

ECEN315 - Assignment 2

Joshua Benfell - 300433229

May 12, 2020

1 Question 1

```
1 clear
2 clf
3 figure(1)
4 subplot(1,2,1)
5 hold on
6 aVals = [1 2 4 8];
7 Legend = cell(length(aVals), 1);
8 StepInfo = cell(length(aVals), 1);
9 TimeConstants = cell(length(aVals), 1);
10 i = 1;
11
12 for a = aVals
13     sys = tf(a, [1 4 a]);
14     step(sys)
15     StepInfo{i} = stepinfo(sys);
16     Legend{i} = strcat(num2str(a), 'V Step');
17     [Wn, Z] = damp(sys);
18     TimeConstants{i} = 1./(Wn .* Z);
19     i = i + 1;
20 end
21 hold off
22 legend(Legend)
23
24 subplot(1,2,2)
25 hold on
26 for a = aVals
27     sys = tf(a, [1 4 a]);
```

```

28     pzmap( sys )
29 end
30 legend( Legend )

```

Table 1: Time Constants and Settling Time for Various Step Inputs

Step Size	Time Constants	Settling Time
1	$\begin{pmatrix} 3.7321 \\ 0.2679 \end{pmatrix}$	14.879
2	$\begin{pmatrix} 1.7071 \\ 0.2929 \end{pmatrix}$	6.9996
4	$\begin{pmatrix} 5 \\ 5 \end{pmatrix}$	2.1970
8	$\begin{pmatrix} 5 \\ 5 \end{pmatrix}$	2.1082

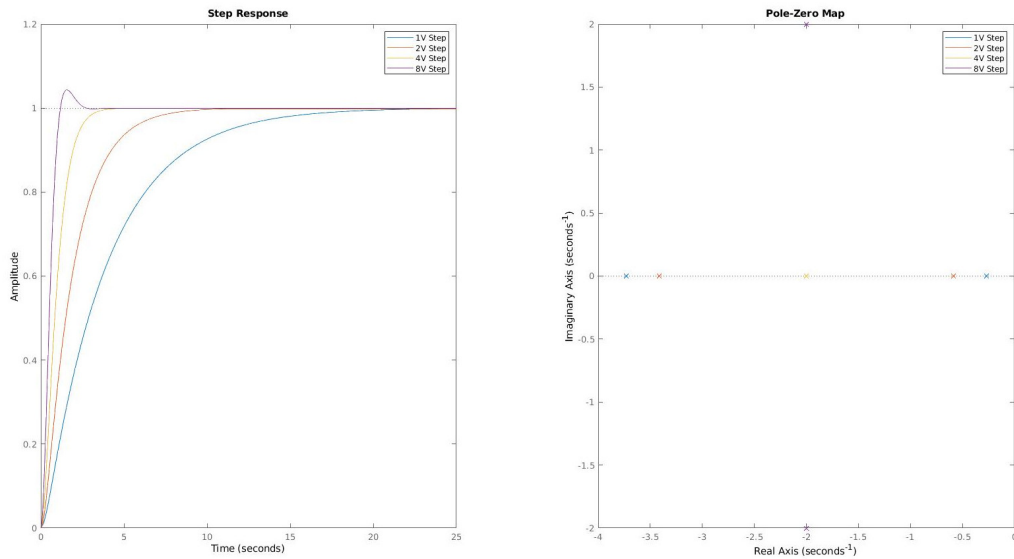


Figure 1: Step Response to various step sizes and PZ Map of those step responses

By changing the value of a , the transfer function goes from over to underdamped. a values of 1 and 2 make the system over damped, 4 is critically damped and 8 makes it under damped, reflected by the overshoot. We can confirm the nature of the step responses by looking at the poles. When a is 1 or 2, the poles are real and negative but not the same, so we know that

it's over damped. When a is 4, the poles are negative and equal, so it's critically damped. Finally when a is 8, the poles are complex which introduces oscillatory factors to the response, and therefore the system is underdamped.

From the settling times we can see that the higher a is the faster the response settles. This makes sense as critically damped is the fastest that it can settle without oscillation, whereas over damped systems go slower and under damped systems are faster at rising, but if there's too much oscillation it will take a while to settle. Because there's small oscillation, it makes sense to settle faster than the critically damped system.

There appears to be two time constants for the over damped systems and 1 for the critically damped and under damped systems. I'm not entirely sure what this means but, the two time constants seems to indicate that there are multiple exponential components that are competing as they are different values. When the values are the same then the exponentials are similar.

2 Question 2

```

1 clear
2 clf
3 %a
4 sys = tf(0.0425, [1 2.45 0]); % Angular Displacement
5 step(5 * sys, 10);
6 %Graph is pretty much a ramp
7 %This makes sense as the motor shaft will continue to
  spin while powered.
8 %If we were to plot between -180 and 180 we would see
  this loop around like
9 %a sawtooth wave
10 hold on
11 %b
12 %Angular Velocity, so have to differentiate the TF i.e.
  multiply by s
13 sys2 = tf(0.0425, [1 2.45]);
14 step(5 * sys2);
15 % steady state is 0.0867 rads/sec
16 legend("Angular Displacement TF", "Angular Velocity TF
  ")

```

The blue line in fig. 2 is the response for the transfer function in a. This transfer function just continues with a positive gradient after a step response

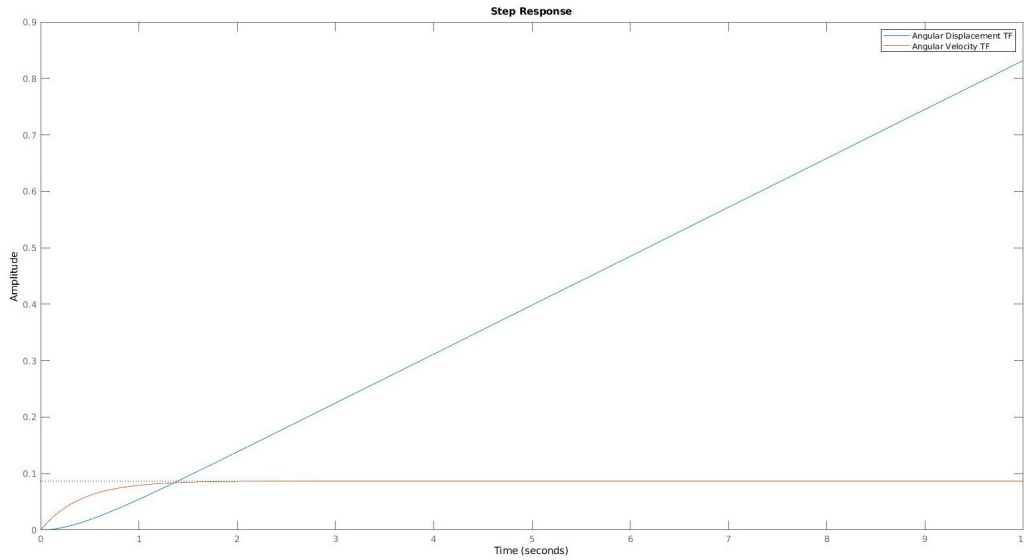


Figure 2: Step Response of $\frac{\Theta_L(s)}{E_a(s)}$ and $\frac{\Omega_L(s)}{E_a(s)}$

is added. This makes sense as it is a motor, so unless you wrap around once you hit 2π radians, the displacement will technically always increase.

This is reflected in part b. When we want to get the angular velocity over the applied voltage, we differentiate the function. To do this in the frequency domain, we multiply by s . This results in the following transfer function:

$$\frac{\Omega_L(s)}{E_a(s)} = \frac{0.0425}{s + 2.45} \quad (1)$$

This is a first order transfer function that has a steady state of $0.01735 * E_a$. For a 5V step, this is a steady state gain of 0.0867 rads/sec which is how much the motor shaft spins every second.

3 Question 3

```

1 sys1 = tf(26, [1 3 16]);
2 sys2 = tf(0.4, [1 0.02 0.04]);
3 sys3 = tf(1.07e7, [1 1.6e3 1.07e7]);
4 linearSystemAnalyzer(sys1, sys2, sys3);
5
6 [wn1, zeta1] = damp(sys1)
7 [wn2, zeta2] = damp(sys2)
8 [wn3, zeta3] = damp(sys3)

```

Table 2: Relevant values for the systems defined in question 3

System	ω_n	ζ	T_s (s)	T_p (s)	T_R (s)	$OS\%$
$G_1(s)$	4	0.375	2.66	0.86	0.357	28
$G_2(s)$	0.2	0.05	3.8	15.7	5.42	85.4
$G_3(s)$	3.2711×10^3	0.2446	0.00433	0.000979	0.0003864	45.2

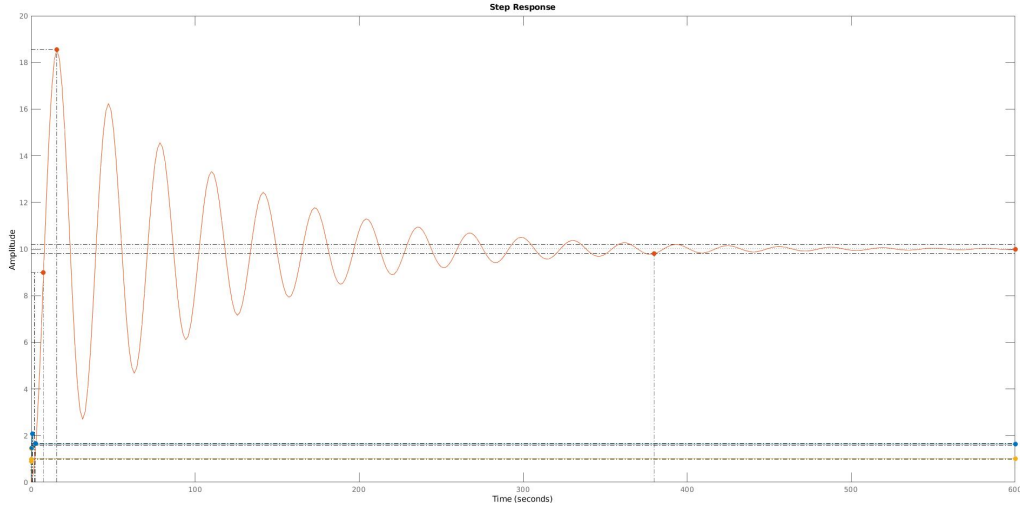


Figure 3: Usage of the LTI Viewer exported to figure

In fig. 3 the systems are assigned as follows, G_1 is blue, G_2 is Orange, G_3 is yellow (light brown). All systems are under damped but, G_1 and G_3 are the least under damped, the overshoot the target a little bit and settle quickly. G_2 is severely under damped, it oscillates quite a lot before settling.

4 Question 4

```

1 clear
2 clf
3
4 a = 1;
5 b = 8;
6 k = 10.8e8;
7 J = 10.8e8;
8
9 %a
10 controller = tf([k k*a], [1 b]);
11 spacecraft = tf(1, [J 0 0]);

```

```

12 sys = feedback(controller * spacecraft , 1);
13
14 percentages = [1 0.8 0.5];
15 i = 1;
16 figure(1)
17 hold on
18 Legend = cell(length(percentages),1);
19 for p = percentages
20     spacecraft = tf(1, [J*p 0 0]);
21     sys = feedback(controller * spacecraft , 1);
22     step(10 * sys)
23     Legend{i} = strcat(num2str(p), ' of J');
24     i = i + 1;
25 end
26 legend(Legend);
27 ylabel("Actual Altitude \theta(t)")

```

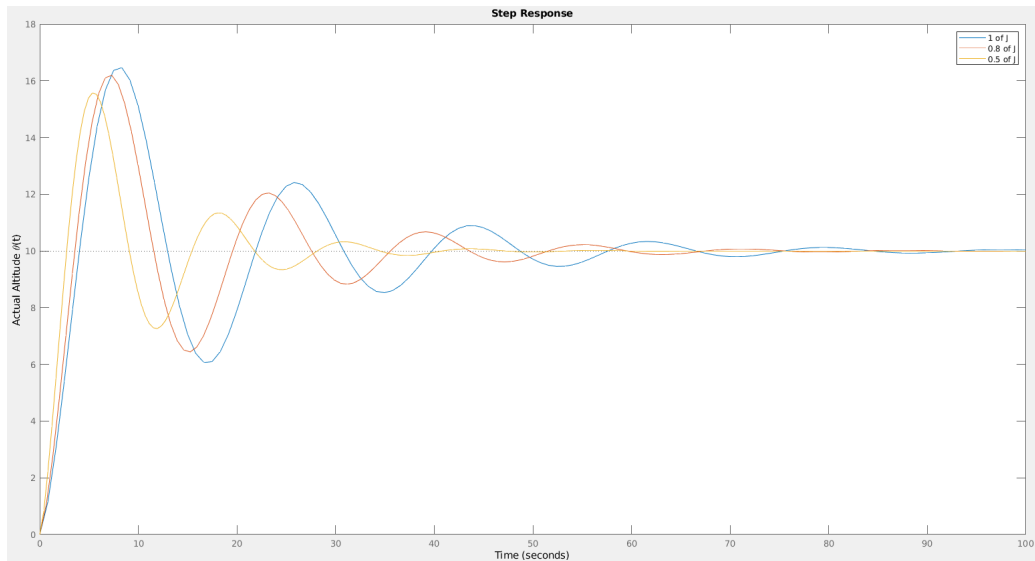


Figure 4: Satellite Step Responses for various moments of inertia

As seen in fig. 4 the decrease in J causes a faster transient response and smaller overshoot which means faster settling. This is because the Satellite doesn't have enough damping. Any push in a direction may overshoot the target and requires a push in the other direction to keep heading towards it. When the Satellite has a higher moment of inertia it resists the angular change and torque more so requires more torque to slow down and move in

the other direction. Which is why we see larger amplitudes for higher J. With lower J, it takes less time to slow, stop and start the Satellite turning.

5 Question 5 (6 in the handout)

```

1 clear
2 clf
3
4 G = tf(6, [1 7 6 0]);
5 C = [4, 5, 6, 7, 8];
6
7 Legend = cell(length(C), 1);
8 i = 1;
9 figure(1)
10 hold on
11 for a = C
12     sys = feedback(series(a, G), 1);
13     step(sys, 40);
14     Legend{i} = strcat(num2str(a), 'V Step');
15     i = i + 1;
16 end
17 hold off
18 legend(Legend)
19
20 figure(2)
21 subplot(1,2,1)
22 H = feedback(series(tf([0.1 0.1], [1 0]), G), 1);
23 step(H, 100);
24 title("Step Response with Controller  $C = 0.1(1 + \frac{1}{s})$ ", 'interpreter', 'latex')
25 subplot(1,2,2);
26 pzmap(H);

```

For the first part of this question, we can see in fig. 5 that a gain controller of 7 provides no damping on the system. This indicates that anything higher than 7 will make the system unstable and this is the maximum value that can be used for stability.

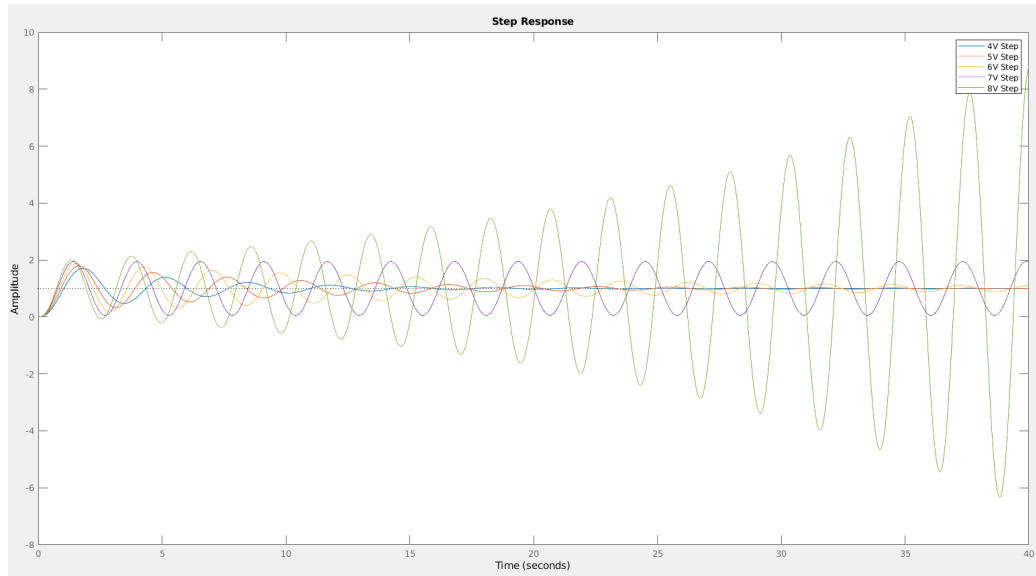


Figure 5: Gain Controller Step Response Stability

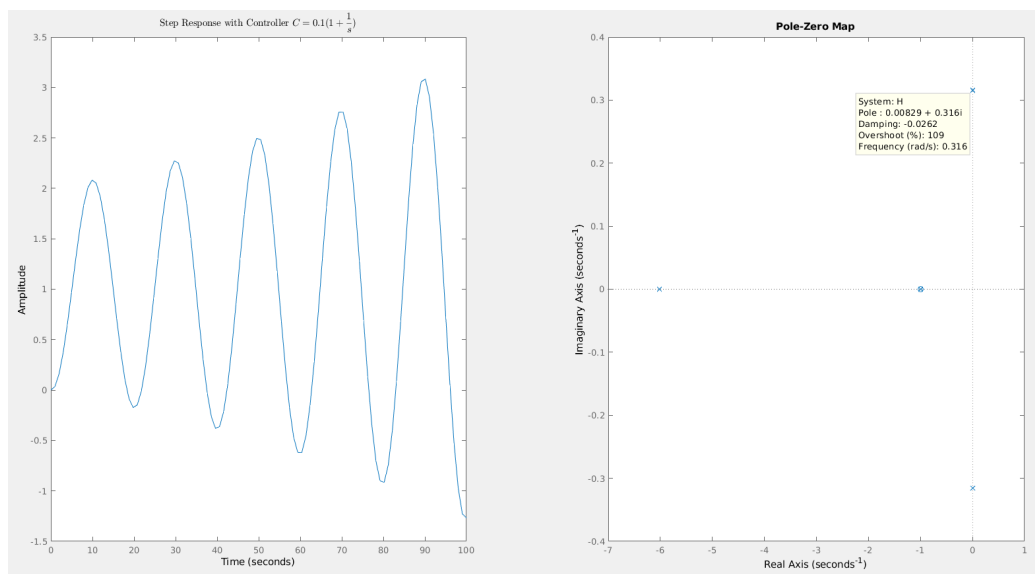


Figure 6: PI Controller Stability with low steady state gain