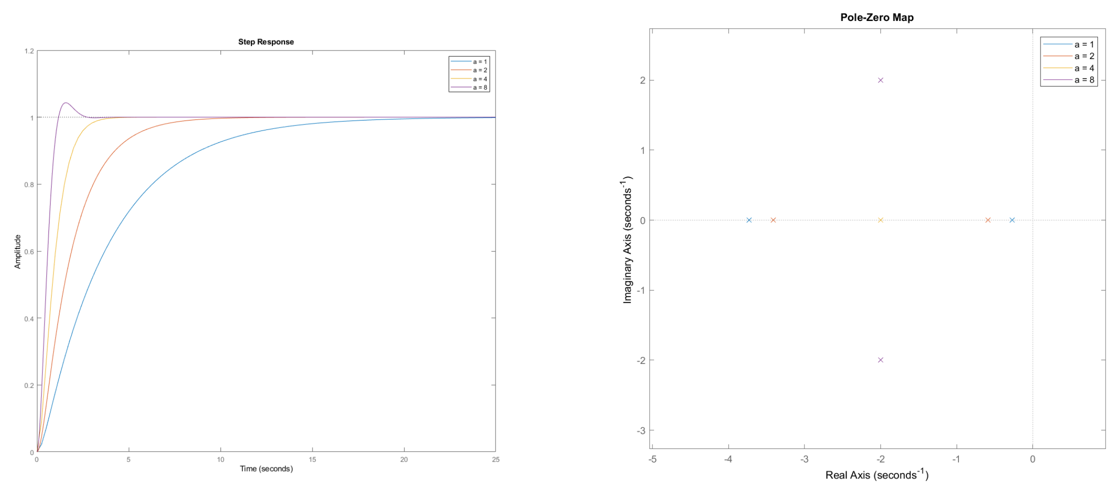
ECEN315 : Modelling in MATLAB
Assignment 2 : Submission

Daniel Eisen : 300447549

May 15, 2020

Question 1



(a) Step Response for varying a values

(b) Corresponding Pole/Zero Map

Figure 1: $\frac{O}{I} = \frac{a}{s^2+4s+a}$

For A = 1:
T1 = 3.732051 T2 = 0.267949
T = 1.000000
Ts = 14.878894

For A = 2:
T1 = 1.707107 T2 = 0.292893
T = 0.707107
Ts = 6.999617

For A = 4:
T1 = 0.500000 T2 = 0.500000
T = 0.500000
Ts = 2.916978

For A = 8:
T1 = 0.250000 T2 = 0.250000
T = 0.250000
Ts = 2.108152

MATLAB output

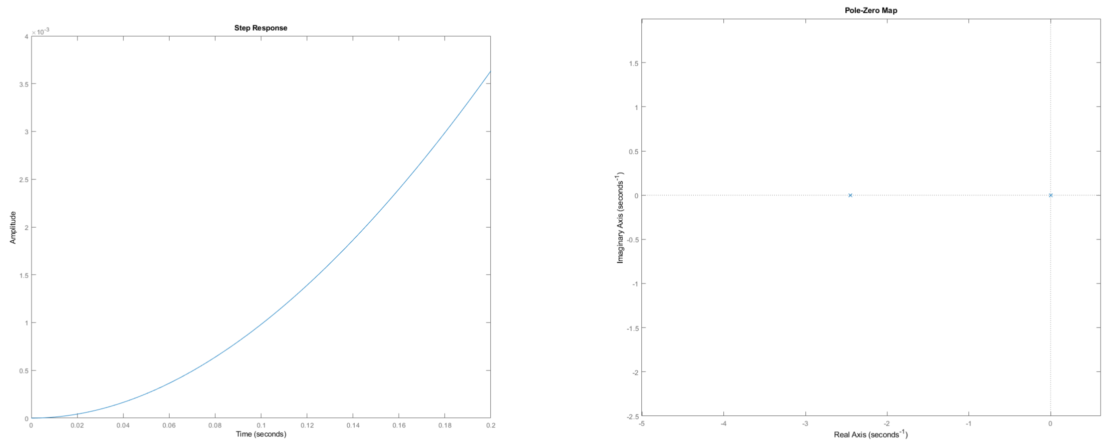
Time constants were calculated via the values given by the damp function (ω_N, ζ).

$$\tau_n = \frac{1}{\omega_N \cdot \zeta}$$

$$\tau = \sqrt{\tau_1 \cdot \tau_2}$$

Question 2

a)

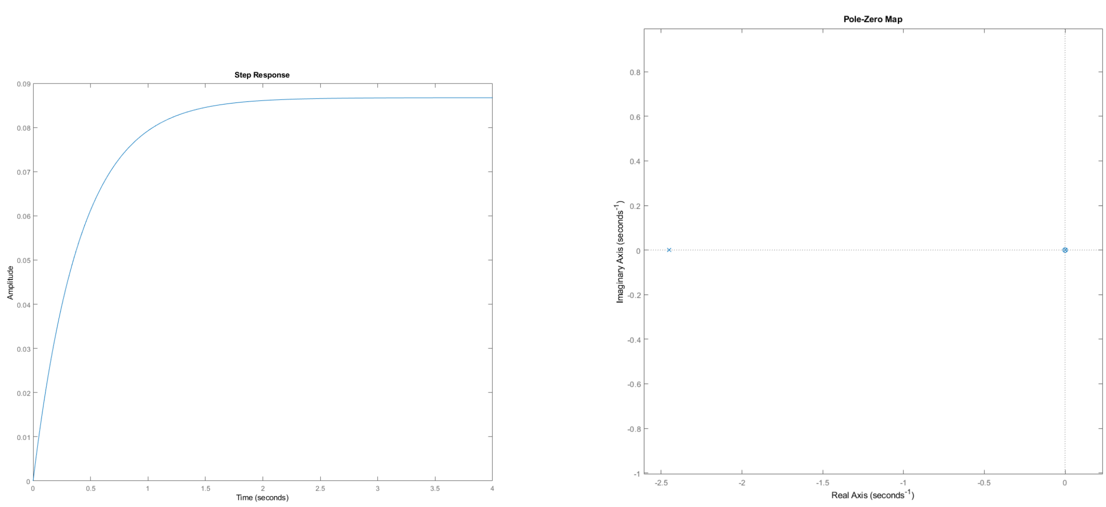


(a) (b)

Figure 2:

Figure 2 shows that step response increases without bounds. This is however to be expected as its the angular displacement of the motor's shaft. Its initial curve shows change but is evened out as it meets max angular velocity.

b)

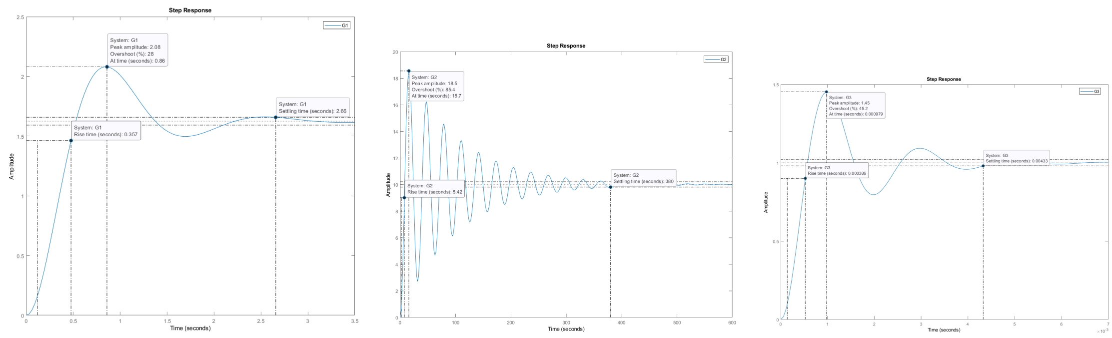


(a) (b)

Figure 3:

Figure 3 shows the step response of the angular velocity, which is seen to reach a max value (steady state).

Question 3



(a) (b) (c)

Figure 4:

```
G_1(s): Z=0.375000, Wn=4.000000,    Ts=2.657041,   Tp=0.859632,  Tr=0.356904, OS=28.025548

G_2(s): Z=0.050000, Wn=0.200000,    Ts=380.016288, Tp=15.707963, Tr=5.416048, OS=85.446128

G_3(s): Z=0.244567, Wn=3271.085447, Ts=0.004325,   Tp=0.000979,  Tr=0.000386, OS=45.241508
```

Figure 4 shows each of the of three systems:

$$G_1(s) = \frac{26}{s^2 + 3s + 16}$$
$$G_2(s) = \frac{0.4}{s^2 + 0.02 + 0.04}$$
$$G_3(s) = \frac{1.07{\times}10^7}{s^2 + 1.6{\times}10^3 s + 1.07{\times}10^7}$$

With the rise, settling and peak/overshoot data marked at their occurrence. The verbatim text shows the output of the MATLAB script's calculated values; (ζ, ω_n, τ_s, τ_p, τ_r, $\%OS$)

Question 4

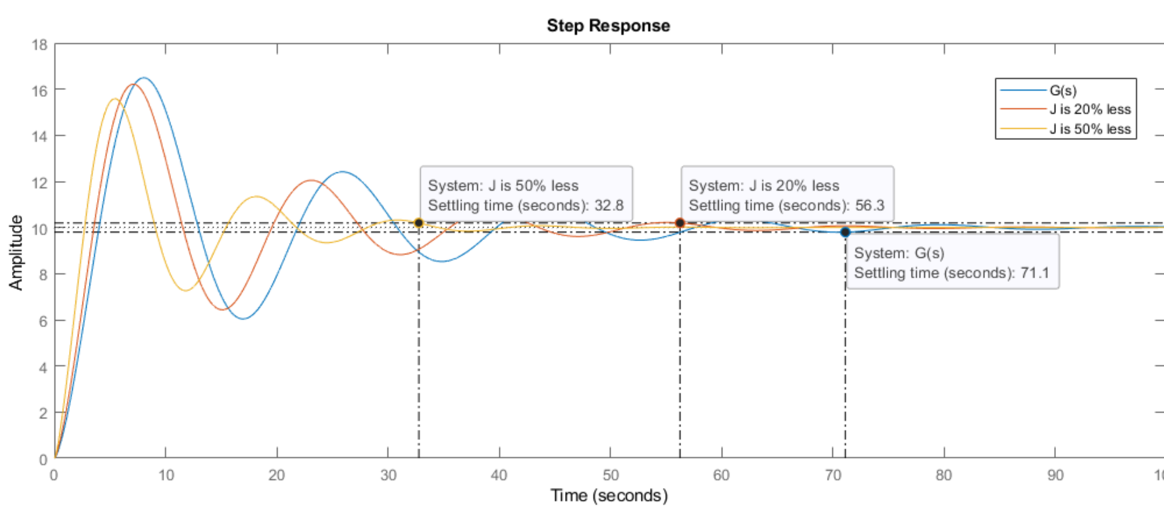


Figure 5: $G(s) = \frac{\Theta(s)}{\Theta_d(s)}$

Figure 5 plots the output of a 10° step input to G(s), compared with the reduction of J by 20% and 50% and the effect on the settling time displayed.

MATLAB computed the TF to be equal to, $\frac{\Theta(s)}{\Theta_d(s)} = \frac{1.08e09s+1.08e09}{1.08e09s^3+8.64e09s^2+1.08e09s+1.08e09}$

Question 5

a)

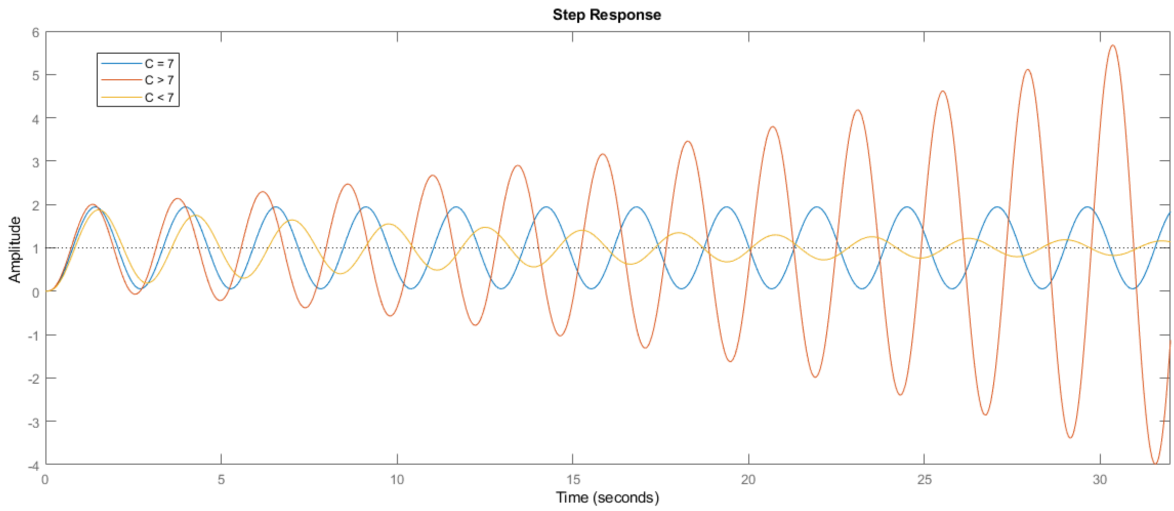


Figure 6:

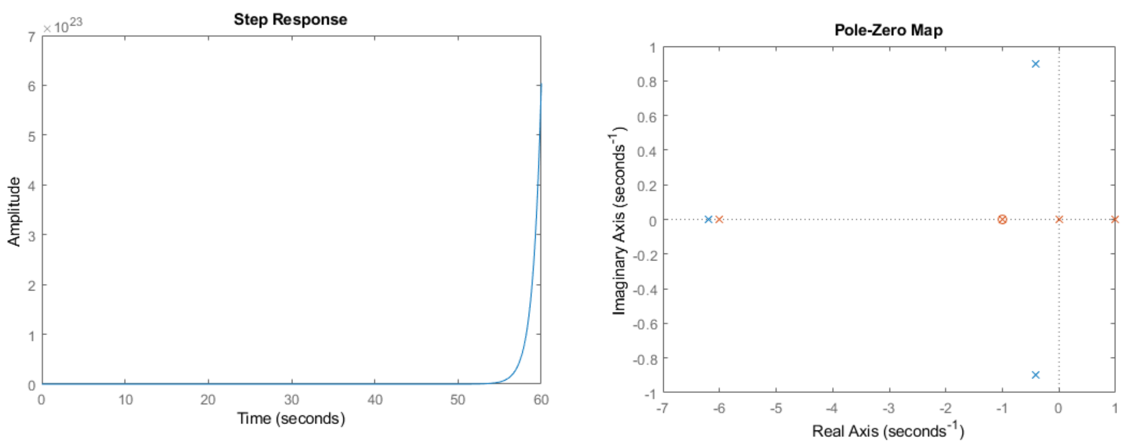
`MatLab Output: Minimum series gain > 7.000000`

Using a search loop iterating the value of C, the gain value that gives a marginally stable system was found to be 7.

$\therefore$ the system is unstable at $C > 7$

Figure 6 shows step response for the foundry case C=7 and either side.

b)



(a) (b)

Figure 7:

Using a lead-lag controller $C(\frac{s-z}{s-p}) = 0.01(\frac{s+1}{s-1})$ to place a pole at positive 1 with a lower steady state gain than (a): 7 compared to -0.01).

Appendices

Task 1

```
clear;
clc;

s = tf('s');

figure(1);
hold on
for a = [1 2 4 8]
    sys = a /(s^2 + 4*s + a);
    figure(1);
    step(sys,25);
    [Wn, Z] = damp(sys); %To get T1 and T2
    T1 = 1/Wn(1)*Z(1);
    T2 = 1/Wn(2)*Z(2);
    T = sqrt(T1*T2);
    fprintf("\nFor A = %d:\nT1 = %f T2 = %f\nT = %f\nTs = %f\n",...
        a, T1, T2, T,...
        stepinfo(sys).SettlingTime)

end
legend('a = 1', 'a = 2', 'a = 4', 'a = 8');
hold off;

figure(2);
hold on
for a = [1 2 4 8]
    sys = a /(s^2 + 4*s + a);
    [x,t] = step(sys,25);
    pzmap(sys)
end
legend('a = 1', 'a = 2', 'a = 4', 'a = 8');
hold off;
```

Task 2

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

%a
motor_pos = 0.0425/(s*(s+2.45));
opt = stepDataOptions('StepAmplitude', 5);
figure("name","Motor Position Step Reponse");
step(motor_pos,opt,0.2)
figure("name","Motor Position Poles");
pzmap(motor_pos)

%b
motor_vel = motor_pos * s; %differentiate
opt = stepDataOptions('StepAmplitude', 5);
figure("name","Motor Velocity Step Reponse");
step(motor_vel,opt)
figure("name","Motor Velocity Poles");
pzmap(motor_vel)
```

Task 3

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

G1 = 26 / (s^2 + 3*s + 16);
linearSystemAnalyzer(G1);
[Wn1,Z1] = damp(G1);
fprintf("G_1(s): Z=%f, Wn=%f, Ts=%f, Tp=%f, Tr=%f, OS=%f \n\n",...
    Z1(1), Wn1(1),...
    stepinfo(G1).SettlingTime,...
    stepinfo(G1).PeakTime,...
    stepinfo(G1).RiseTime,...
    stepinfo(G1).Overshoot)


G2 = 0.4 / (s^2 + 0.02*s + 0.04);
linearSystemAnalyzer(G2);
[Wn2,Z2] = damp(G2);
fprintf("G_2(s): Z=%f, Wn=%f, Ts=%f, Tp=%f, Tr=%f, OS=%f \n\n",...
    Z2(1), Wn2(1),...
    stepinfo(G2).SettlingTime,...
    stepinfo(G2).PeakTime,...
    stepinfo(G2).RiseTime,...
    stepinfo(G2).Overshoot)


G3 = 1.07E7 / (s^2 + 1.6E3*s + 1.07E7);
linearSystemAnalyzer(G3);
[Wn3,Z3] = damp(G3);
fprintf("G_3(s): Z=%f, Wn=%f, Ts=%f, Tp=%f, Tr=%f, OS=%f \n",...
    Z3(1), Wn3(1),...
     stepinfo(G3).SettlingTime,...
     stepinfo(G3).PeakTime,...
     stepinfo(G3).RiseTime,...
    stepinfo(G3).Overshoot)
```

Task 4

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

% Constants
a=1;
b=8;
k=10.8E8;
J=10.8E8;

%a
controller = k*(s+a)/(s+b);
spacecraft = 1/(J*s^2);
G = feedback(controller*spacecraft,1) %closed loop TF


%Percent reduction
spacecraft1 = 1/((J*0.8)*s^2);
spacecraft2 = 1/((J*0.5)*s^2);

G1 = feedback(controller*spacecraft1,1);
G2 = feedback(controller*spacecraft2,1);

% degree input
G=G*10;
G1=G1*10;
G2=G2*10;

%plotting b and c
linearSystemAnalyzer(G,G1,G2)
```

Task 5

```
%% a
clear;
clc;
s = tf('s');

G = 6 / (s*(s+6)*(s+1));

for c = (5:0.01:10)
    if stepinfo(feedback(c*G,1)).PeakTime >= Inf
        fprintf("Minimum series gain > %f\n", c)
        linearSystemAnalyzer(feedback(c*G,1), feedback((c+1)*G,1),feedback((c-1)*G,1))
        break;
    end
end

%% b
clear;
clc;
s = tf('s');

G = 6 / (s*(s+6)*(s+1));
%Lead Lag: C((s-z)/(s-p))
LL = 0.01*((s+1)/(s-1));
linearSystemAnalyzer(feedback(LL*G,1))
figure(1)
hold on
pzmap(feedback(G,1))
pzmap(feedback(LL*G,1))
hold off
```