Noe Lepez Da Silva Duarte

AE 4610 – Lab 1: Overview of MATLAB and SIMULINK

1) i) Eigenvectors:

$$\begin{matrix} 0.0314 - 0.0618i & 0.0314 + 0.0618i & 0.9995 + 0.0000i & 0.9995 + 0.0000i \\ 0.9976 + 0.0000i & 0.9976 + 0.0000i & -0.0300 + 0.0005i & -0.0300 - 0.0005i \\ -0.0005 + 0.0040i & -0.0005 - 0.0040i & 0.0001 - 0.0000i & 0.0001 + 0.0000i \\ 0.0026 - 0.0024i & 0.0026 + 0.0024i & -0.0004 - 0.0016i & -0.0004 + 0.0016i \end{matrix}$$

Eigenvalues: -0.9005 + 0.7293i, -0.9005 - 0.7293i, -0.0095 + 0.0528i, -0.0095 - 0.0528i

ii)

$$\frac{-4 s^2 - 3.256 s - 0.07952}{s^4 + 1.82 s^3 + 1.38 s^2 + 0.0308 s + 0.003864}$$

iii)

-0.9005 + 0.7293i
-0.9005 - 0.7293i
-0.0095 + 0.0528i
-0.0095 - 0.0528i

The poles and eigenvalues are the same.

iv)

$$\begin{bmatrix} -1.8200 & -1.3800 & -0.0308 & -0.0039 \\ 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix}$$

The state space model gotten from MATLAB and the one provided in the question are different. That is because the state-space representation is a mathematical model of a physical system. As such, while the numbers inside of the matrix may be different, they still represent the same systems and inputs.

Noe Lepez Da Silva Duarte

v)

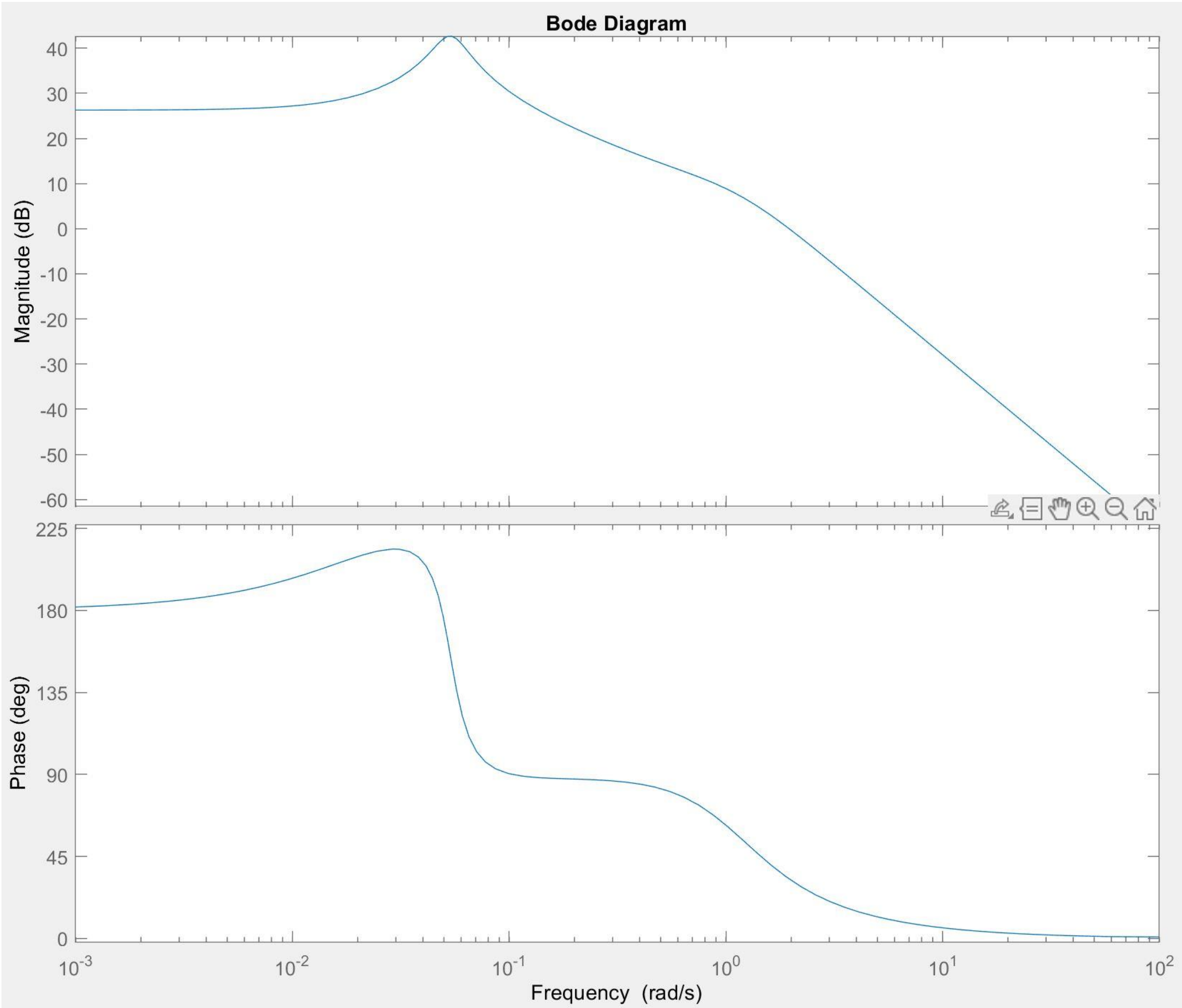

Figure 1. Bode plot of G1

vi)

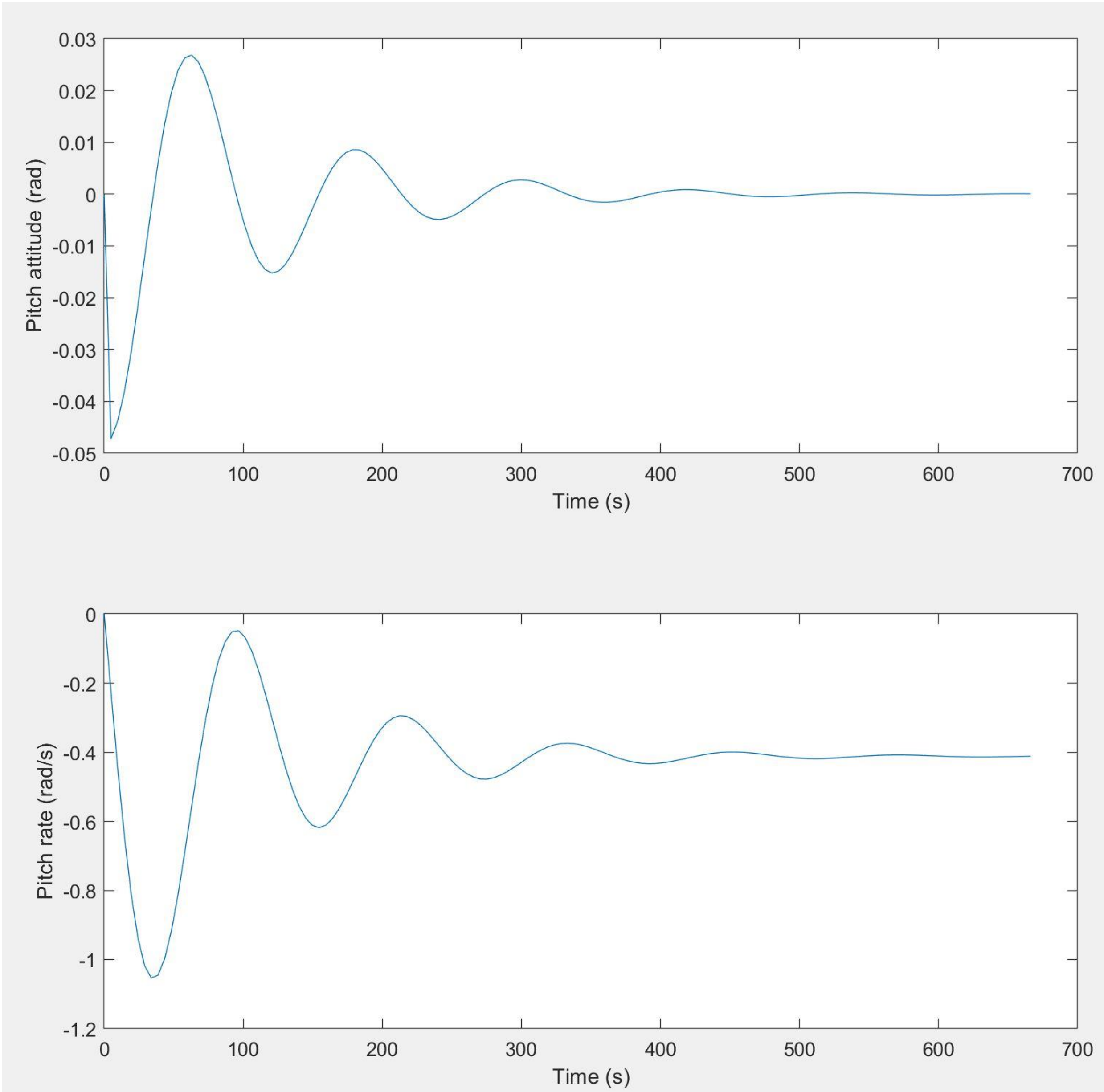

Figure 2. Step response of pitch attitude and pitch rate responses to a step elevator input of 0.02 rad

Noe Lepez Da Silva Duarte

```
%% Question 1
A = [-0.02 0.1 0 -32.2 ; -0.040 -0.8 180 0 ; 0 -0.003 -1.0 0 ; 0 0 1 0];
B = [ 0 ; -8 ; -4 ; 0];
C = [0 0 0 1];
D=0;

% Part i
[V,S]=eig(A)

% Part ii
[num,den]=ss2tf(A,B,C,D);
G1=tf(num,den)

% Part iii
[z, p, k] = zpkdata(G1)

% Part iv
[A2, B2, C2, D2]=tf2ss(num,den)

% Part v
bode(G1)

% Part vi
C= [0 0 1 0; 0 0 0 1];
D= [0 ; 0];
G3 = ss(A,B,C,D);
[res,t] = step(0.02*G3);

set(0,'defaultAxesFontSize', 10);
subplot(2,1,1)
plot(t, res(:,1))
xlabel("Time (s)")
ylabel("Pitch attitude (rad)")
hold on
subplot(2,1,2)
plot(t, res(:,2))
xlabel("Time (s)")
ylabel("Pitch rate (rad/s)")
```

Figure 3. MATLAB code for question 1

2)

```
3.5363 (s+0.2545)
-----------------
      (s+1)
```

K = 3.5363

a = 0.2545

b = 1

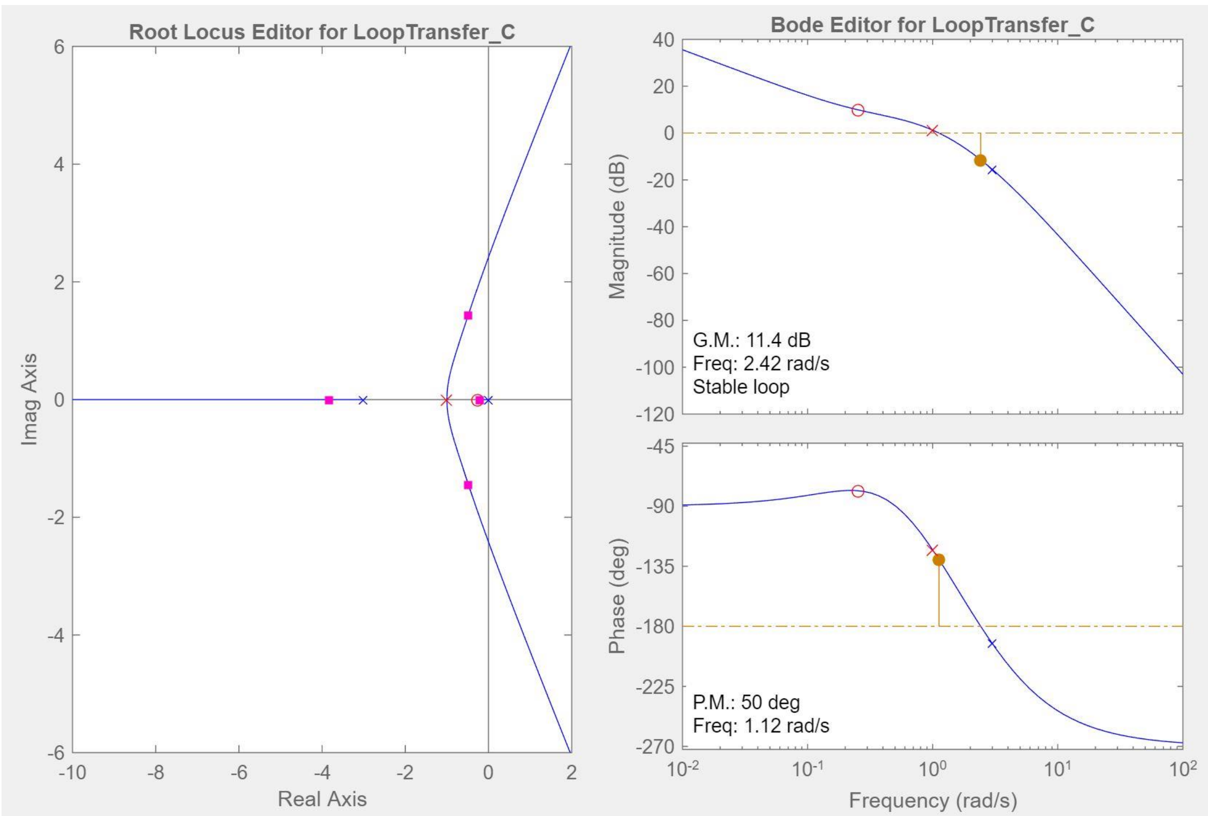

Figure 4. Root locus and bode plot of compensator

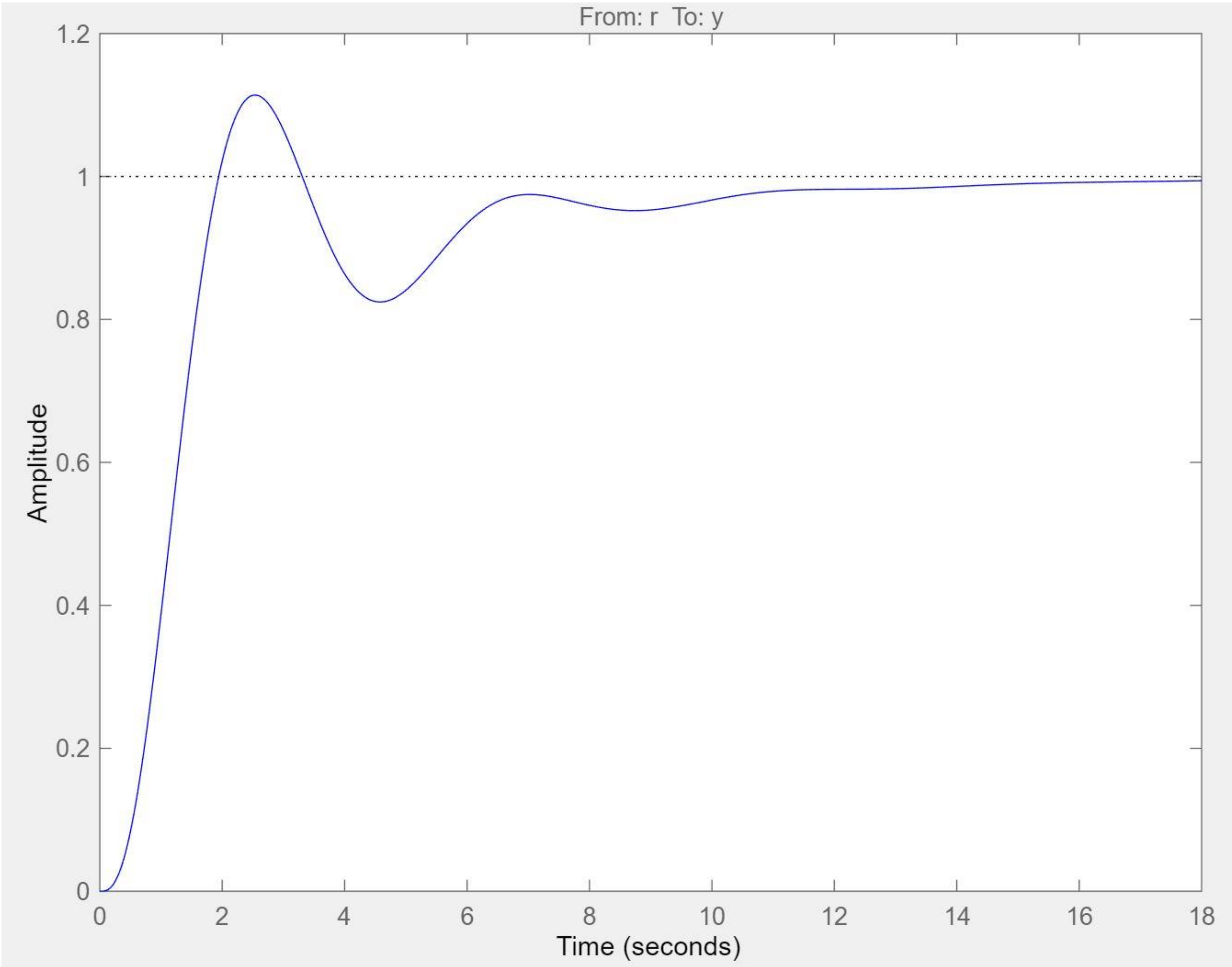

Figure 5. Step input response of system with compensator

```
%% Question 2

G=tf([2] , [1 4 3 0]);
C=tf([1 0.5] , [1 1]);

rltool(G,C)
```

Figure 6. Code for question 2

3)

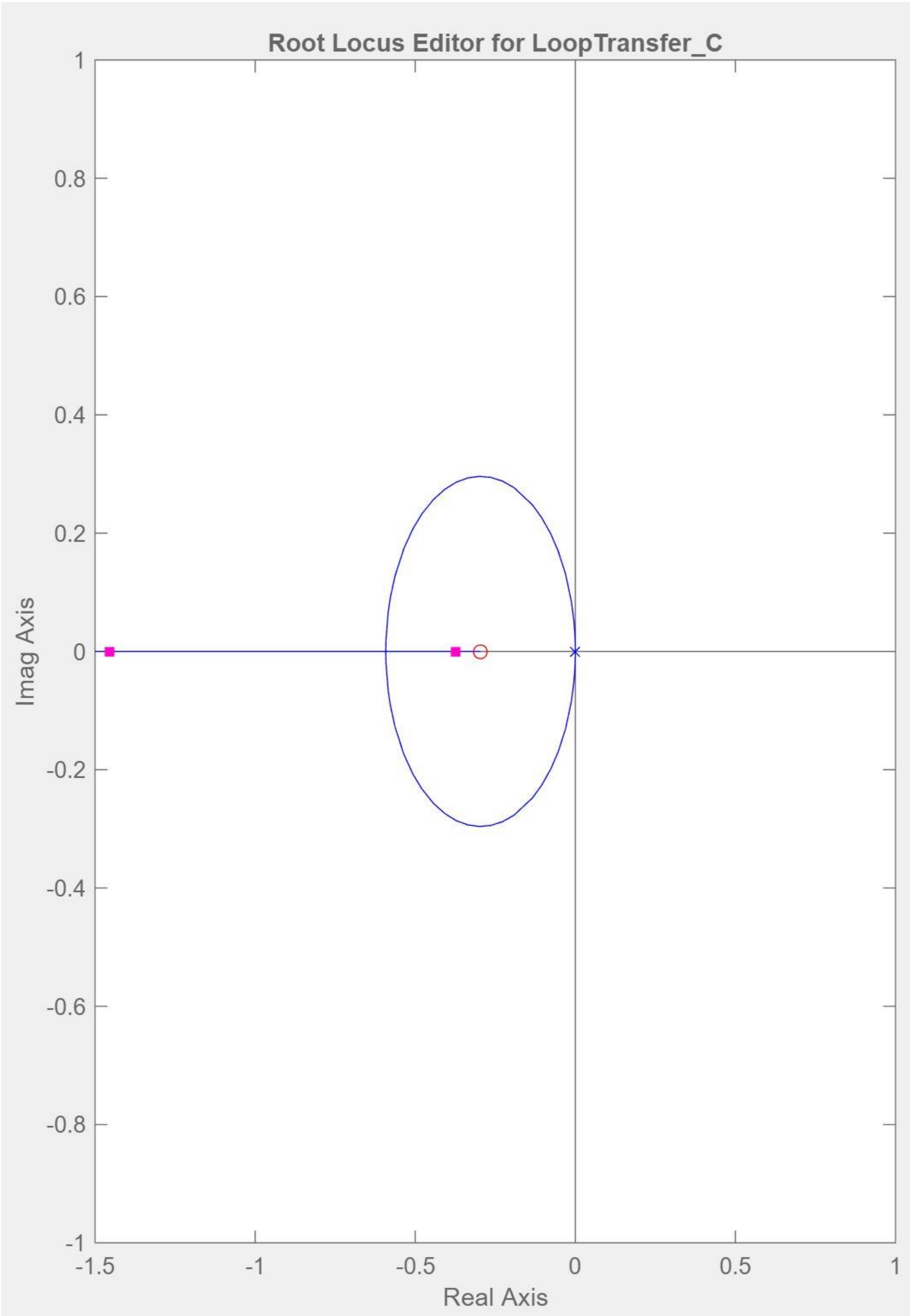

Figure 7. Root locus plot of system

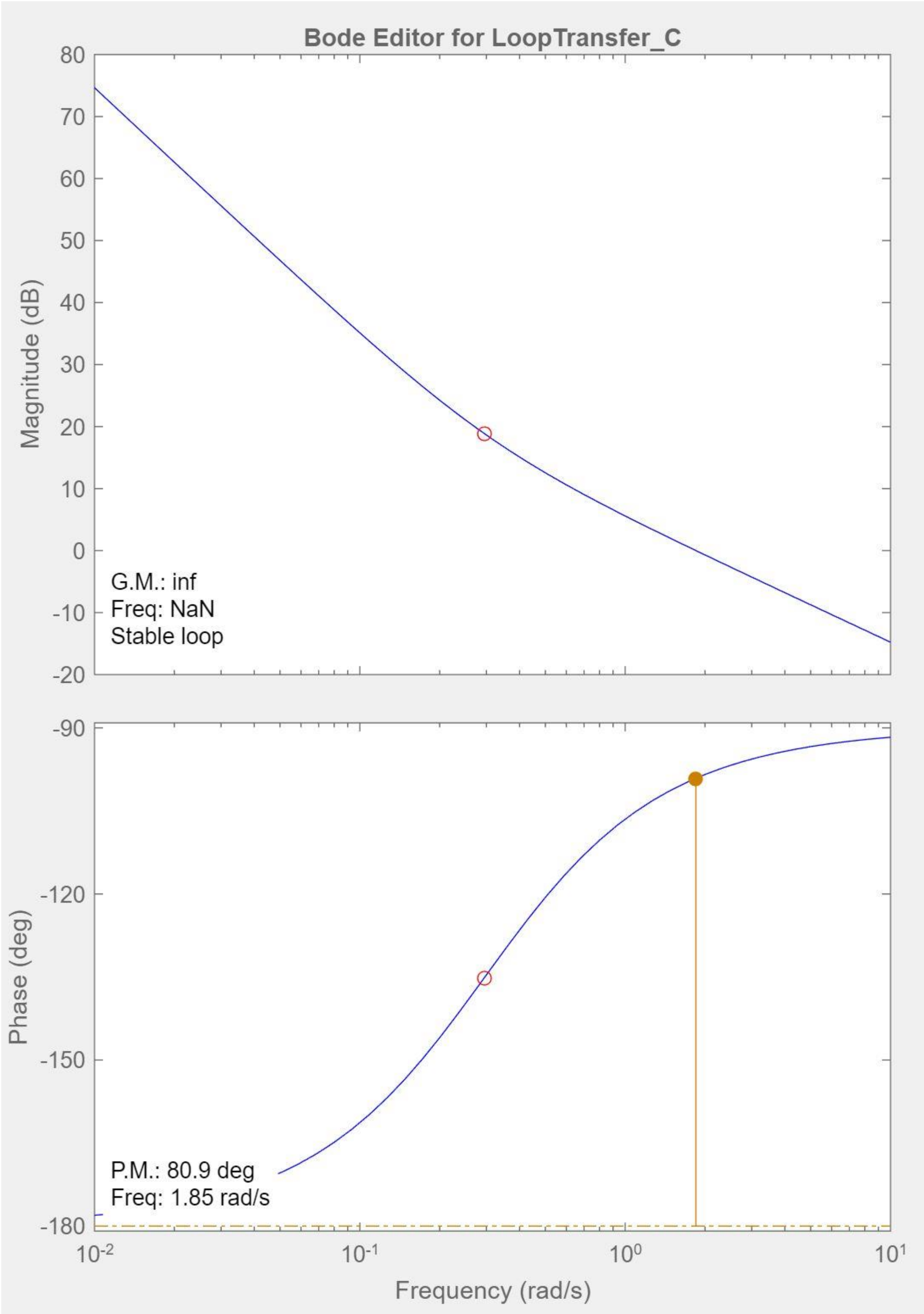

Figure 8. Bode plot of system

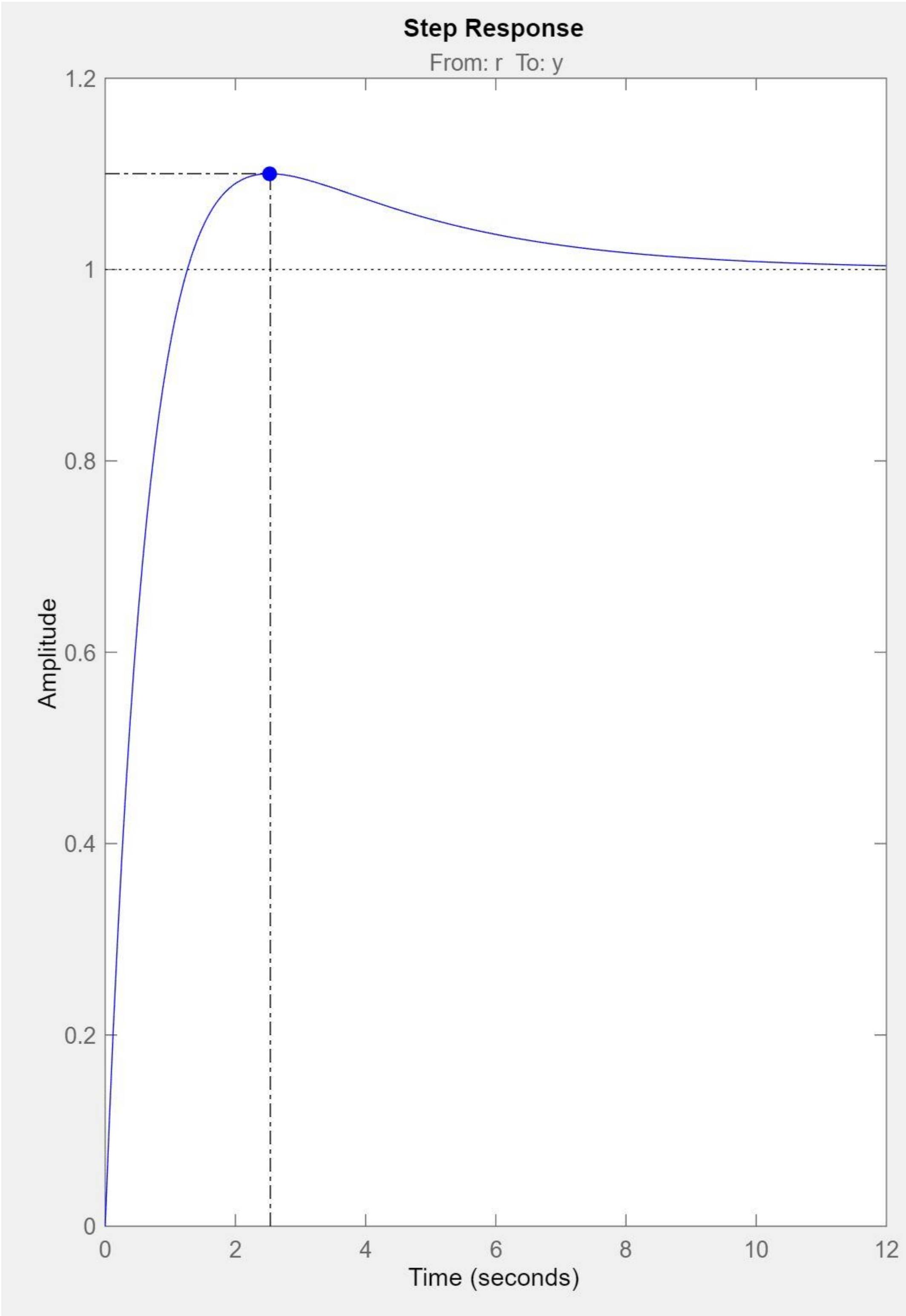

Figure 9. Step response of system without time delay

C = 0.91216 (s+0.296)

Poles at: -0.372, -1.45

```
>> damp(IOTransfer_r2y)

   Pole        Damping       Frequency      Time Constant
                            (rad/seconds)     (seconds)

 -3.72e-01     1.00e+00      3.72e-01         2.69e+00
 -1.45e+00     1.00e+00      1.45e+00         6.88e-01
```

Figure 10. Results from the control system designer

Damping ratio: 1

Gain crossover frequency: 1.85 rad/s

Phase margin = 80.9 deg = 1.412 rad

Time delay: 1.412/1.85 = 0.76s

Maximum allowable time delay: **0.38s**

Noe Lepez Da Silva Duarte

4)

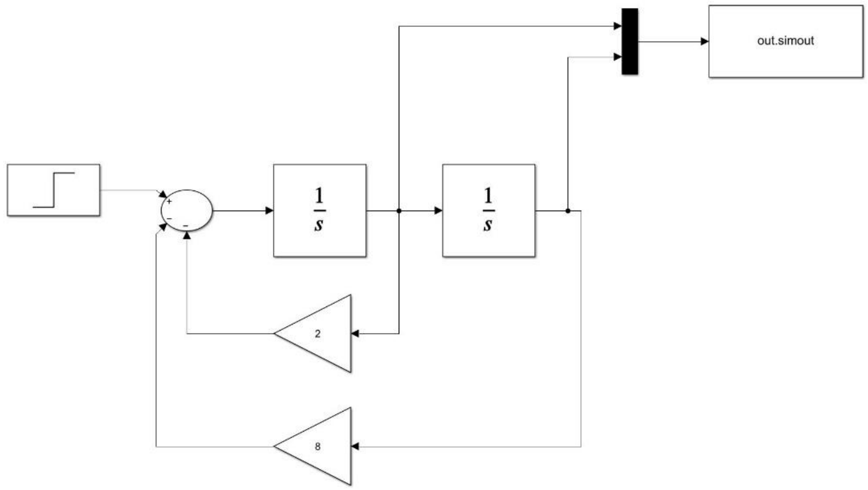

Figure 11. Simulink model

i)

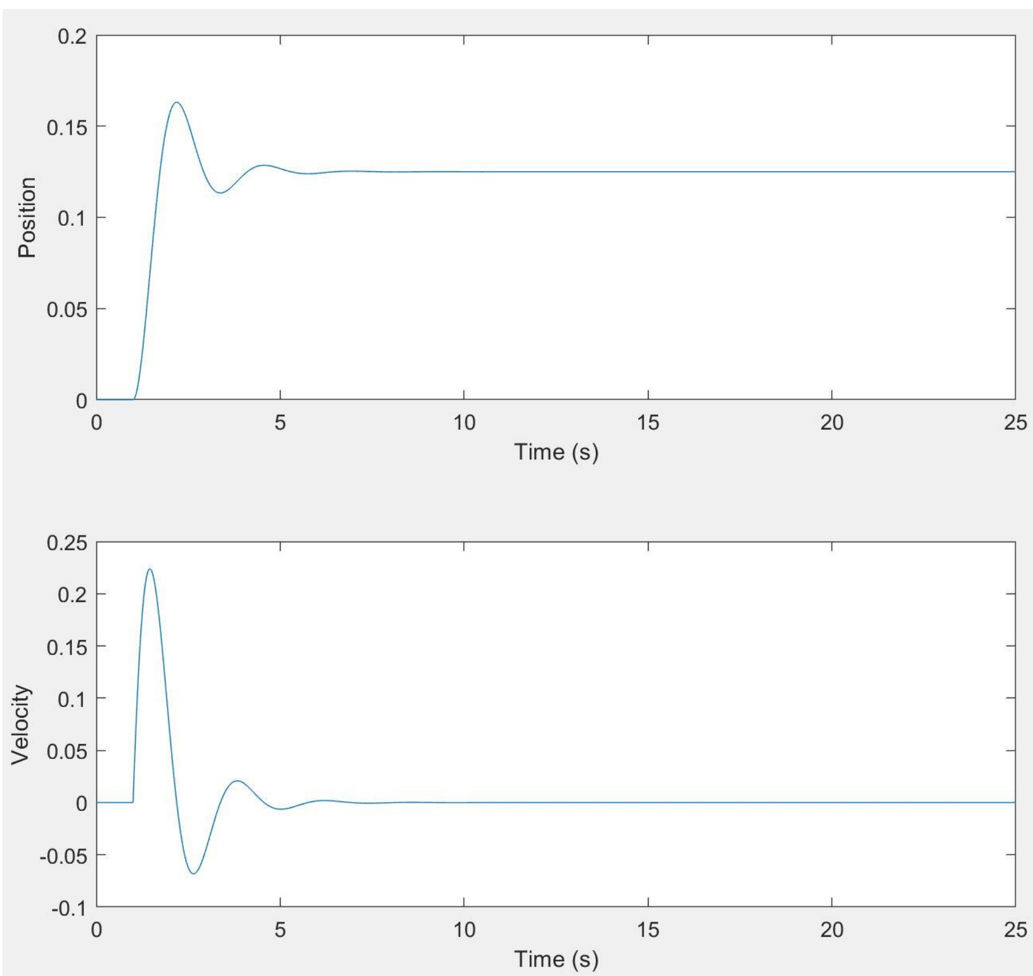

Figure 12. System response to a unit step input at 1s

Noe Lepez Da Silva Duarte

ii)

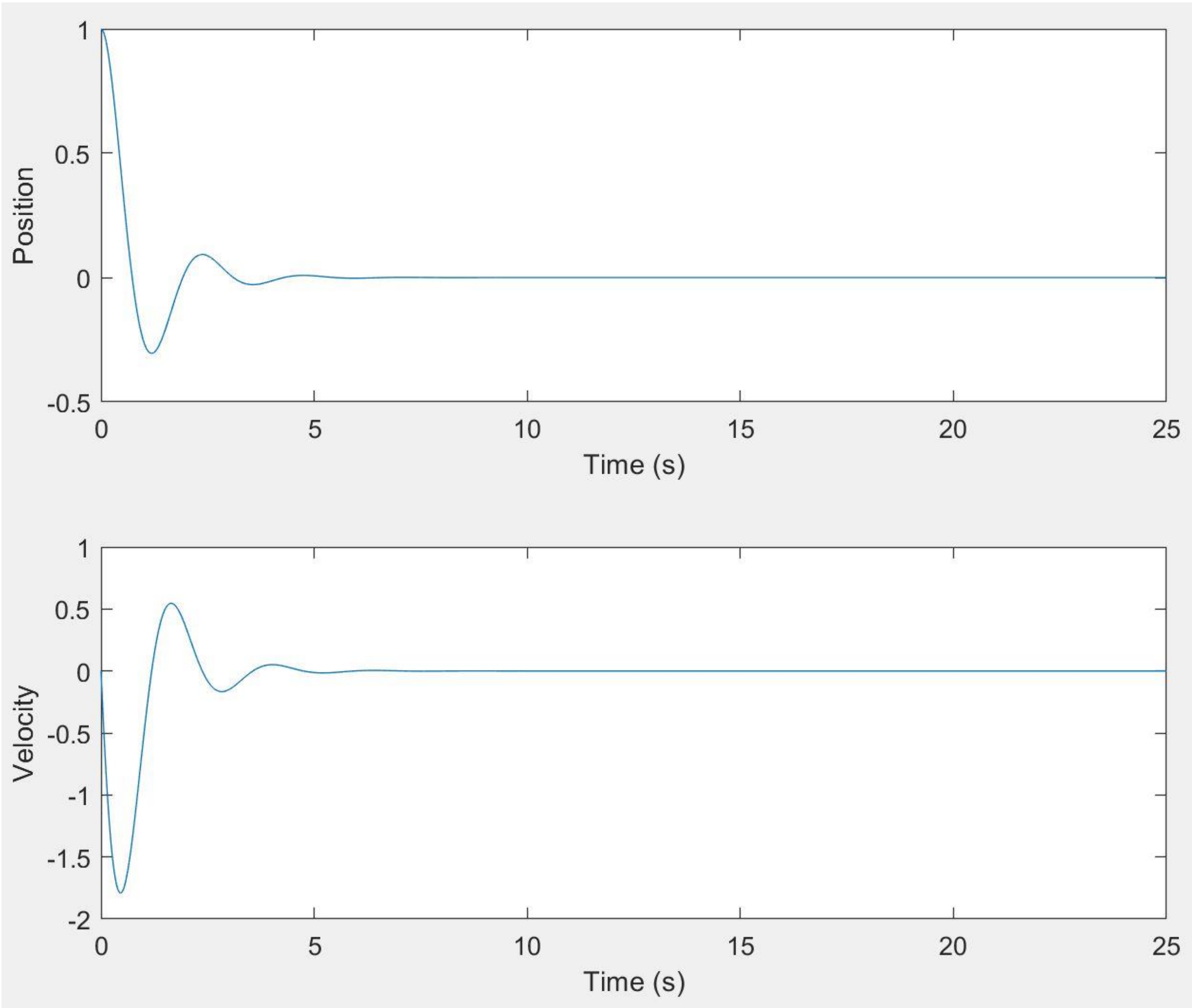

Figure 13. System response with an initial position of 1

iii)

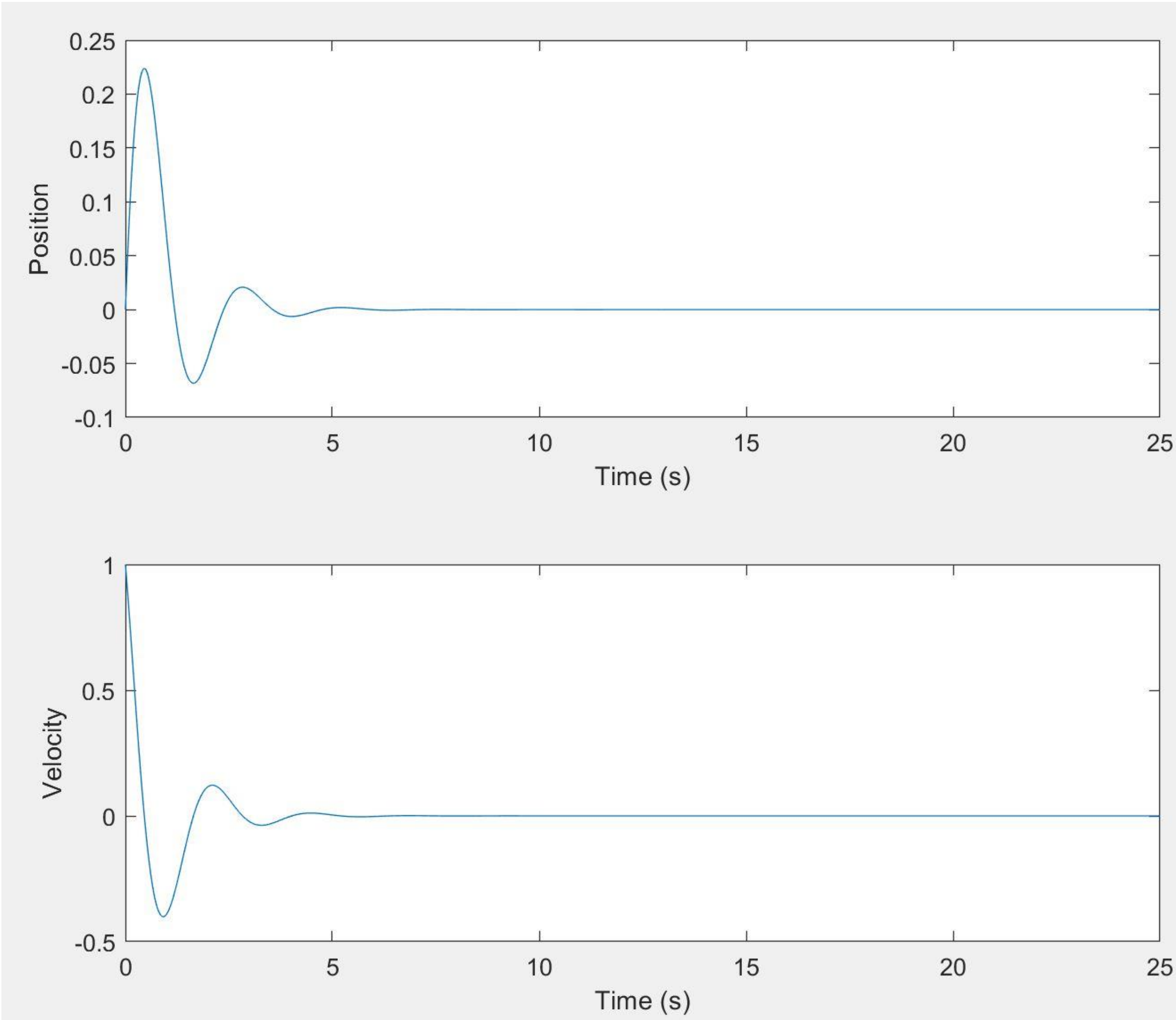

Figure 14. System response with an initial velocity of 1

iv)

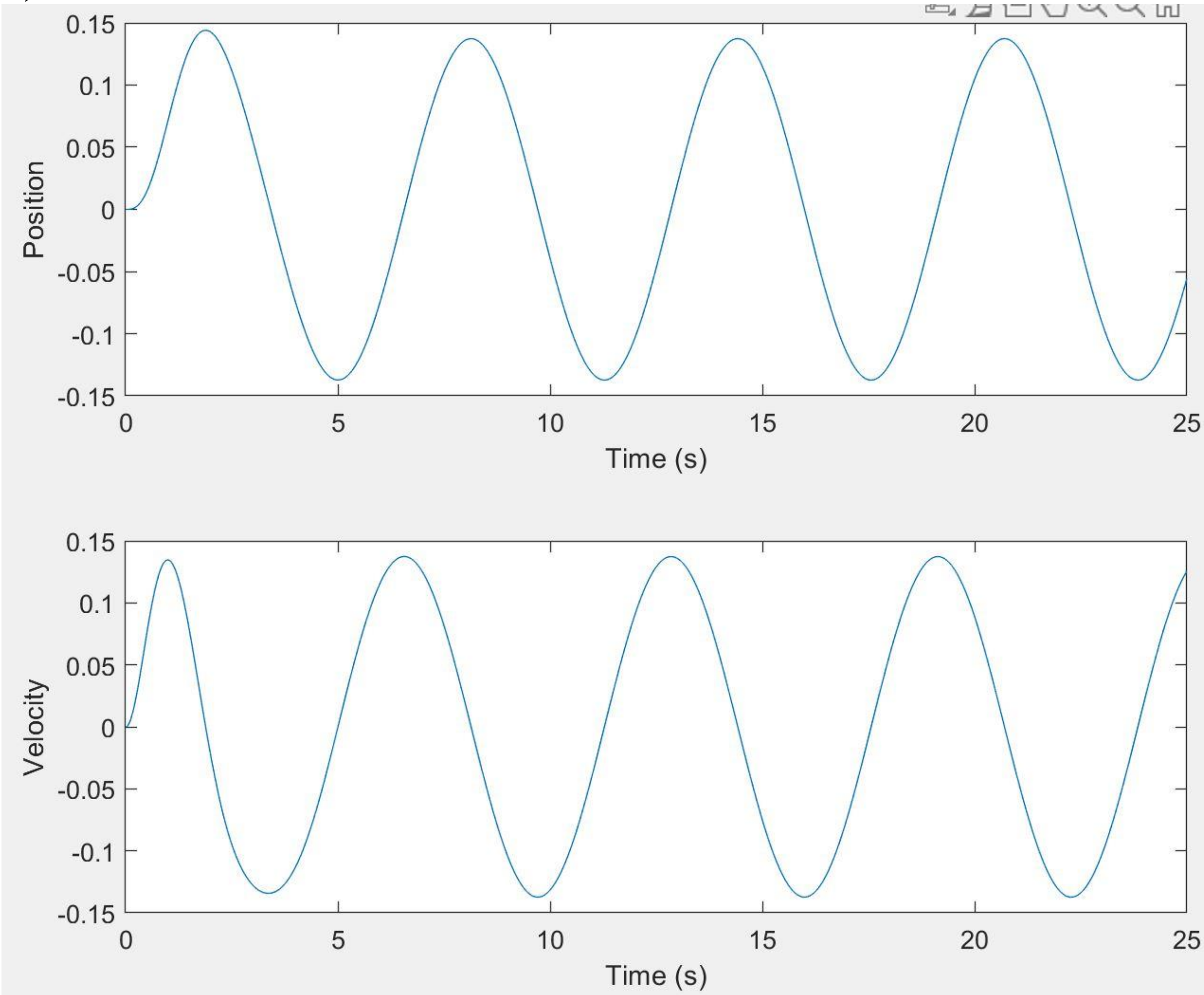

Figure 15. System response with sinusoidal input

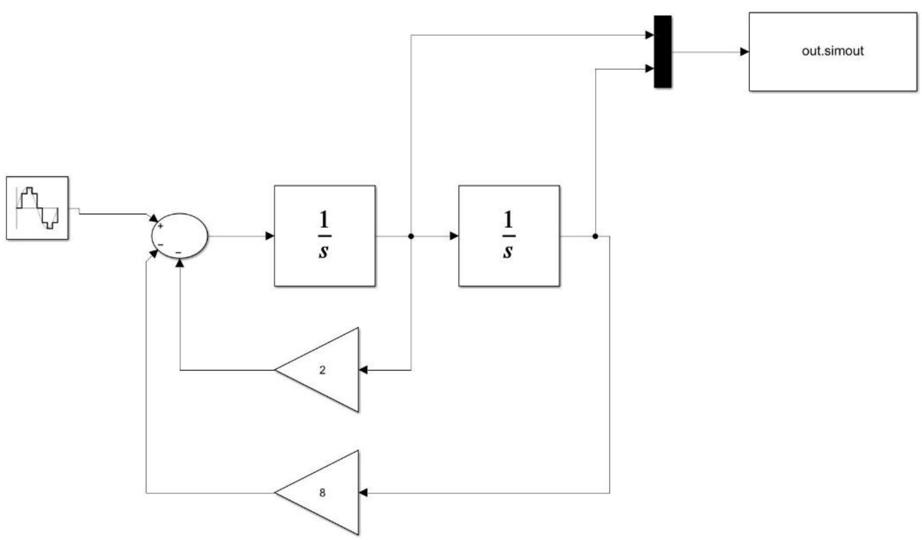

Figure 16. Simulink diagram for question 4 part 4

Noe Lepez Da Silva Duarte

5)

```
a = sim("Q5");
response = a.get('x');
time = a.get('tout');
plot(time, response(:,1))
xlabel("Time (s)")
ylabel("Position")
```

Figure 17. Code used for question 5

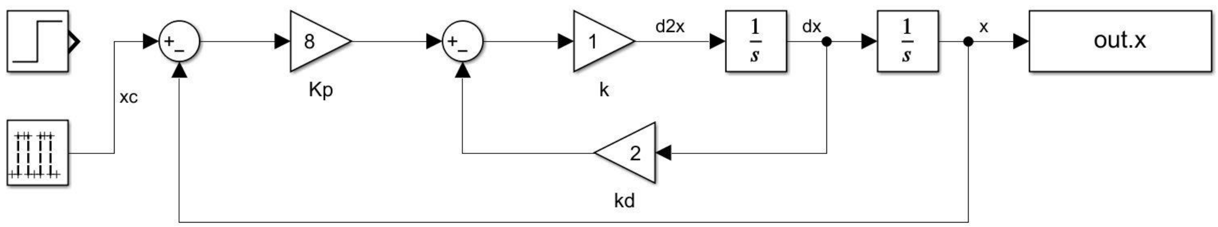

Figure 18. Simulink diagram used for question 5, both step and pulse input are present to allow them to be checked easily if needed

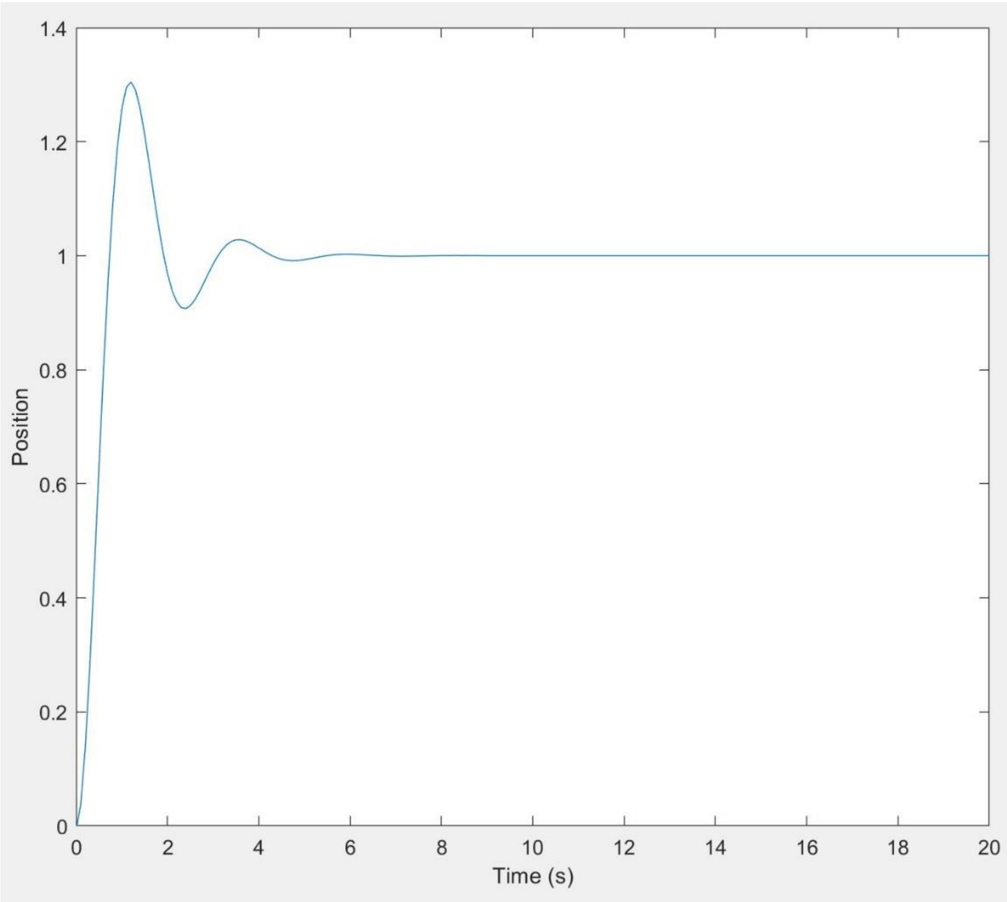

Figure 19. Position and velocity response of the system given in question 5 from a step input

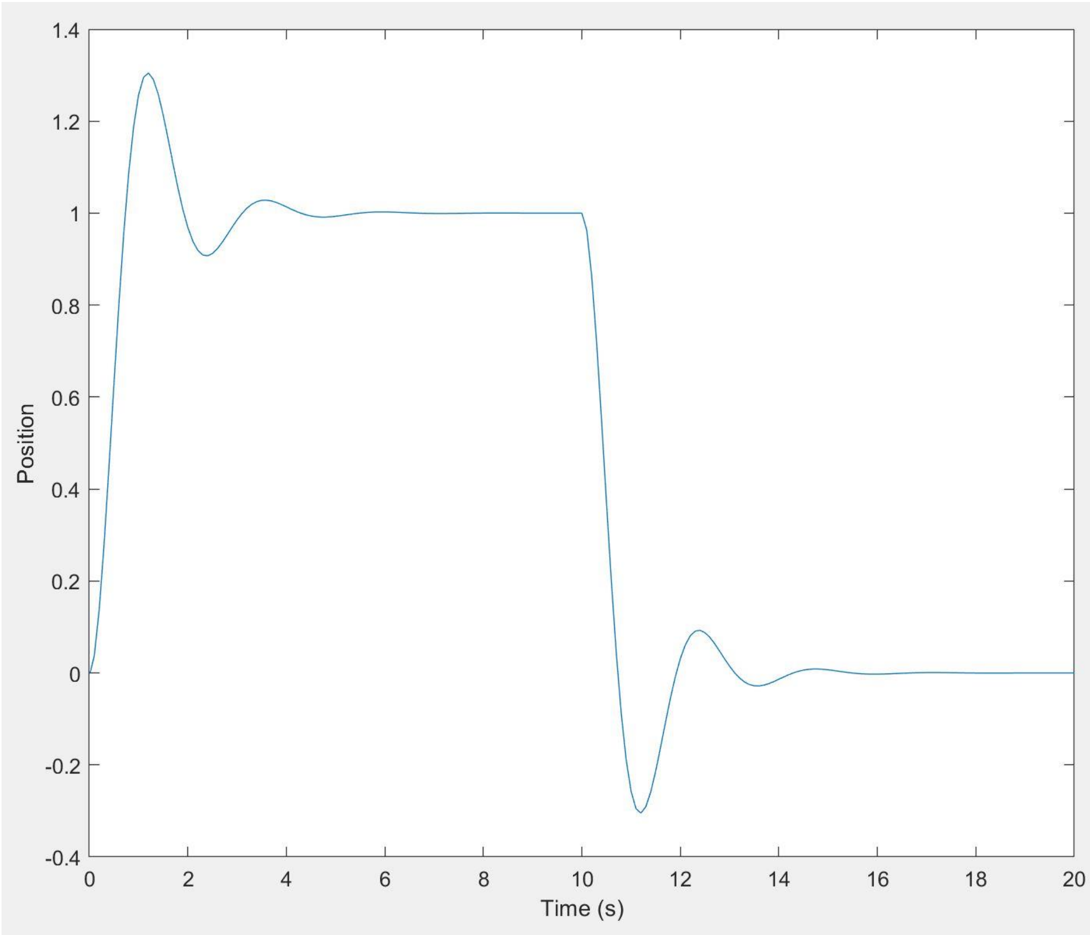

Figure 20. Position and velocity response of the system given in question 5 from a pulse input

6)

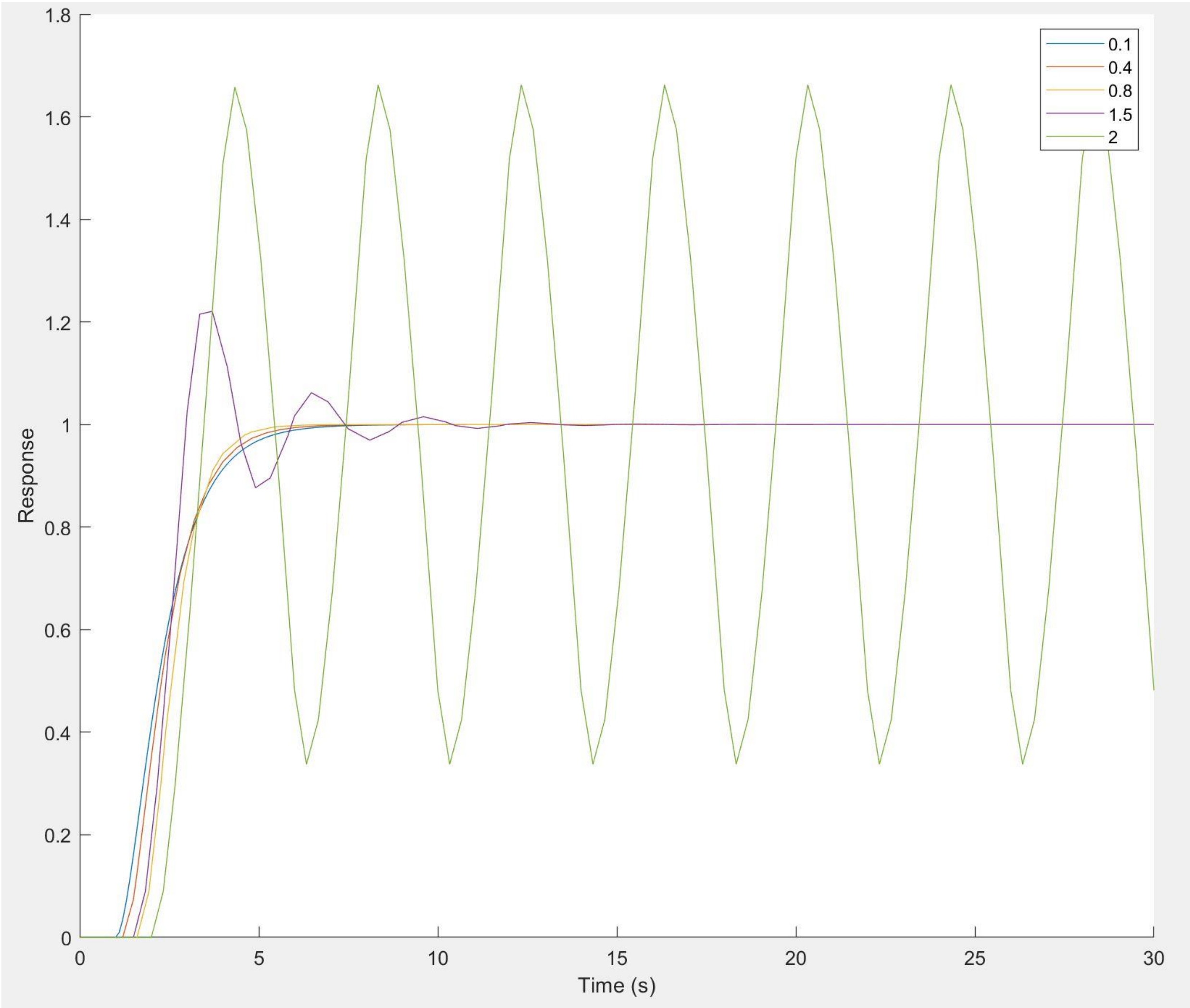

Figure 21. Response to given block diagram for different sample times

We can clearly see from the figure above that, as the sample time of the simulation increases, the response becomes less and less accurate, and starts to diverge highly from the expected response once the sample time is above 1. We can also see that the curves become more jagged with higher sample times, due to the low number of samples.

Noe Lepez Da Silva Duarte

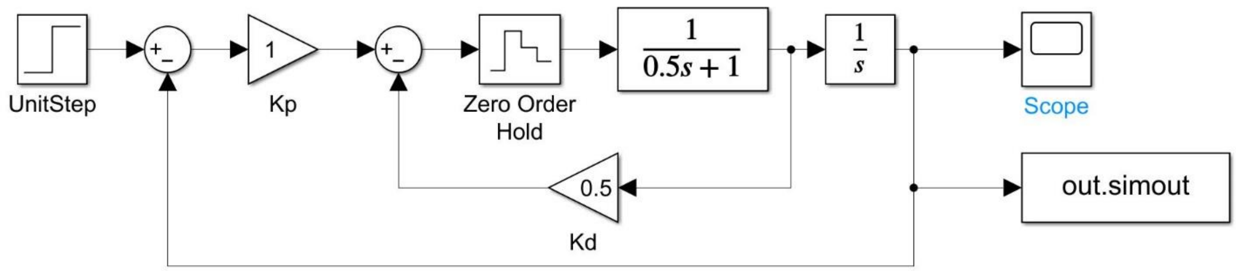

Figure 22. Simulink diagram of given block diagram

```
for i = ["0.1", "0.4", "0.8", "1.5", "2"]
    hold on
    set_param('Q6/UnitStep','SampleTime', i)
    a = sim("Q6");
    response = a.get('simout');
    time = a.get('tout');
    plot(time, response(:,1))
end
legend(["0.1", "0.4", "0.8", "1.5", "2"])
xlabel("Time (s)")
ylabel("Response")
```

Figure 23. Code for question 6

7)

```
A = [-0.02 0.1 0 -32.2 ; -0.040 -0.8 180 0 ; 0 -0.003 -1.0 0 ; 0 0 1 0];
B = [ 0 ; -8 ; -4 ; 0];
C = [0 -1 0 180;
    0 0 0 1];
D = [0;0];

a = sim("Q7");
set_param('Q7/State-Space','A','A','B','B','C','C','D','D')
Vc = a.get('Vc');
time = a.get('tout');
subplot(2,1,1)
plot(time, Vc)
xlabel("Time (s)")
ylabel("Vc (ft/s)")
hold on
subplot(2,1,2)
ths = a.get('ths');
plot(time, [ths(:,1), ths(:,2)])
xlabel("Time (s)")
ylabel("Pitch attitude (rad)")
legend('Response', 'Input')
```

Figure 24. Code for question 7

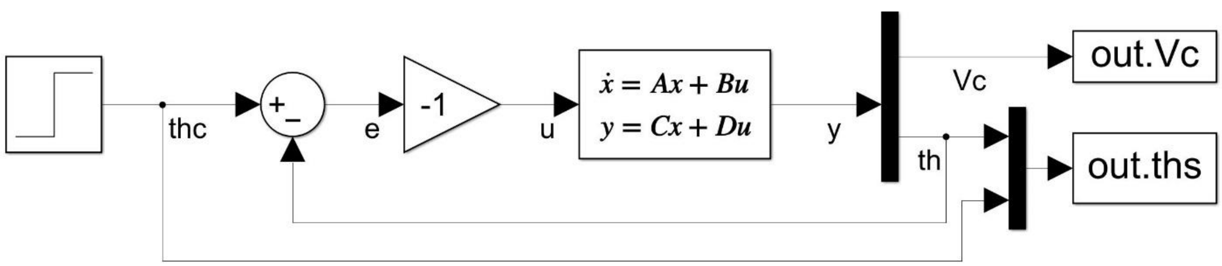

Figure 25. Simulink diagram for question 7

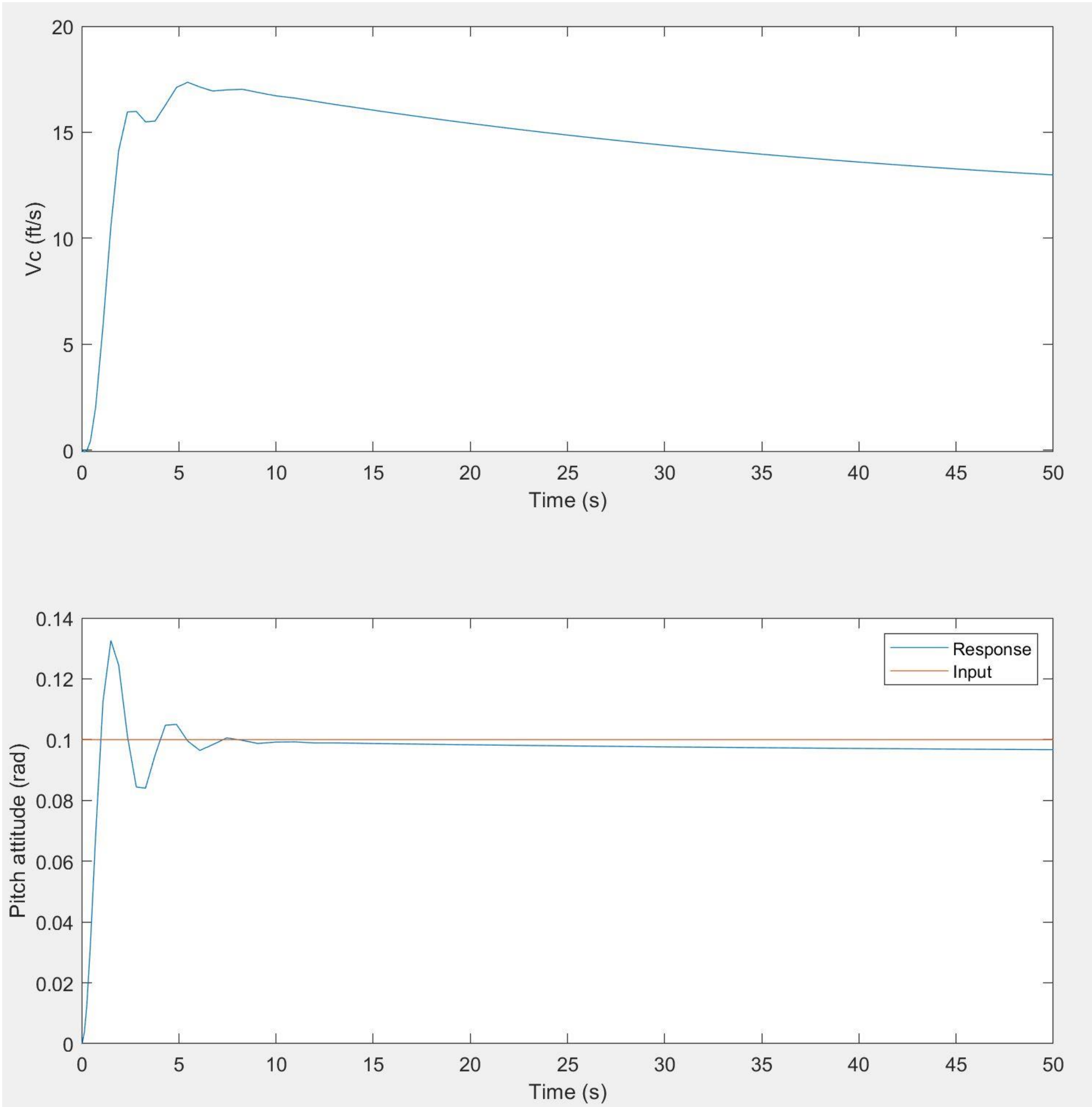

Figure 26. Input and response pitch attitude and climb rate as a function of time.

Noe Lepez Da Silva Duarte

8)

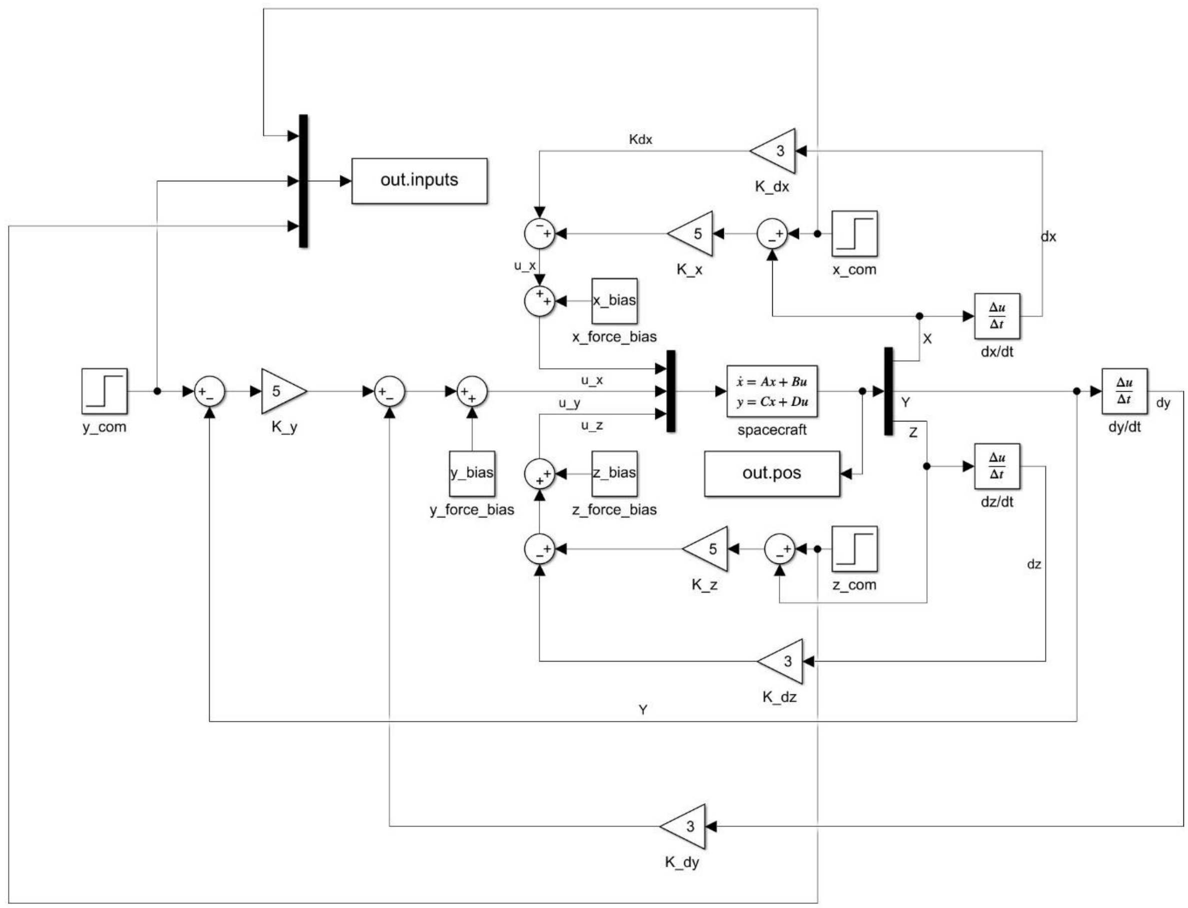

Figure 27. Simulink diagram for question 8

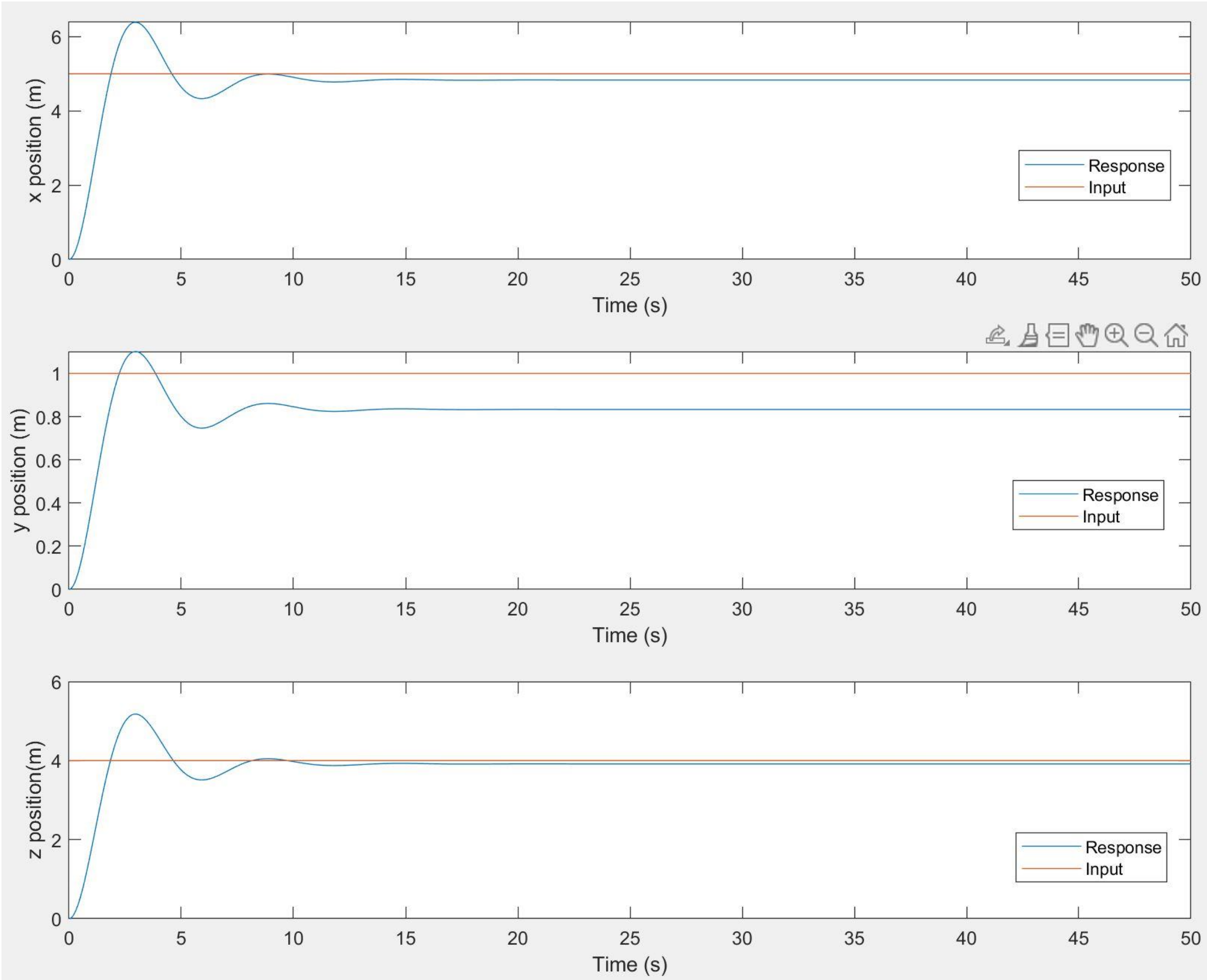

Figure 28. x, y, z position responses of spacecraft

Noe Lepez Da Silva Duarte

```
A = [0 0 0 1 0 0;
     0 0 0 0 1 0;
     0 0 0 0 0 1;
     0 0 0 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 0 0];
B = [0 0 0;
     0 0 0;
     0 0 0;
     inp 0 0;
     0 inp 0;
     0 0 inp];

C = [1 0 0 0 0 0;
     0 1 0 0 0 0;
     0 0 1 0 0 0];

D = [0 0 0;
     0 0 0;
     0 0 0];

x_bias = (-200*5000)/(sqrt(200^2 + 200^2 + 100^2)*m);
y_bias = (-200*5000)/(sqrt(200^2 + 200^2 + 100^2)*m);
z_bias = (-100*5000)/(sqrt(200^2 + 200^2 + 100^2)*m);
```

Where inp = 1000/m