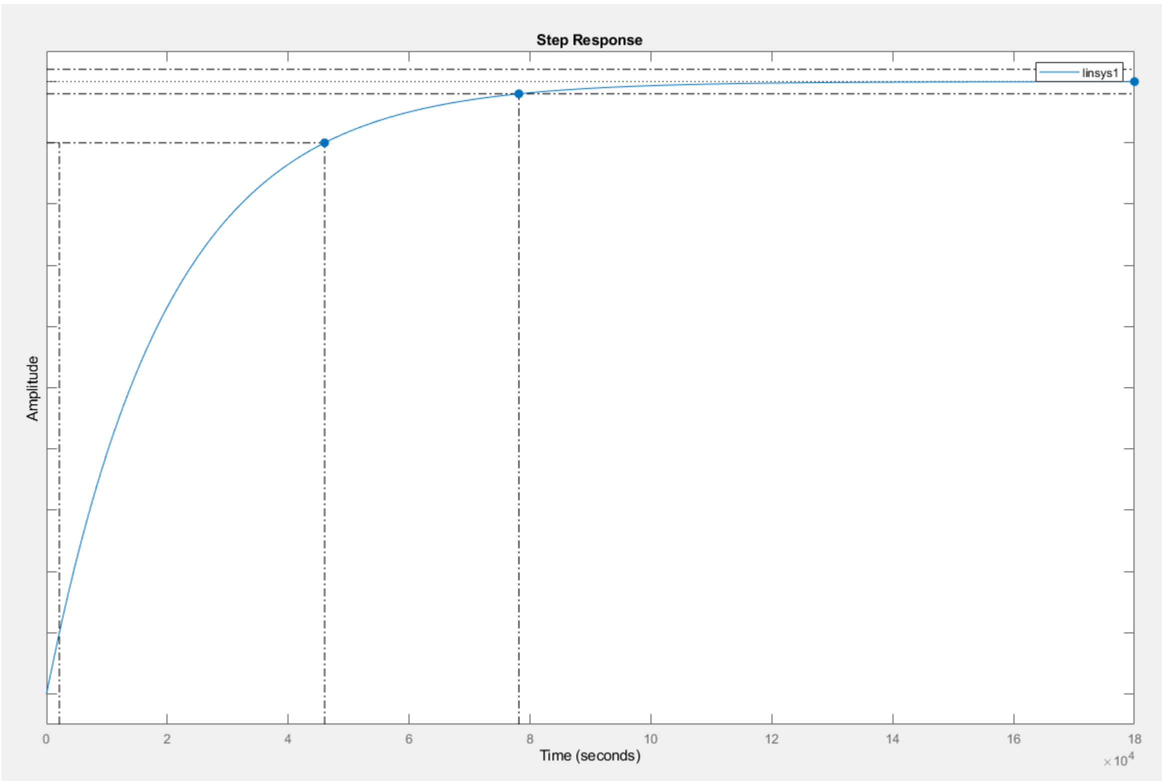

Figure 16: Normalized Step Response of the System After Adding Poles

6.2 Adding Zeros

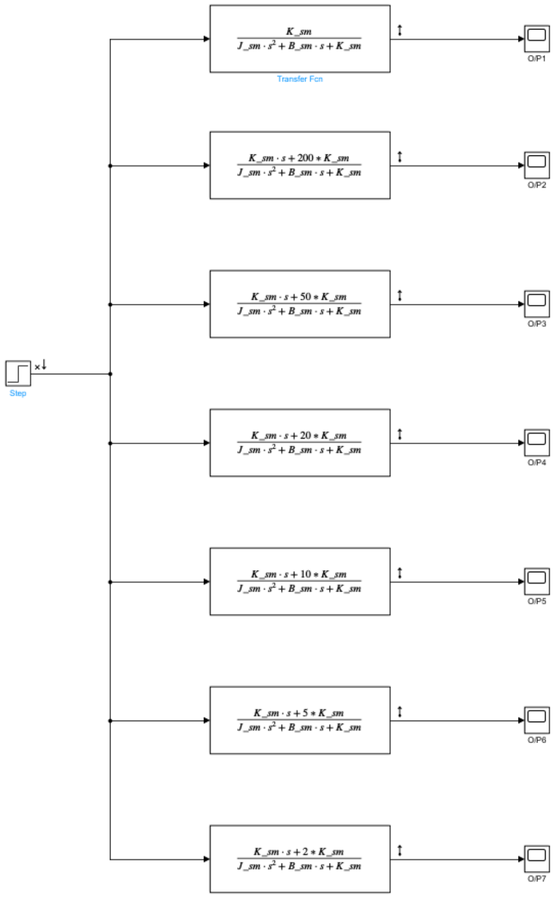

Figure 17: Adding Zeros of Different Values to Existing System

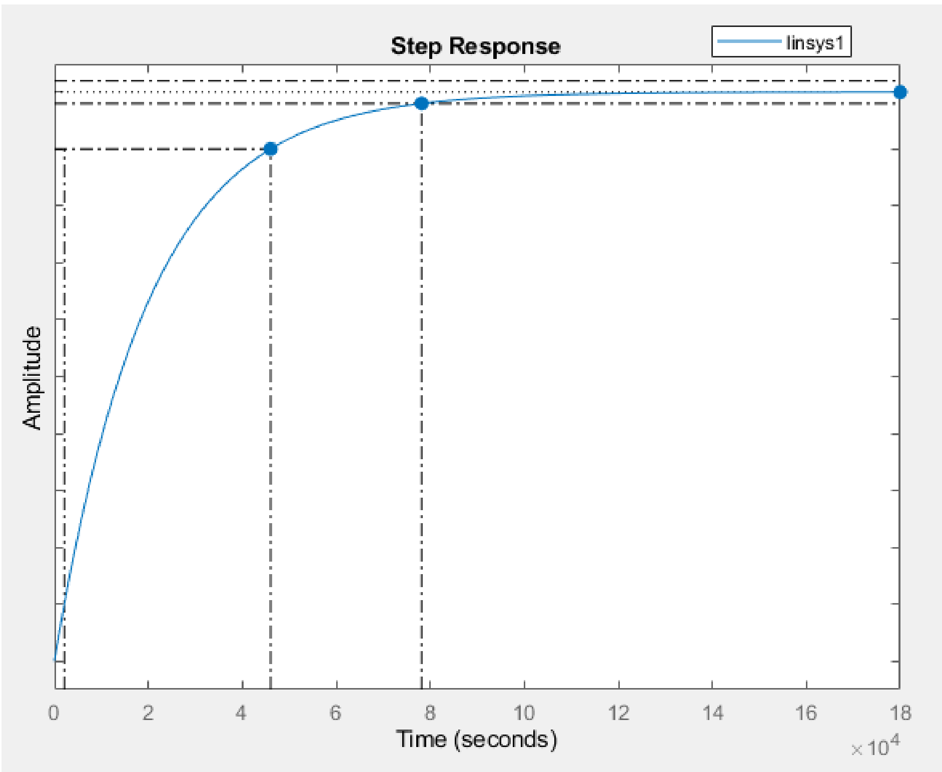

Figure 18: Normalized Step Response of the System After Adding Zeros