Lab 4 Report

Position Control of a Permanent Magnet DC Motor

ECE311

Lab Group: 17

Name of Students: Alexander Mark, Kyu Bum Kim, Jack Khoueiri

Date: December 6, 2021

Output 1

The parameters we found/were given are:

$$initial\ crossover\ = 20\, rad/s$$

$$K = 72.0044$$

$$T = 0.0889$$

$$\alpha = 0.1$$

$$\underline{w} = 35.6\, rad/s$$

How we arrived at these values:

- **initial_crossover**: this value was given to us.
- **K:** K is designed such that the gain crossover frequency of K*G is equal to the previously mentioned initial_crossover. This will cause the magnitude Bode plot of K*G to cross 0 dB at w = initial_crossover = 20, as intended. The calculations are as follows:

$$K * |G(i * initial\ crossover)| = 1$$

$$K = \frac{1}{|G(20i)|}$$

$$K = 72.0044$$

- **Omega_bar**: omega_bar is chosen such that the frequency ω_{max} of C1 (where phase is maximum) is equal to the crossover frequency of C1*G. This is to maximize the phase margin contribution of the LEAD controller. On the bode plot, we find where $\underline{w}$ satisfies (this is also the value returned by margin() in Matlab):

$$20log(\left|K * G(i\underline{w})\right|) = -20log(\sqrt{\frac{1}{\alpha}})$$

$$\frac{K*G(i\underline{w})}{\sqrt{\alpha}} = 1$$

We find this value to be:

$$\underline{w} = 35.6\, rad/s$$

- **T:** since we have the required values, we applied the formula to solve the value of T:

$$T = \frac{1}{\underline{w}\sqrt{\alpha}} = 0.0889$$

- **alpha**: this value was given to us.

The Bode plot of K*G shows that it is only a shifted version of G as we have designed K such that the crossover frequency is 20 rad/sec. The Bode plot of C1*G further increases crossover frequency to 35.6 rad/sec, but also, unlike K*G, it greatly improves the phase margin (PM) to the desired value of 55°. These observations are found in figure 1.

Bode Plots of LEAD Controller with Plant

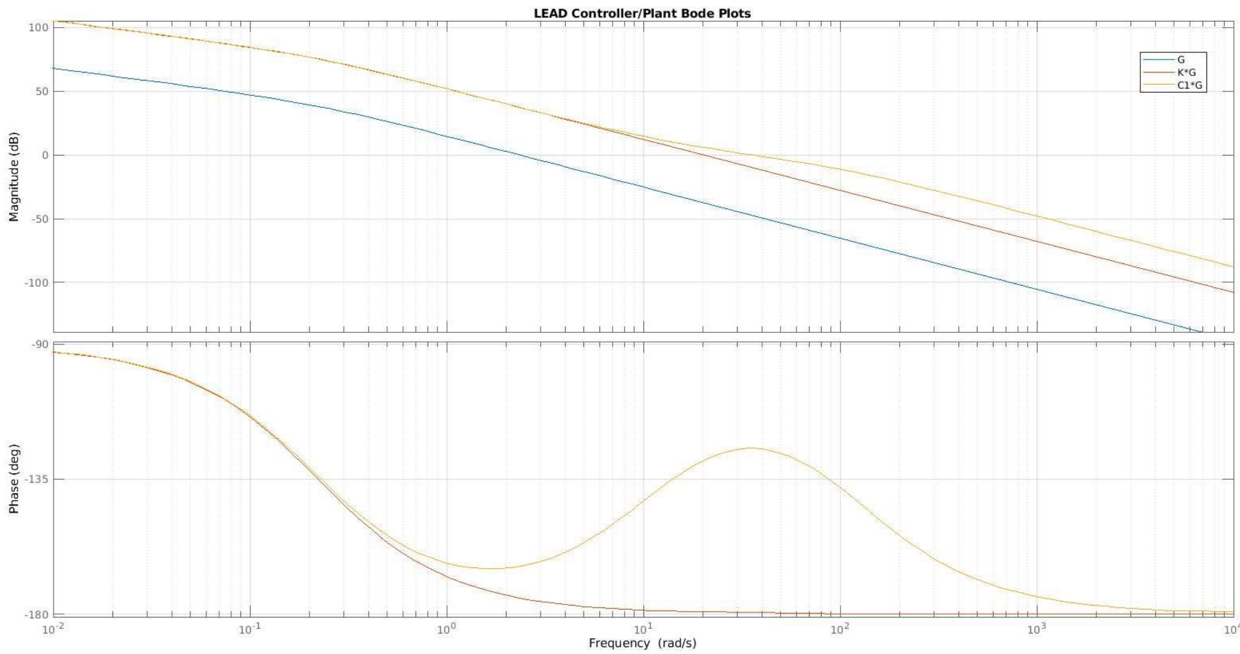

*Figure 1: Bode Plots of G, K*G, C1*G*

The LEAD controller meets SPEC 2 as according to the graph. The crossover frequency of the loop transfer function C1*G is equal to 35.6 rad/s, which is between 20 and 50 rad/s. It also meets SPEC 3 as the PM of 55° is greater than 45°. It does not yet meet SPEC 1 because at the moment, we do not have disturbance rejection. According to the internal model principle (IMP), in order for disturbance rejection to be met, C(s) needs to contain the poles of D(s) but we do not currently have this. The addition of the PI controller in the next section will achieve this.

Output 2

We set Ti to the value shown below:

$$T_I = \frac{10}{\underline{w}} = 0.2812$$

$$\underline{w} = crossover\ frequency\ of\ C1 * G$$

By setting 1/Ti to a value 10x smaller than the initial crossover frequency, we make sure that the PI controller has little effect on the original crossover frequency and the corresponding PM. We observe that this is true in figure 2 as these were the crossover frequencies and PM of C1*C2*G:

$$PM = 49.6°$$

$$w_c = 35.7\ rad/s$$

Phase Margin/Crossover Frequency of C1*C2*G

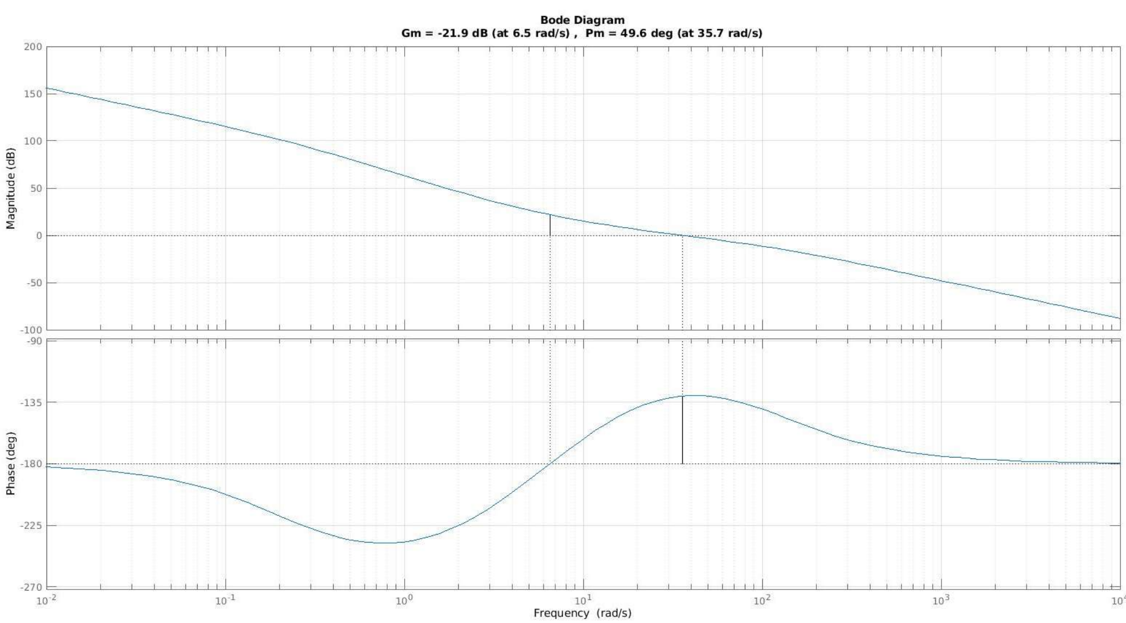

*Figure 2: Phase Margin and Crossover Frequency of C1*C2*G*

In order to check whether SPEC 4 is met, we need to verify that the magnitude plot of G/(1+CG) is below the -34 dB line because this will show that for any given frequency of sinusoidal input of D(s), the transfer function G/(1+CG) will attenuate the signal to at least 2% of its value (20*log(0.02) = -34 dB). This is exactly what SPEC 4 outlines. As seen in figure 3, our controller does meet SPEC 4 as the magnitude graph sits below -34 dB for any frequency value.

Bode Plots of G/(1+CG) with Cascaded Controller (LEAD + PI)

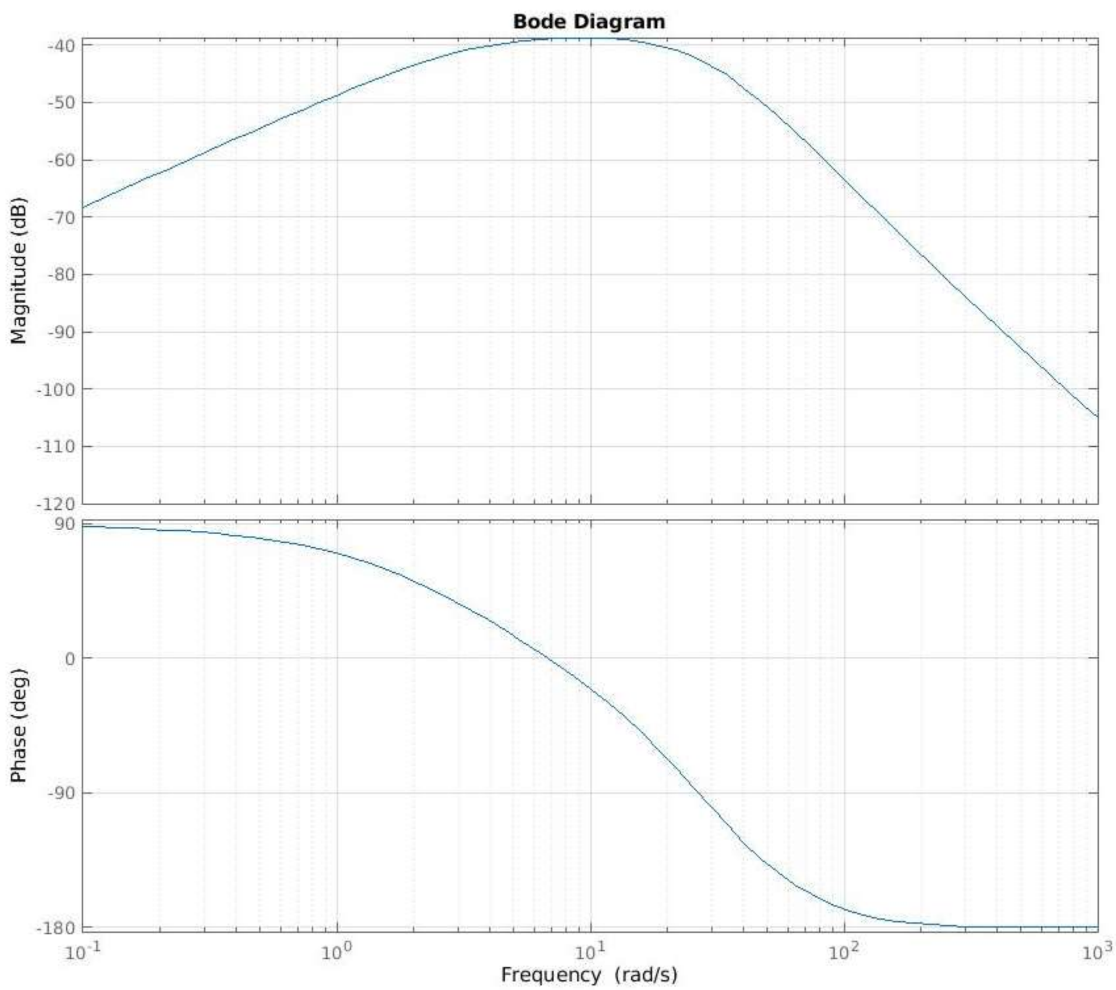

Figure 3: Bode Plots of G/(1+CG)

In figure 4, we present the 2 step responses from the contribution of the reference signal and disturbance. We see that SPEC 1 has been met from these graphs. Asymptotic tracking is present because when looking at y(t) from r(t) contribution, as time approaches infinity, the output signal converges to $\Theta_{des} = \frac{\pi}{2}$. Disturbance rejection is present because when looking at y(t) from d(t) contribution, as time approaches infinity, the disturbance has no effect on the output (ie. the graph goes to 0).

Control Output y(t) with Cascaded Controller (LEAD + PI)

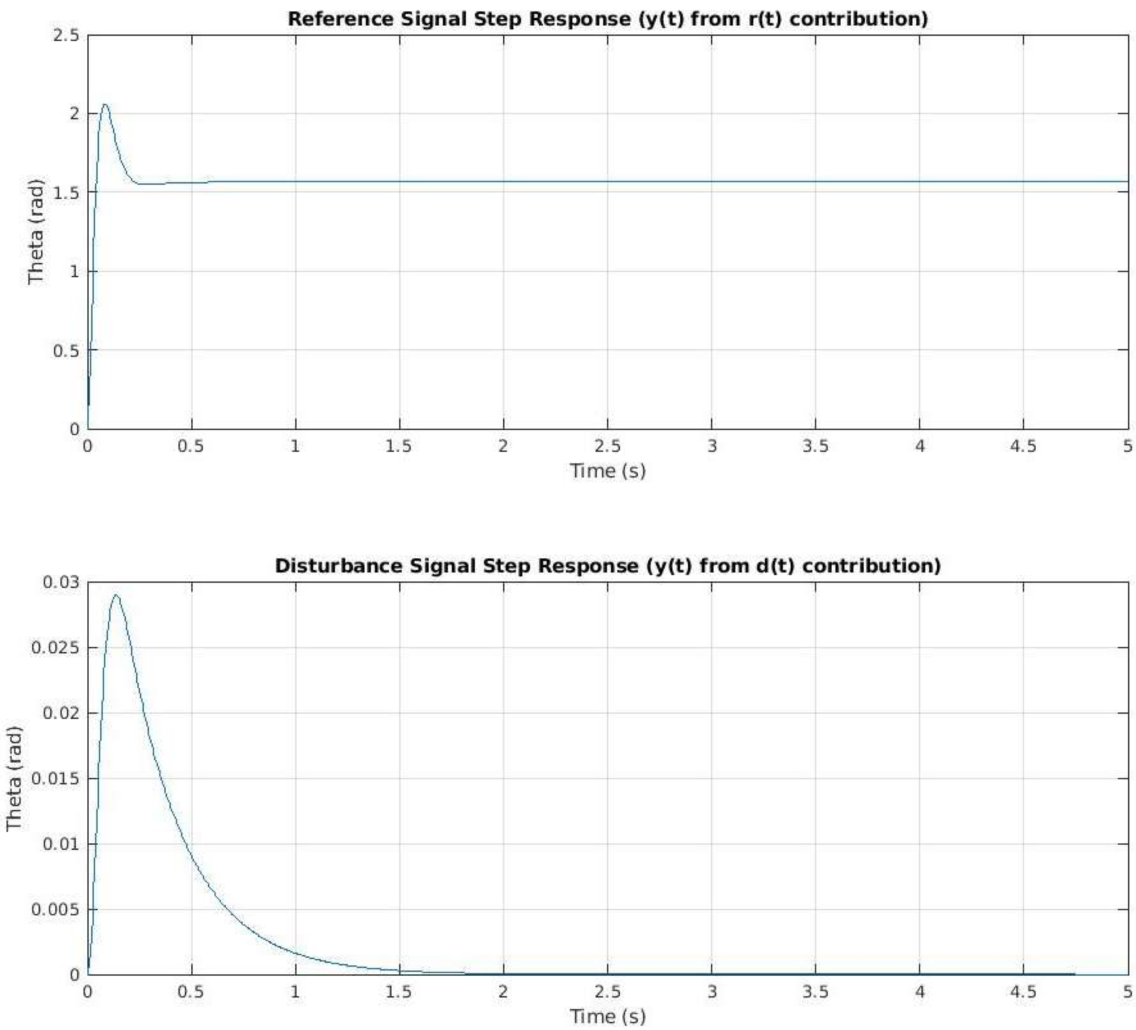

Figure 4: Step Response for y(t) from Contributions of d(t) and r(t)

With the *stepinfo* command, we found the settling time and percent overshoot (OS) of y(t) from r(t) contribution to be the following for our controller:

$$T_s = 0.1986s$$

$$\%OS = 31.17\%$$

Output 3

Ti Value (Wc = 35.6)	Settling Time (seconds)	Overshoot (%)
0.2812 (Wc/10)	0.1986	31.17
0.7029 (Wc/25)	0.2240	26.21
1.4058 (Wc/50)	0.2313	24.44
2.1087 (Wc/75)	0.2330	23.84
2.8116 (Wc/100)	0.2337	23.54

Table 1: Tuning Ti of the PI controller

We started with a Ti value of 10x smaller than the crossover frequency, gradually increasing the multiplier to 100x. As we increase Ti, we can see that the percentage OS decreases while the settling time increases.

As the Ti value decreases, 1/Ti increases. As we shrink the crossover frequency, the 1/Ti value decreases. Because the integral part of the controller charges based on the error and discharges based on the 1/Ti term, the larger the 1/Ti term, the greater the discharge resulting in a larger OS but a smaller settling time. The error will discharge with a larger multiplier, meaning the controller will try to converge to the desired value at a higher rate which causes the OS to increase. Because of the higher rate at which the controller tries to control the error, the settling time decreases.

Initial Crossover (rad/s)	Settling Time (seconds)	Overshoot (%)	Maximum Magnitude (dB)
10	0.4006	30.60	-25
20 (default)	0.1986	31.17	-40
40	0.0989	31.45	-50
80	0.0494	31.60	-60
160	0.0247	31.67	-75

Table 2: Tuning Initial Crossover of the System

We started with a lower initial crossover frequency of 10 rad/s and doubled this frequency every time up to 180 rad/s. We can see that the settling time consistently decreases as we increase initial crossover frequency. The maximum magnitude decreases with an increase in crossover frequency therefore, attenuates a sinusoidal disturbance better. The OS also grows slowly as initial crossover gains.

We understand that as we increase the crossover frequency, we are increasing the bandwidth of the closed-loop system (CLS) transfer function. This is due to the derivation of our bandwidth's maximum and minimum bounds, Wc < Wb < 2Wc. This explains why the settling time of the CLS decreases. We derived in class that settling time is inversely proportional to the bandwidth of the system. Therefore, as we increase the crossover frequency, we are in fact increasing the bandwidth which results in the reduction of the settling time of the system.

Moreover, from further derivations based on zeta, the percentage OS and PM formulas are inversely proportional; meaning a higher PM results in a lower percentage OS. As we increase the crossover frequency, the PM changes in small increments hence we only get slight changes in OS. This is because the controller we designed, the phase margin follows the crossover frequency. Therefore, the phase margin changes slightly with a change in crossover frequency resulting in minimal changes to the OS.

Increasing the initial crossover, we are increasing the gain of C. Therefore, the denominator of G/(1+CG) becomes a more significant value. When obtaining the magnitude of the disturbance, we can conclude that there will be larger magnitudes being subtracted hence decreasing the magnitude of the system and therefore increasing the disturbance attenuation.

Output 4

Below are the three figures produced using the baseline control parameters using an actuator saturation as the input to the system:

$$K = 72.0044$$

$$T = 0.0889$$

$$\alpha = 0.1$$

$$T_I = \frac{10}{\underline{w}} = 0.2812$$

Control Input u(t) with Actuator Saturation

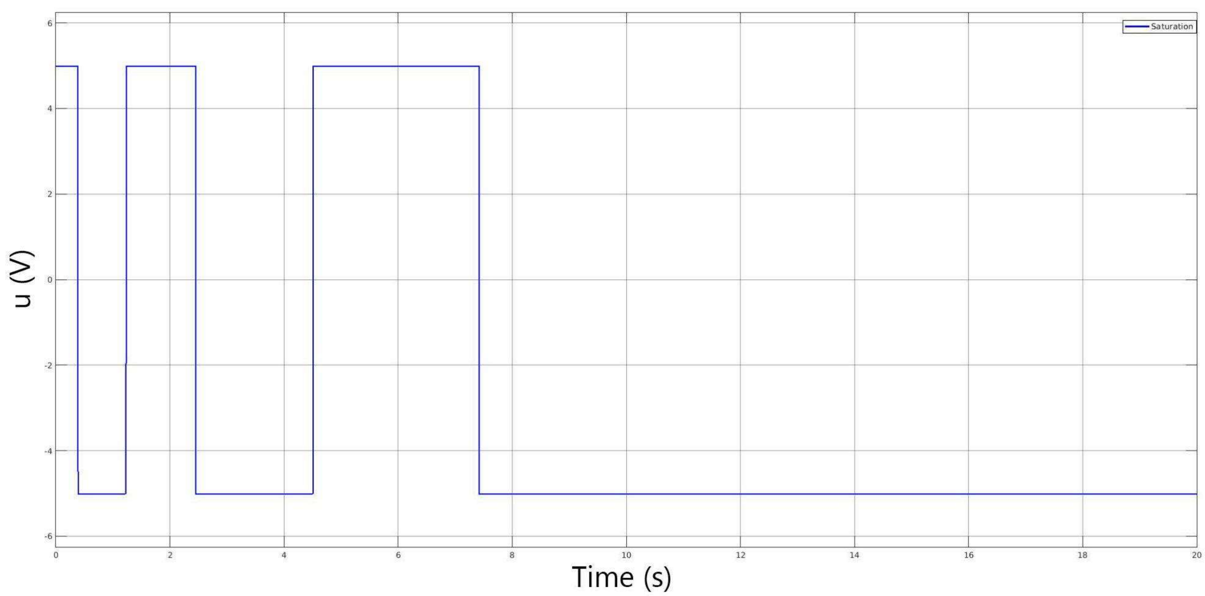

Figure 5: Control Input u(t) using Baseline Control Parameters

Control Output y(t) with Actuator Saturation

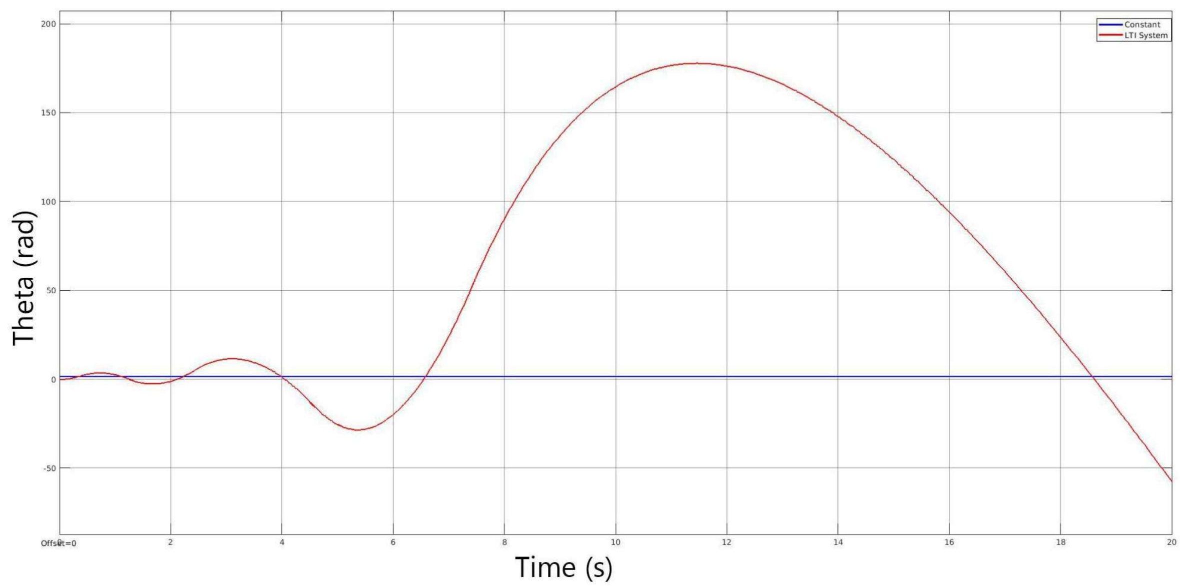

Figure 6: Control Output y(t) and the Reference Signal (thetades = pi/2) using Baseline Control Parameters

<u>Tracking Error e(t) with Actuator Saturation</u>

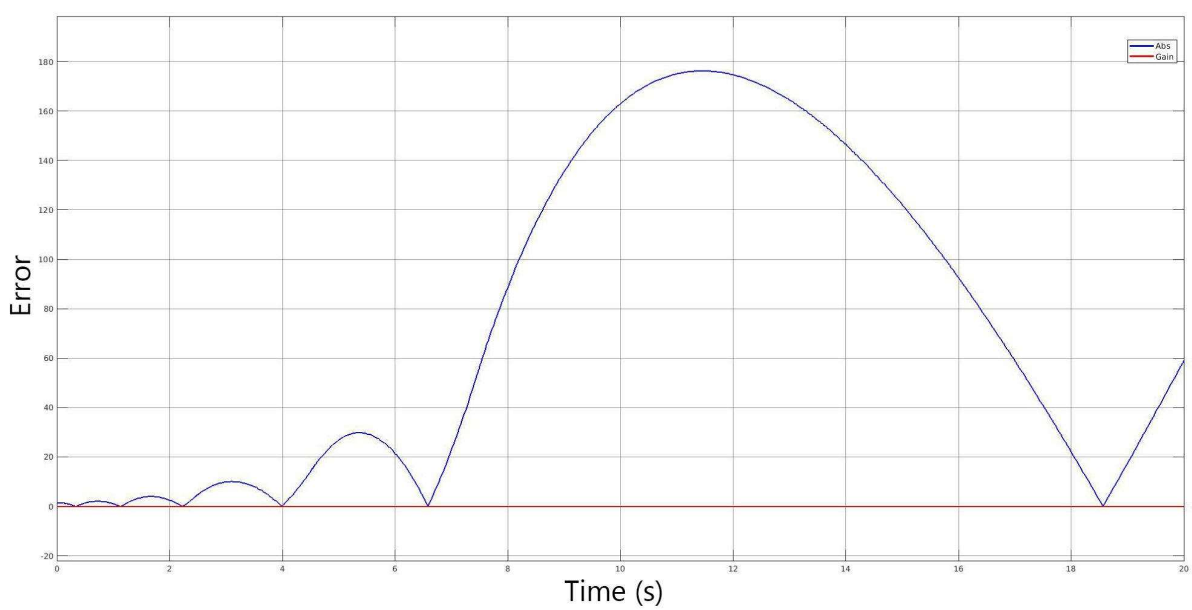

Figure 7: Tracking Error e(t) using Baseline Control Parameters

We can see from figures 5 - 7, the presence of the actuator actually eliminates the settling time of the closed loop system for the baseline parameters. The controller does not understand that the output is saturated to a constant value of +/- Vlim. This causes the controller to be unable to input precise inputs into the plant. As a result, the controller is easily able to overshoot the desired value and is unable to control the plant to steady state; it oscillates further and further from it instead. The system never settles to the desired reference signal value.

To tune the initial_crossover, we both increased and decreased the value. We found that the limiting value for this parameter is:

$$initial\ crossover = 15\ rad/s$$

Any value above 15 rad/s resulted in instability. Instability occurs because with large values of initial_crossover, the gain of the controller becomes higher. Therefore, the controller will try to output larger changes on the plant and these changes will get saturated to the limit values, making the controller unable to provide finer adjustments. This causes the system to become unstable as the controller is unable to reach steady state. For smaller values of initial crossover, the settling time decreases.

Similarly to output 3, we increased the Ti value gradually. At around when Ti equals 0.44, the model approaches a steady state value prior to the disturbance. Any value of Ti lower than that, instability ensues. As we increase Ti, the rate at which the integrator discharges error decreases, hence reducing the amount of error reduction within the system. In the presence of an integral action controller, the controller accumulates the error. Because of the introduction of the actuator saturation, the output of the system will experience more drastic changes; small changes are not possible as the input is only Vlim or -Vlim to the plant. With less precision on changing the output, the output will have greater error and the integral controller will accumulate all these large error values. Eventually, the controller will be trying to correct for larger and larger amounts of error, causing instability in the system.

Output 5

By setting K_{aw} = 1/Ti = 3.56, we get the tracking error graph seen in figure 8. Using the cursor tool in Matlab, we measured the settling time for this value of K_{aw} to be:

$$T_s = 0.75s$$

<u>Tracking Error e(t) with Integrator Anti-Windup</u>

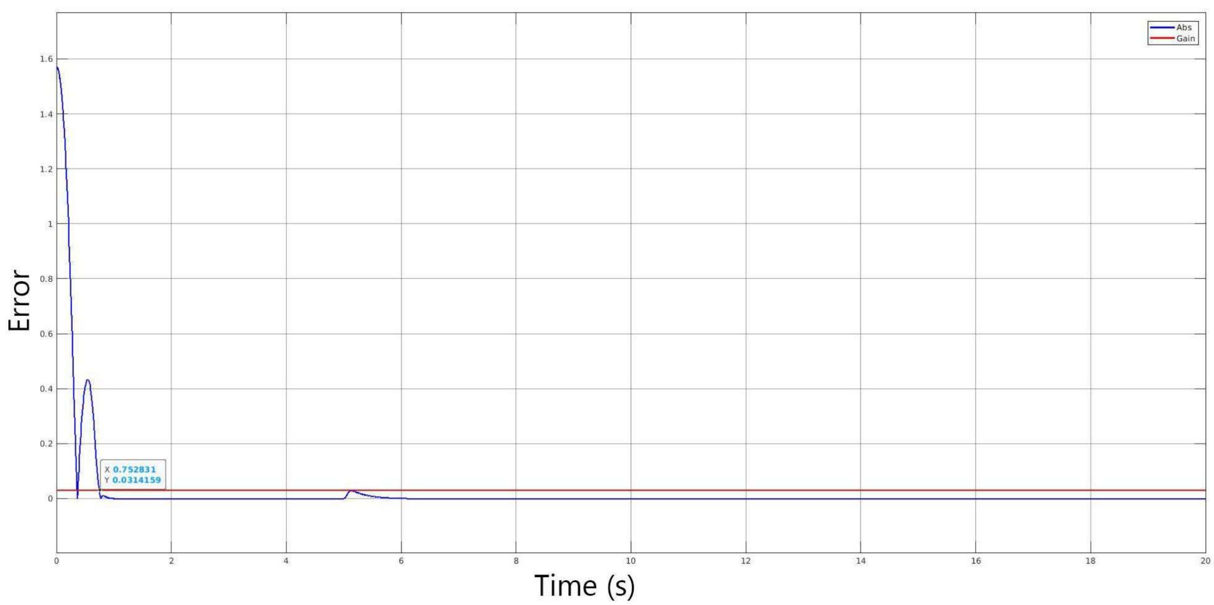

Figure 8: Tracking Error e(t) when Kaw = 1/Ti. The marked value shows the settling time measured empirically (Ts = 0.75).

The antiwindup controller is many times better than the baseline controller we had in output 4. The settling time is a finite and a small value (0.75s) here which is better than the untuned unstable system we had in output 4. As well, the percent overshoot is smaller than output 4 because this system is able to better manage the actuator saturation and control the output to the desired value (the integrator does not get out of hand with the integrator antiwindup). Furthermore, the introduction of the disturbance is quickly mitigated and is not an issue while it appears to have caused the system to go unstable in output 4.

To tune K_{aw}, we found that large values, beyond a threshold, will cause the integrator to be discharged too much, thereby increasing the settling time. With smaller values of K_{aw}, the controller does not discharge the large accumulation of the integrator term fast enough, and we see that the tracking error graph becomes very similar to that of output 4 (figure 7).

As mentioned before, if we only increase the value of K_{aw} slightly, then the settling time actually improves because it is discharging the effect of the integrator just enough to make it converge to steady state quicker. The results we found that achieved the best settling time are found in figure 9 and below:

$$K_{aw} = 1.3 * 1/T_I = 4.65$$

$$T_s = 0.63s$$

Tracking Error e(t) with Integrator Antiwindup

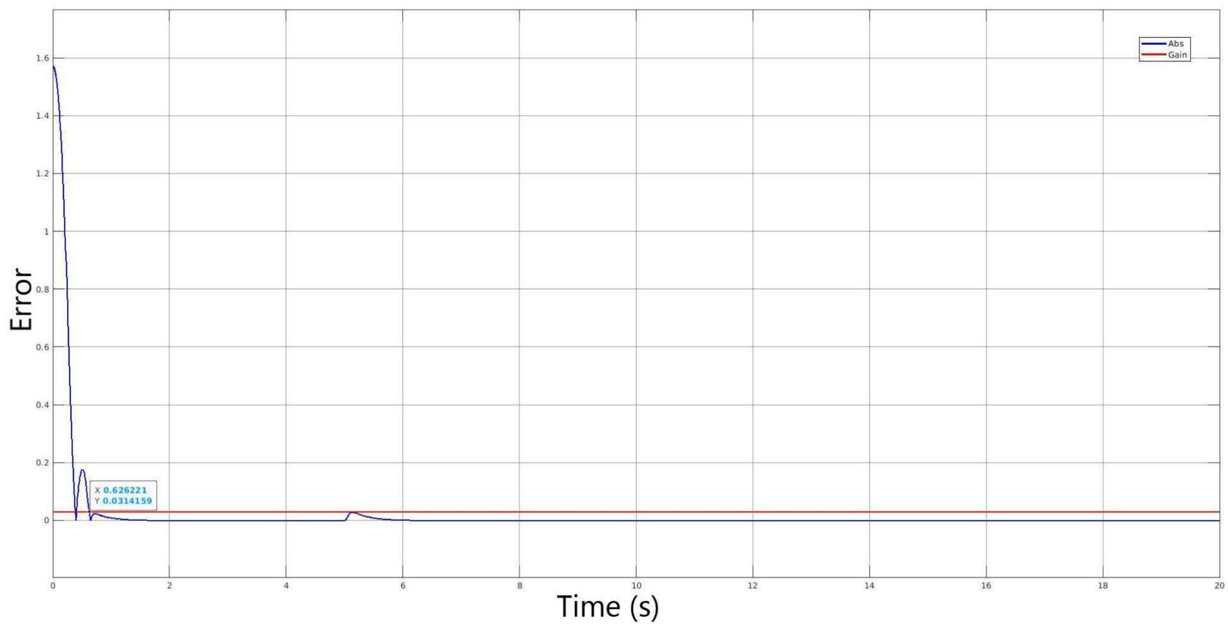

Figure 9: Tracking Error e(t) when Kaw = 1.3(1/Ti). The marked value shows the settling time measured empirically (Ts = 0.63).*

With the tuned value of K_{aw}, the performance has improved slightly in terms of the settling time. As well, the percent overshoot is smaller and this is expected because the effect of the integrator is dampened with a larger K_{aw}. This overall, makes the servomotor better at controlling the output to the desired reference signal.