ENGS 26 Final Project Report

Walden Marshall

March 2021

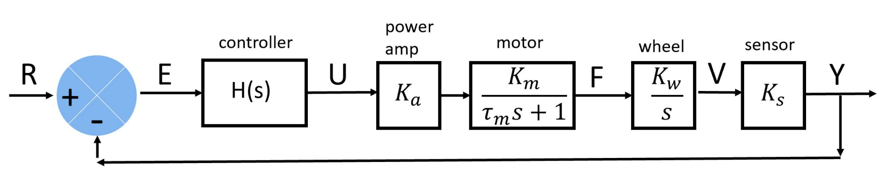
I Overview

In this project, we were tasked with designing a compensator for the Lego car used in previous labs. The Lego car would now be equipped with an infrared distance sensor. Using the output signal from the sensor, our compensator was supposed to bring the car to a stop at specific distance from a wall. Of course, there are numerous ways to do this, and the compensator must be fine tuned in order to provide an acceptable response.

II System Modeling

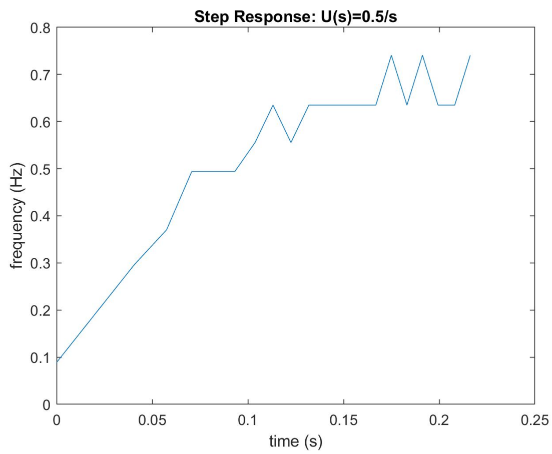
Before designing a compensator, we must develop a model of the system. The general model for the system is as follows.

Some control effort voltage, $U(t)$, is the input to the power amplifier. The power amplifier, which is modeled as a constant gain, K_a, drives the car's motor. The car's motor is modeled as a standard lowpass filter with two parameters: a DC gain, K_m, and a time constant, τ_m. Considering the motor's frequency of rotation as its output, the next block is the wheel, which translates the frequency of rotation into a position. Since the car's velocity is proportional to the frequency of the wheels, and position is the integral over time of velocity, the wheel's transfer function is modeled as an integrator with some gain, K_w, equal to the ratio of the car's velocity to its frequency of rotation. Finally, the optical sensor has a transfer function from the position of the car to the sensor's output voltage, which is modeled as a gain, K_s. The block diagram for the closed loop system is shown below.



The first unknown parameter to consider is K_a. When the power amplifier receives a 2.5 V input, its output is equal to the battery voltage. I measured my battery voltage to be very close to 10 V, so $K_a = 4$ V/V.

Next, the motor needed to be characterized. To find the time constant, I measured the signal from the Lego car's built in tachometer as the car was given an input step response. Because the frequency-to-voltage converter included in the lab kit introduced a time constant and would corrupt the data, I instead took the raw tachometer signal and processed it in MATLAB to create a plot of frequency versus time. The portion of this plot relevant to finding the time constant is shown below.



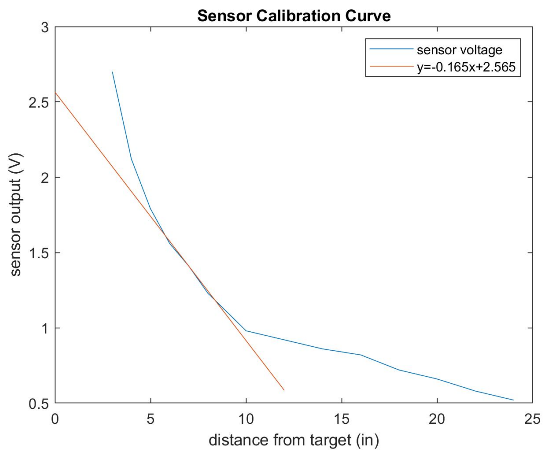
In the measured time domain response, the input to the power amplifier was a 0.5 V step, and therefore the input to the motor was a 2 V step. The response of the motor to a 2 V step should take the following form:

$$f(t) = 2K_m(1 - e^{-\frac{t}{\tau_m}})$$

Using MATLAB's powerful curve fitting toolbox, I was able to obtain values for both parameters. I found $\tau_m = 74$ ms and $K_m = 0.36$ Hz/V. To further support my result for the value of K_m, I put a piece of colored tape on the wheel of the car and took a video of the car in steady state motion. I watched the video in slow motion, counting the number of rotations, and noting the time it took. I found that 0.36 Hz/V was indeed the DC gain.

The next parameter I determined was the gain of the wheel, K_w. Since K_w is the ratio between the frequency of the wheel and the velocity of the car, K_w is equal to the circumference of the wheel. Using calipers, I measured all four wheels, and found an average diameter of 1.195 inches, meaning the circumference was 3.75 inches and $K_w = 3.75$ (in/s)/Hz.

The final unknown parameter was the gain of the sensor. While the sensor output voltage as a function of distance is actually nonlinear, it was suitable to approximate it as linear around some equilibrium. A curve of the sensor voltage versus distance from a reflective object is shown below, along with an estimate of the tangent line when the sensor is 7 inches from the target.



The sign of the slope is negative, but that is because it is the distance from the target, rather than the distance the car has traveled. Hence, $K_s = 0.165$ V/in. Note that this approximation is only valid when the car is close to 7 inches away from the target. Therefore, the value of the reference should be set to the voltage the sensor outputs when it is 7 inches away from the target, which was measured to be 1.41 V.

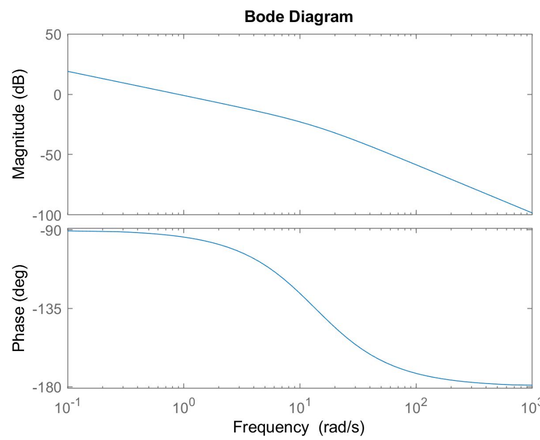
III Open Loop Analysis

The open loop transfer function, $G_{ol}(s)$ from $U(s)$ to $Y(s)$ is the product of all the blocks between U and Y.

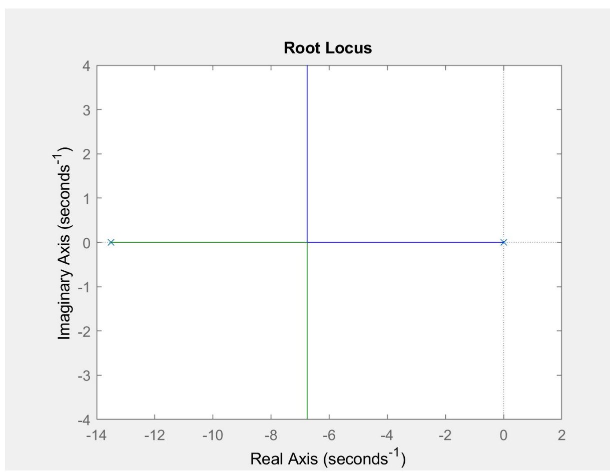
$$G_{ol}(s) = \frac{K_a K_m K_w K_s}{s(\tau_m s + 1)}$$

The step response is not a particularly useful tool to analyze the open loop system, because the step response will exponentially approach a line of slope $K_a K_m K_w K_s$ rather than approach a final value. This makes sense because without a feedback signal, a constant voltage to the motor will move the car forever, approaching a constant velocity.

The transfer function has poles at the the origin and at $s_p = -1/\tau_m$, so the magnitude Bode plot will have a slope of -20 dB/dec until the point $1/\tau_m$, after which it will have slope -40 dB/dec. MATLAB confirms this result.



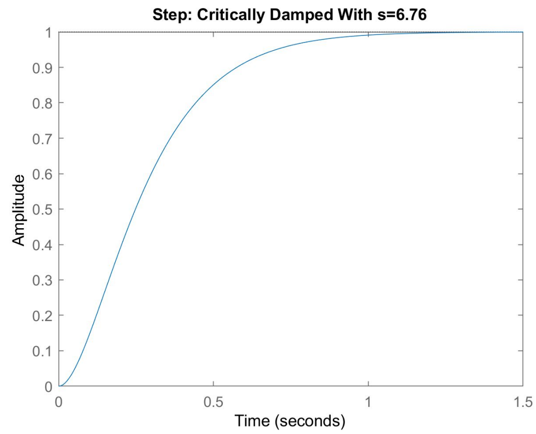
Consider a root locus plot of the open loop system.



As is expected for a system with two real poles and no zeroes, the points where the two poles converge is the average of the two.

$$\sigma_a = \frac{1}{2}(0 - 13.51) = 6.76$$

Therefore for a proportional controller, for the gain at which the critical damping is achieved, the location of the repeated pole is $s_p = 6.76\ \mathrm{s}^{-1}$. The step response for some abrbitrary system with a critically damped characteristic equation is shown below. I have normalized it so that it has zero steady state error.



MATLAB reports a rise time of 0.5 seconds and a settling time of 0.86 seconds.

IV Design Specifications

The first and most obvious design specifications are that my closed loop system must be asymptotically stable, and that it must have zero steady state error.

Since a rise time of 0.5 seconds and a settling time of 0.86 is achievable through a simple critically damped proportional controller, I will seek to make the system respond at least 20% faster in each category with my controller.

While speed of response is important, I need to also consider overshoot. One parameter that has a large effect on overshoot is the damping ratio, ζ. For many systems, a damping ratio of $\zeta = 1$ is the most desirable, but for this one, that may not be the case. I will consider any damping ratio between 1 and $\sqrt{1/2}$ acceptable, so long as other design specifications are met.

The reason that I choose $\sqrt{1/2}$ as my minimum allowable damping ratio is that this damping ratio is the one used in a second order Butterworth filter. The Butterworth filter has the property that is as underdamped as possible without having gain above the DC value near the resonant frequency. By ensuring that $\zeta \geq \sqrt{1/2}$, I also ensure that there is no frequency at which the system resonates.

The open loop system has a pole at 13.5 rad/s. I will hope to maintain a bandwidth of at least half that value, so the -3 dB frequency, ω_{bw}, should occur no lower than 6.8 rad/s.

Finally, my closed loop system should meet the standard gain margin requirement of at least 6 dB and the standard phase margin requirement of at least 30°.

V Compensator Design

Consider the compensator with a proportional gain K_p gain. The closed loop transfer function is the following:

$$G_{cl}(s) = \frac{G_{ol}H}{1 + G_{ol}H} = \frac{K_a K_m K_w K_s K_p}{\tau_m s^2 + s + K_a K_m K_w K_s K_p}$$

Based on the s^0 term, we can tell this system has zero steady state error. One of the largest benefits of an integral term in a controller is that it can correct a steady state error, but since there is no steady state error in this system when a proportional controller is used, an integral term loses much of its advantage. Therefore, I initially decided opt for a PD controller, which tends to provide faster responses.

However, after further examination of the system, a PD controller is not a wise choice. This is because the optical sensor is digital, and actually outputs discrete voltages rather than continuous values. The sensor's sharp changes between discrete levels results in derivatives that are either extremely high or zero. This will cause the derivative portion of the controller to behave extremely unpredictably, and not refelect the actual velocity of the car.

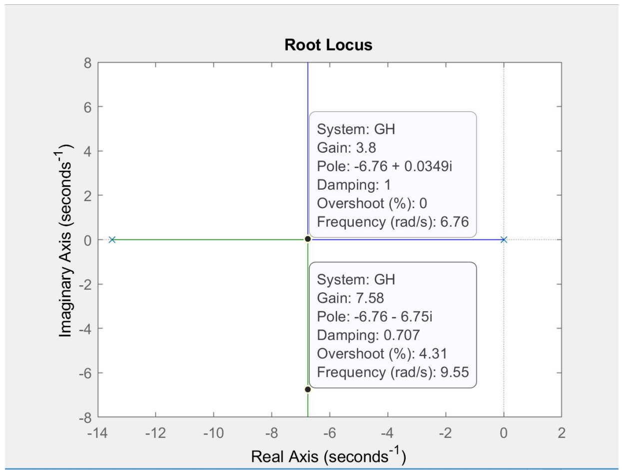
Since it is unwise to include a derivative gain, I will instead use a purely proportional compensator. Therefore, the controller transfer function is $H(s) = K_p$.

The product of the gains $K_a K_m K_w K_s$ appears multiple places in the transfer function. These gains are parameters of the system which cannot be adjusted, so to reduce the size of some future expressions involving these gains, I will express their product using its numerical value, $K_a K_m K_w K_s = 0.89 \text{ s}^{-1}$.

Therefore, the closed loop transfer function takes the following form:

$$G_{cl}(s) = \frac{G_{ol}H}{1 + G_{ol}H} = \frac{0.89K_p}{\tau_m s^2 + s + 0.89K_p}$$

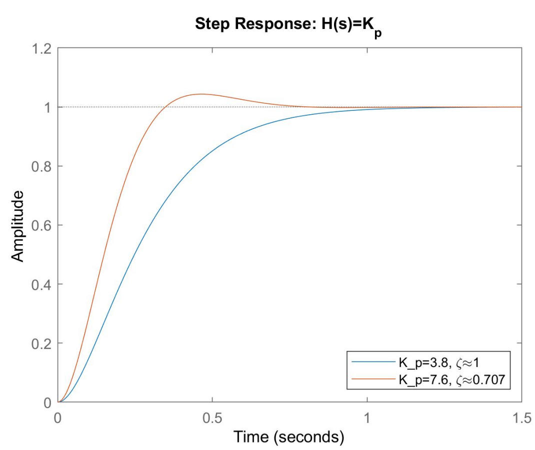
In order to determine a proper value for this gain, we must consider the root locus of $G_{ol}H = K_p \frac{0.89}{(\tau_m s+1)s}$. This root locus is shown below, with included MATLAB calculations of certain parameters at two points of interest.



The first point of interest is the critical damping point. If we were to restrict the system poles to being purely real, then this would be the optimal solution, since it ensures the dominant pole is as far away from the origin as possible.

The second point of interest is the point at which the imaginary part of the pole is equal to the real part of the pole. This is the location at which the damping ratio is $\zeta = \sqrt{1/2}$. As discussed in the Design Specifications section, this damping ratio maximizes bandwidth while ensuring that there is no resonance at any frequency.

From the calculated parameters of the root locus, it is apparent that the gain will have to be in the range $3.8 \leq K_p \leq 7.6$. In order to acquire a better understanding of what the gain should be, examine the step responses evaluated at the minimum and maximum values.



From the plot, it is obvious that the step response has a much faster rise time when $\zeta = \sqrt{1/2}$ at the cost of having overshoot. The actual values MATLAB

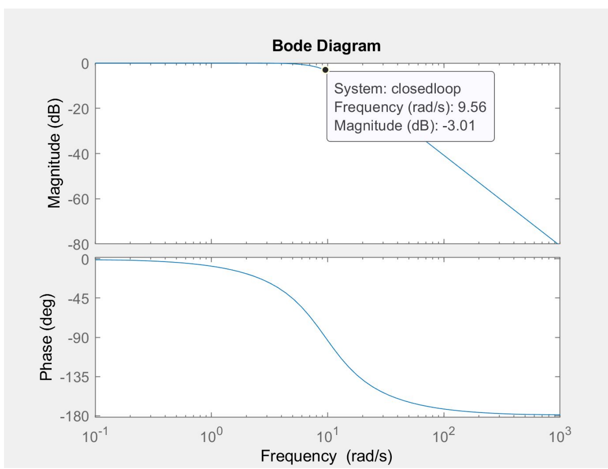
returns for rise time, settling time, and overshoot are shown in the table below.
‘

damping	critical	$\sqrt{1/2}$
rise time (s)	0.50	0.22
settling time (s)	0.86	0.62
overshoot (%)	0	4.3

The 4.6% overshoot is a small price to pay for a greatly improved rise and settling times. Therefore, it is best to have the system as underdamped as possible within the damping range I have designated as allowable, $\sqrt{1/2} \leq \zeta \leq 1$. So it seems the best value for the controller gain will be $K_p = 7.6$. However, we should confirm that this gain will satisfy all other design requirements.

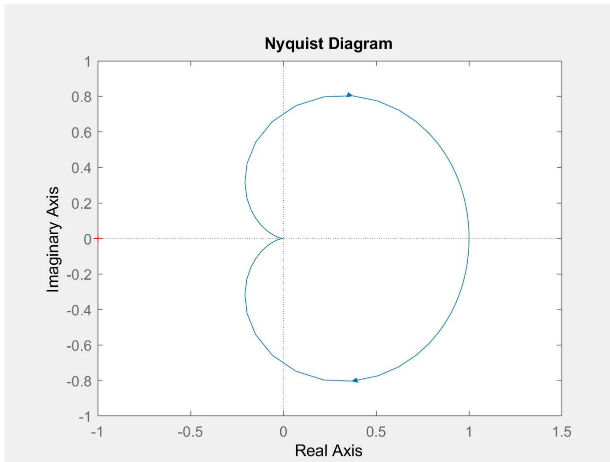
When $K_p = 7.6$, the rise time is 0.22 s which is a 56% improvement on the critically damped rise time. The settling time is 0.62 s, which is a 28% improvement on the critically damped settling time. As stated previously, the gain $K_p = 7.6$ is designed so that $\zeta = \sqrt{1/2}$, which satisfies the requirement for damping. Therefore, the proposed controller meets the requirements for quickness of response and damping.

To consider frequency response specifications, examine the system's Bode plot.



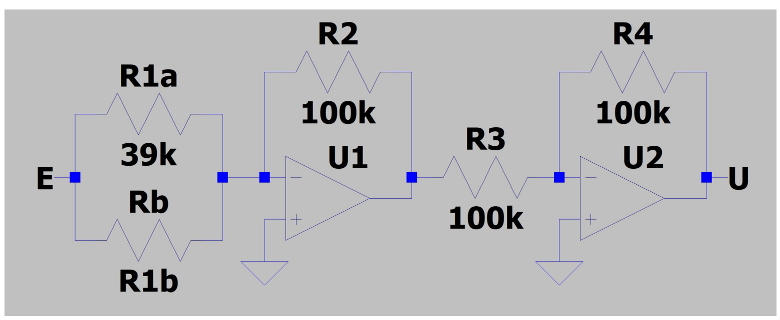
The system has a -3 dB frequency of 9.56 rad/s, which meets the design requirement of $\omega_{bw} \geq 6.8$ rad/s. The Bode plot shows a completely flat passband with no peaks above 0 dB, so the specification that there be no resonance is satisfied. Therefore, the system meets the frequency response requirements.

To consider the stability robustness requirements, examine the Nyquist plot of $L(s) = G_{ol}H = \frac{0.89K_p}{\tau_m s^2+s}$.



The Nyquist plot intersects the real axis at the origin, so the gain margin is infinite. MATLAB's control system toolbox calculator states that the phase margin is 176°, which far exceeds the minimum required value of 30°. Therefore, the stability margin requirements are met.

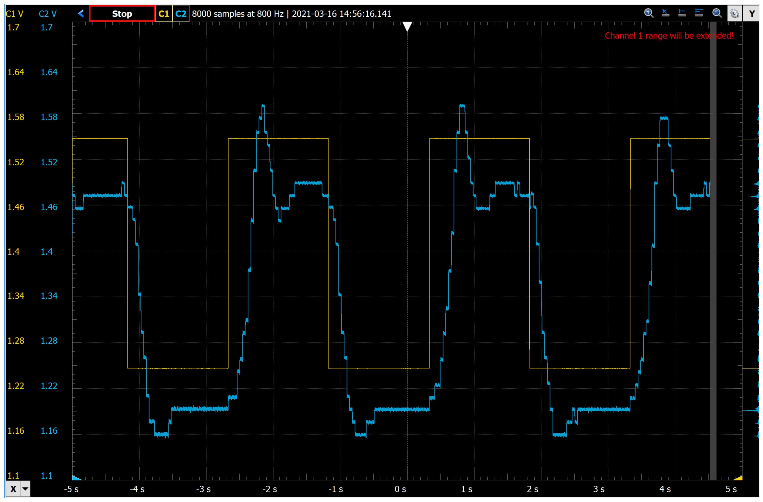
The compensator design adheres to all system specifications, so $H(s) = 7.6$ is an acceptable final design. In order to implement this controller, the op amp circuit below is used.



The first op amp stage provides a gain of -7.56 and the second stage a gain of -1.

VI Experimental Results

To experimentally test the system, I provided a reference square wave which went between 1.25 V and 1.55 with a period of 3 s. Given the sensor's calibration, this should make the car step between 6 and 8 inches. The measured results are shown below. The yellow signal is the input reference and the blue signal is the output of the optical sensor.



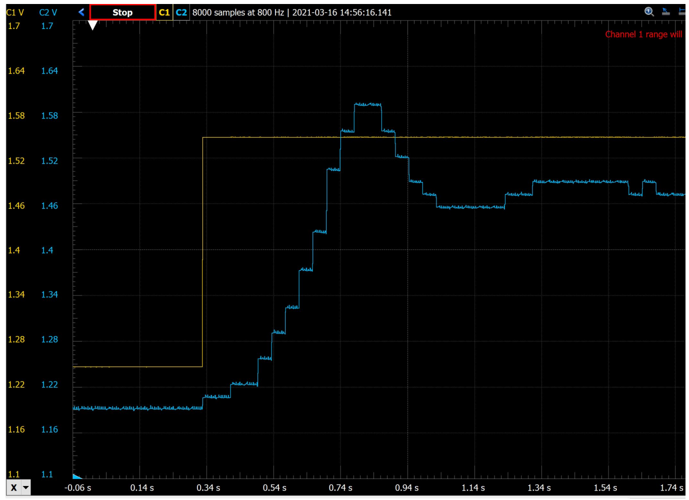
One of the first observations I make when observing this plot is that there is some steady state error while none is expected. Observe, though, that in both cases, the error is the roughly the same magnitude, and it is below the expected voltage. Therefore, it is not caused by a flaw in the controller or system modeling as much as it is controlled by some offset built into the system. This could be caused by a number of things.

One explanation is that we may have slightly mismatched resistors in our summing junction. If the component tolerance is large enough, it no longer behaves like an ideal summing junction, and we could have introduced a small offset.

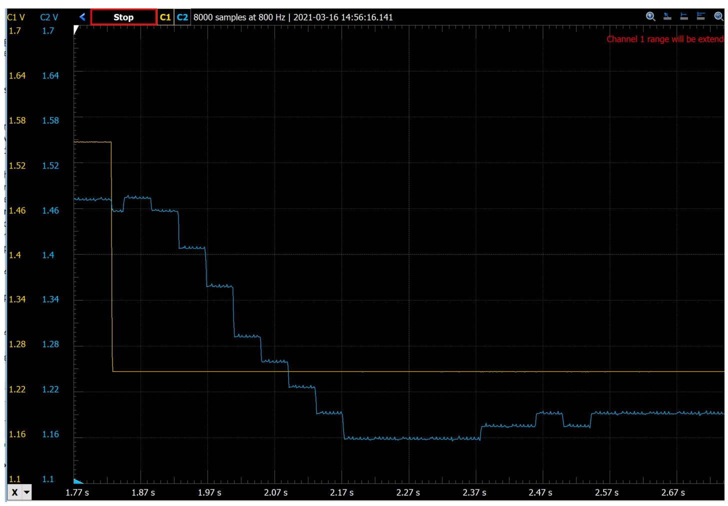
Another possibility is that "stiction" occurs once the sensor signal has settled within some amount of the reference. This occurs because even after it is amplified by the controller, the small difference between reference and output signal is not enough to actually make the motor turn at all. I think this is the more likely cause for the offset we see.

Another interesting observation is the lack of symmetry in the sensor signal. The cause for this is the sensor nonlinearity. When the reference signal goes high, the car moves towards the target, which causes its effective gain to increase. Conversely, when the reference goes low, the car moves away from the wall, causing the sensor gain to decrease. This explains why we see much more overshoot and ringing in the rising step but see a very damped response in the falling step.

It is not easy to determine exact time domain response parameters from the plot but they can be estimated fairly easy. First examine a close-up of the rising step.



This response shows a rise time of approximately 0.35 s and a settling time of approximately 0.9 s. Next examine the falling step.



The estimated value for the rise time is 0.3 s and the estimated settling time is 0.5 s.

Between the rising and falling step, the average measure rise time is about 325 ms and the average measured settling time is about 700 ms. The predicted values were a 220 ms rise time and a 620 ms settling time. The measured results match the theoretical results fairly well, though each is about 100 ms slower. This is likely caused by second and third order effects we know the motor experiences, preventing it from accelerating as fast as might otherwise be expected.

The experimental rise and settling times are faster than the theoretical rise and settling times for the critically damped system, though they are not 20% faster, which was the design requirement I hoped to meet. However, given the second and third order effects of the motor which are hard to account for, seeing any improvement at all relative to the critically damped system is perfectly satisfactory.

Overall, the experimental results match the simulated results very well, given the model of the system we have used. What slight deviations from simulation did exist all had fairly simple explanations.

VII Conclusion

While the design criteria for step response were technically not met by the empirical results, I still believe the result to be satisfactory. There were a large number of nonidealities which were very hard to model into the system (discrete sensor outputs, nonlinear sensor gain, motor ”stiction”, second order motor effects), so getting as close as I did to the theoretical results should be considered a success.

Initially, I devoted large amounts of time to designing and implementing a PD controller which had much better theoretical performance characteristics than the P controller I ended up using. However, once I discovered the discrete nature of the sensor output, I was forced to abandon this idea, and settle for the simpler proportional controller.

This taught me that in engineering design problems, it is often best to start testing earlier rather than later, when possible. If I had used tested some arbitrary PD controller early on, I would have discovered it was a design destined to fail and not wasted many hours tuning its gains.

This project, and this class in general, also taught me another valuable lesson about design in engineering: optimization is not always possible, and even if it is, it is often not worth the extra effort. If I am able to use simplifications and estimations to fairly easily design a system that is within 5% of some ideal value, it is probably not worth spending days trying to solve a high dimensional optimization problem to get within 2% percent of that value. The importance of effectively budgeting time cannot be understated.

Overall, this project allowed me to learn new strategies for modeling systems and understanding their nonidealities as well as well as valuable design process strategies that I will take with me into my future engineering career.