Lab #3 Report
Digital Simulation of a Closed Loop System

Kush Gupta
Section AB*6*
Vishrut Dixit
Khaled Al shehri

January 24, 2018

Total: ___/45

1 Results ___/15

1.1 Plots ___/6

See Figures 1 and 2 for the responses to a unit step as the reference and disturbance, respectively.

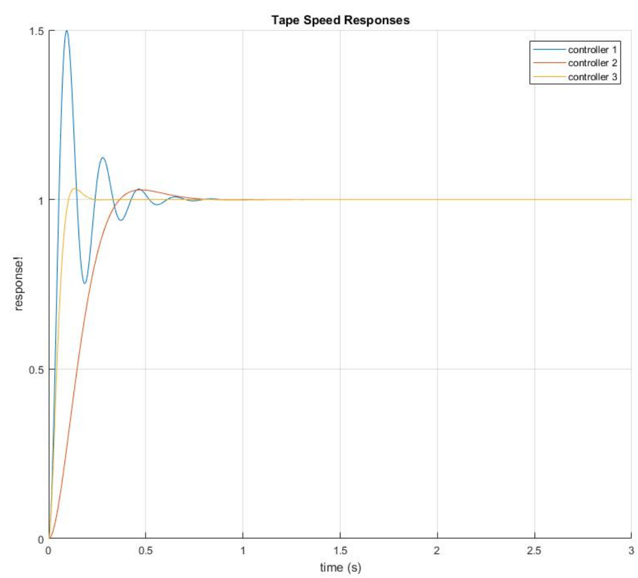


Figure 1: Response of three controllers due to unit step reference.

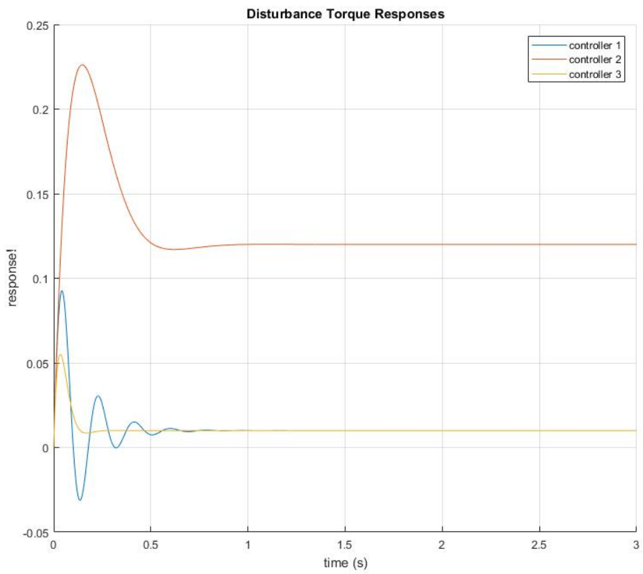


Figure 2: Response of three controllers due to unit disturbance.

Table 1: Time Response to a Unit Step for ω_r

parameters	Controller 1		Controller 2		Controller 3	
	Prelab	Lab	Prelab	Lab	Prelab	Lab
M_p [%]	49.8%	49.8%	2.8%	2.84%	2.8%	3.2%
t_r [s]	0.035	0.036	0.220	0.229	0.067	0.064
t_s [s]	0.402	0.390	0.320	0.312	0.097	0.088
K	19.4		1.067		19.4	
K_r	1.031		1.562		1.031	
K_d	0		0		0.031	

1.2 Step Response to ω_r ___/6

1.3 Comparison ___/3

(Compare M_p, t_r and t_s from Prelab with those from Lab. Are they close to each other? Which controllers met the specifications?)

The calculated values for each of the controllers were very close to the actual values that we observed in each case that we evaluated. This is likely because we are currently using a simulation of the control system to obtain the responses, which does not introduce any extraneous variables into the output. Based on the values that we obtained, it is clear that not all of the controller configurations were particularly successful.

Controller 1 had a large overshoot ($M_p = 0.417$) but otherwise achieved a disturbance response < 0.01. However, because of its $\zeta = 0.217 < 0.75$, we can see that the response is not ideal given our constraints.

Controller 2 had a disturbance response that was very high (0.12), meaning that any disturbance encountered by the system would have a large effect on the output of that system. We want to prevent this because a good controller should not be heavily affected by disturbance.

Controller 3 was designed to meet the specifications, and it did as can be seen from the data we gathered. The introduction of the tachometer into the system allowed us to counter the effect of the disturbance more effectively than in Controller 2 without unintentionally increasing the overshoot like was observed in Controller 1. Thus, Controller 3 was the most effective of the ones we designed in this lab, as it responded to a step input quickly and was not drastically affected by disturbance.0

2 Deriving e_{ss} Components ___/12

(For the system in Figure 3.1 in Lab Manual, derive the relationship between steady state error ($e_{\text{ss}} = \omega_r - \omega$) and natural frequency ω_n. Consider the error as a function of both ω_r and τ_d, and model these as step inputs. Since the system is linear, superposition allows the two components to be calculated separately and then summed. Notice that e_{ss} is not the same thing as "e" in the block diagram. ($e = K_r\omega_r - \omega$, this is the error signal.))

$$\omega_n^2 = 36 + 60K$$

$$e_{ss} = \Omega_r(s) - \Omega(s) = \omega_r - \frac{K_r(\omega_n^2 - 36)}{s^2 + 15s + \omega_n^2}\omega_r - \frac{4s + 12}{s^2 + 15s + \omega_n^2}\tau_d$$

Hint: Using the Final Value Theorem, solve the following subproblems:

- What is e_{ss} due to a step in ω_r ($\tau_d = 0$)? In order to minimize this error component, what value of ω_n should we choose?
- What is e_{ss} due to a step in τ_d ($\omega_r = 0$)? In order to minimize this error component, what value of ω_n should we choose?

By the Final Value Theorem, we can see that a step response in ω_r will result in the following error value:

$$\omega_r(1 - K_r + \frac{36}{\omega_n^2})$$

Based on this, $\omega_n^2 = \frac{36}{K_r - 1}$ will minimize the error term.

Similarly, using the Final Value Theorem when the disturbance is modeled as a step response results in the following error value:

$$-\frac{12}{\omega_n^2}\tau_d$$

Thus, a larger value of ω_n^2 will minimize the error in this case. This makes sense given the fact that we calculated on the prelab that values of K that are larger will be more stable.

From this, we can see that there is a conflict between the values of ω_r necessary to minimize the overall error in the system. As such, it is necessary to choose values that result in specifications that are partial to the application where this controller is being used.

3 ζ and ω_n in Controller Three ___/18

3.1 Write down the expressions of ζ and ω_n ___/12

(For controller 3, derive the relationship between ζ, ω_n and the gains K and K_d.)

$$\zeta = \frac{15 + 60K_dK}{2\sqrt{36 + 60K}}$$
$$\omega_n = \sqrt{36 + 60K}$$

Discussion: If we increase K, what happens to ζ and ω_n? And at what rate does ζ change (*linearly, exponentially, as* K^2 etc)? How does ω_n change? What if we increase K_d? What happens to ζ and ω_n?

Based on the relations that we derived, the values of ω_n and ζ will vary based on K and K_d. As K increases, ω_n will increase by $\sqrt{K}$. The value of ζ also increases by the same factor of $\sqrt{K}$.

The value of ω_n does not change with K_d, while ζ changes at a linear rate corresponding to K_d

3.2 Using these equations, show how the pole locations change as $K_d > 0$ increases in value ___/6

$$\text{Poles} \quad s_{1,2} = \frac{-2\zeta\omega_n \pm \sqrt{(2\zeta\omega_n)^2 - 4\omega_n^2}}{2}$$

(As K_d increases, what is the trajectory of poles? Again, you can use MATLAB® to sketch a plot.)

As can be seen in the figure, the poles vary based on the value of K_d that is chosen. For larger values of K_d, the dominant pole will converge to zero while the inferior pole will move further and further away from the origin into the LHP. This is seen in the figure.

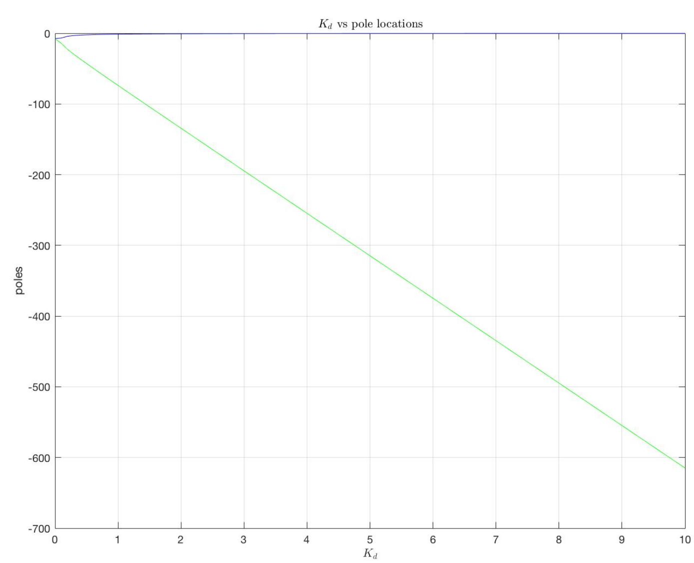
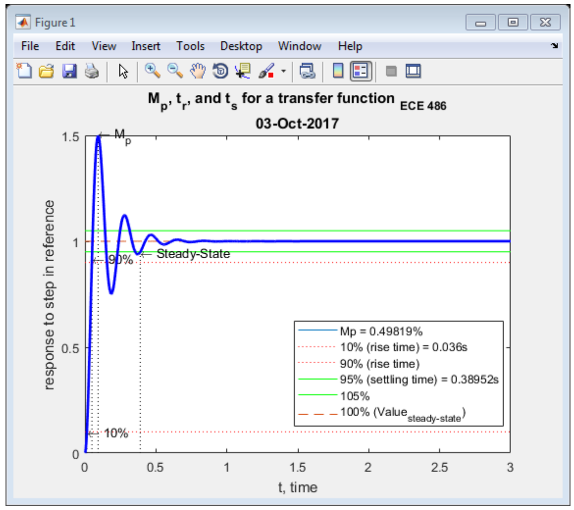


Figure 3: Pole locations for $0 < K_d <$ value.



Sample plot showing response specification values.

MATLAB Code used in Lab #3:

```
%
% Setup values for SIMULINK and then display results
%

K = 19.4;
Kr = (3/(5*K))+1;
J = 0.25;
B = 3;

%------------------

PlotResponse(ctrl1_td_t, ctrl1_td_w, ctrl1_wr_t, ctrl1_wr_w);

%
% Plot the response for a particular controller
%

function [] = PlotResponse(tout1, w1, tout2, w2)
close all
hold on
grid on
plot(tout1, w1);
plot(tout2, w2);
legend('td_w', 'wr_w');
xlabel('time (s)');
ylabel('response!');
title('Controller 2 Responses');

%
% Calculate values of $M_p$, $t_r$, and $t_s$ for various responses and display results
%

[Mp_1, tr_1, ts_1] = StepResponseMetrics(ctrl1_wr_w,ctrl1_wr_t, 1, 1)
[Mp_2, tr_2, ts_2] = StepResponseMetrics(ctrl2_wr_w,ctrl2_wr_t, 1, 1);
[Mp_3, tr_3, ts_3] = StepResponseMetrics(ctrl3_wr_w,ctrl3_wr_t, 1, 1);
```

```
% ----  pole calculations
K = 1;
Kd = 0:.1:10;

zeta = (15+60*K*Kd)/(2*sqrt(36+60*K));
w = sqrt(36+60*K);

p1 = (-2*zeta*w + sqrt((2*zeta*w).^2 - 4*(w^2)))/2;
p2 = (-2*zeta*w - sqrt((2*zeta*w).^2 - 4*(w^2)))/2;

plot(Kd, p2, 'g')
hold on
grid on
plot(Kd, p1, 'b')

xlabel('$K_{d}$', 'interpreter','latex');
ylabel('poles');
title('$K_{d}$ vs pole locations','interpreter','latex');
```