Lab Assignment 3: DC Motor Servo System

Team members:
Shiva Ghose, @gshiva
John Peterson, @jrpeters
Peter Turpel, @pturpel
Chan-Rong Lin, @pmelin

Teamwork Participation Pledge :: Team 1

I attest that I have made a fair and equitable contribution to this lab and submitted assignment.

My signature also indicates that I have followed the University of Michigan Honor Code, while working on this lab and assignment.

I accept my responsibility to look after all of the equipment assigned to me and my team, and that I have read and understood the X50 Lab Rules.

Name	Email	Signature
Shiva Ghose	gshiva@umich.edu	
John Peterson	jrpeters@umich.edu	
Peter Turpel	pturpel@umich.edu	
Chan-Rong Lin	pmelin@umich.edu	

1.

a.

U1

U1 serves as a difference circuit between ^{-}Ref and ^{+}Ref with a variable overall gain dictated by the potentiometer.

$$V_{out} = \left(\frac{R_6}{R_5 + R_6}\right)\left(\left(1 + \frac{R_2}{R_1}\right)\left(\frac{R_4}{R_3 + R_4}\right)(^{-}Ref) - \left(\frac{R_2}{R_1}\right)(^{+}Ref)\right)$$

$$V_{out} = \left(\frac{R_6}{R_5 + R_6}\right)\left(\left(1 + \frac{20}{50}\right)\left(\frac{20}{50 + 20}\right)(^{-}Ref) - \left(\frac{20}{50}\right)(^{+}Ref)\right)$$

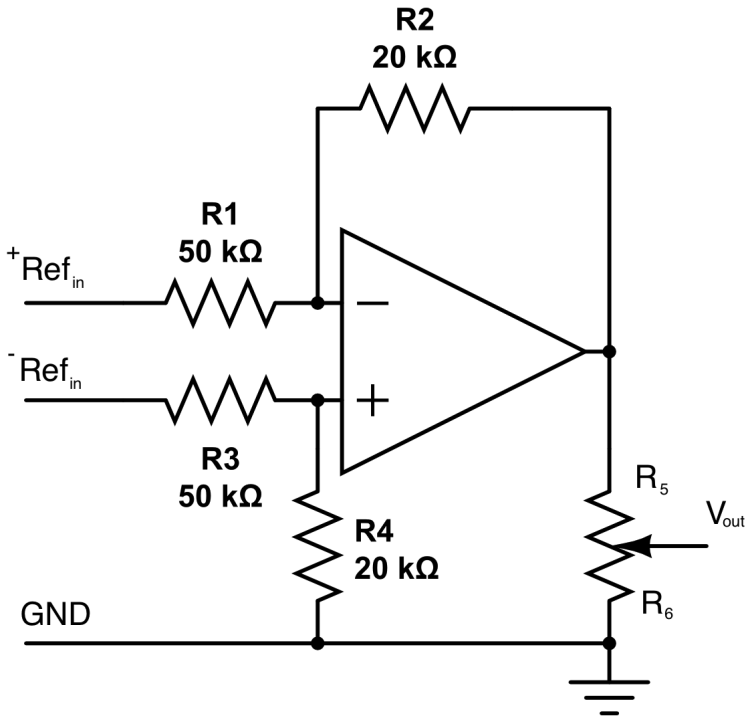

Figure 1: Unit 1 within AMC servo-amplifier

U3

U3 serves as an inverting amplifier which can be tuned by the potentiometer.

$$U3_{out} = -\left(\frac{R_3 + R_4}{R_4}\right)\left(\frac{R_2}{R_1}\right) Ref_{gain}$$

$$U3_{out} = -\left(\frac{R_3 + R_4}{R_4}\right)\left(\frac{10}{5}\right) Ref_{gain}$$

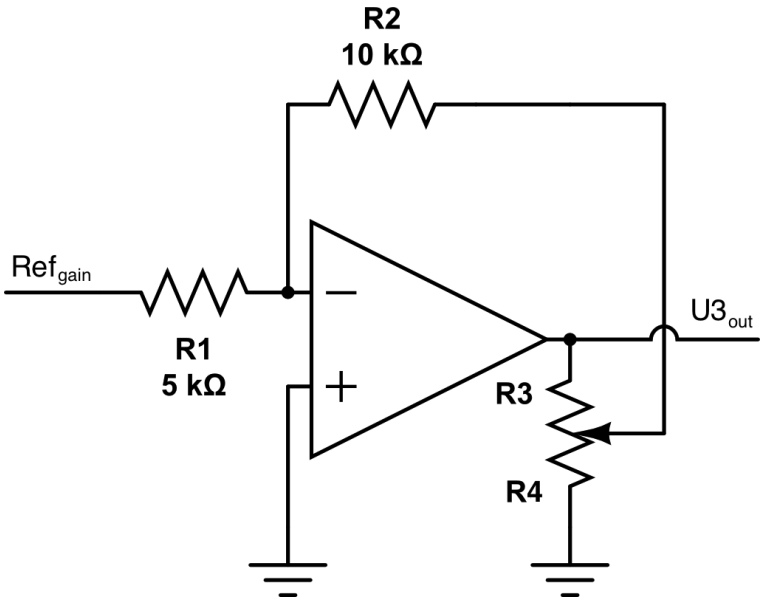

Figure 2: Unit 3 within AMC servo-amplifier

U4

U4 serves as an inverting integrator with a proportional gain when switch 3 is open, and a simple inverting amplifier with the switch closed.

$$U4_{out} = -\left(\frac{1}{R_1}\right)\left(R_2 + \frac{C_1}{s}\right)U3_{out}$$

$$U4_{out} = -\left(\frac{1}{500}\right)\left(500 + \frac{0.01}{s}\right)U3_{out}$$

b.

Current Limiting Resistor. For DC current, an inductor is effectively a short circuit. We need a resistor to limit the maximum current that goes to ground. confirm with him that its a coil in series with resistor

c.

The motor data sheet specifies a max winding temperature of 155 °C; this is the temperature at which the motor can become irreparably damaged. The data sheet also gives a thermal impedance of 11.2 °C/W. Assuming a room temperature of 15 °C, the motor's temperature can only increase by 130 °C before overheating. With these values (and the motor's internal resistance of 1.85 Ω) we can solve for the maximum continuous current:

$$130 = 11.2 * i^2_{continuous} * 1.85$$

$$i_{continuous} = 2.5048\,A$$

With this current and the given motor constant we can then calculate the maximum continuous torque:

$$T_m = k_t * i_{continuous} = 4.24 \times 10^{-2} * 2.5048 = 10.6 \times 10^{-2}\,Nm$$

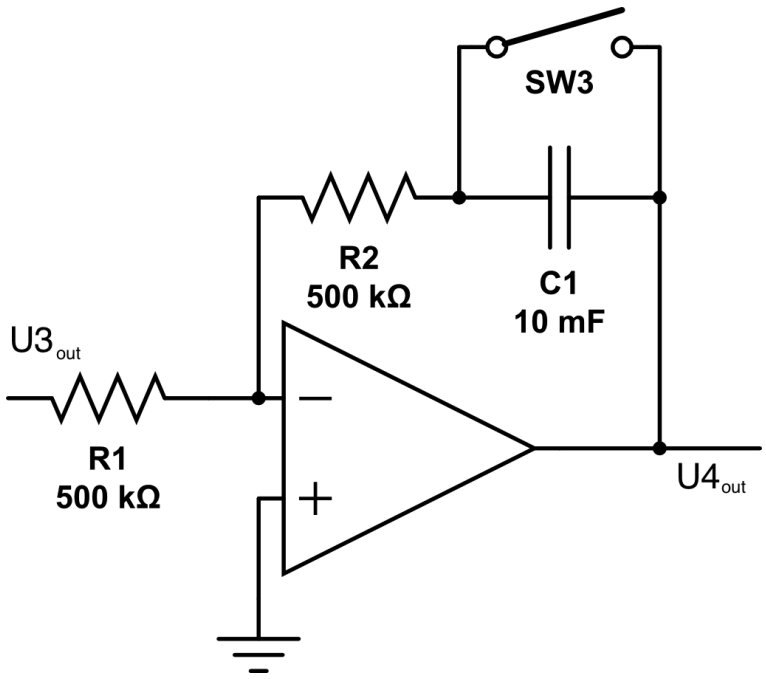

Figure 3: Unit 4 within AMC servo-amplifier

This value is more than the torque value given in the data sheet, implying that our calculated current value is less conservative than the ratings in the data sheet. Based on the given rating for max continuous torque (8.1×10^{-2} Nm), the max continuous current would be 1.91 A, but at that value the temperature rise would be 75.6 °C above ambient. This means that at the max torque given by the data sheet, the motor would be well under the temperature limit. It's possible that the given value is either based on a safety factor to prevent overheating, or there are other modes of breakdown that can result from higher currents. If the lower limit is based on temperature, then we should be able to use slightly higher currents given that we are running for short periods.

The motor's data sheet lists a peak current rating of 13.0 A and a stall torque of 0.54 Nm. With this torque value and the given torque constant (4.24×10^{-2} Nm/A), the peak current should be:

$$i_{peak} = \frac{T_{stall}}{k_t} = \frac{0.54}{4.24 \times 10^{-2}} = 12.74\,A$$

This is less than the given rating for peak current. Choosing this as our peak current limit, the power in the motor is:

$$P = i_{peak}^2 * R_m = 12.74^2 * 1.85 = 300.269\,W$$

Because this is occurring at stall, there is no mechanical power output from the motor, and all of the power goes to heat. If the system were allowed to go to steady-state with no failures, this would lead to a temperature of:

$$Temp = P * R_t = 300.269 * 11.2 = 3363.0128\,^\circ C\ (above\ ambient)$$

Clearly the system could not physical reach this temperature without failing, but it illustrates the reason that peak current can only be sustained for short times. With the given thermal time constant (13.8 min), we can plot the temperature rise of the motor, as shown in figure 4, and estimate that it would overheat after approximately 32.4 seconds at peak current.

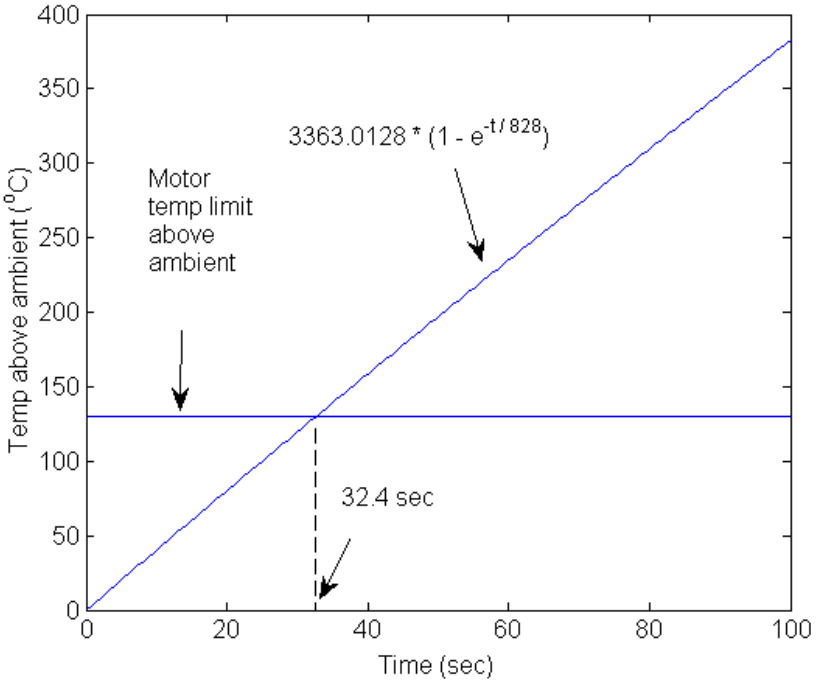

Figure 4: Motor temperature over time at peak current

d.

The 12A8E servo amplifier is capable of outputting maximum continuous and peak currents of +-6A and +-12A, respectively (from the data sheet). However, the servo amplifier includes a potentiometer that can be used to reduce both the continuous and peak currents (while maintaining a 1:2 relationship between them). The lab instructions specify that the potentiometer should be adjusted so that the current limits will be +-3A and +-6A. Additionally, the continuous current limit can be further reduced (without affecting the peak current) by connecting a current limiting resistor between pins P1-10 and P1-2 on the servo-amp. That was not done in this lab. Assuming the adjustment on the current limiting potentiometer was done correctly, the servo-amp should be capable of sending +-3A continuously and +-6A intermittently to the motor. The continuous current limit from the servo-amp is greater than the continuous current rating of the motor, so we can't rely on the servo-amp to prevent the motor from drawing too much current.

The data sheet for the servo-amp specifies a peak current of +-12 A for a maximum of 2 seconds. From information on the Advanced Motion Controls website, this can only be achieved if the current switches across its entire range (-12 A to +12 A, and vice-versa). Any change of smaller magnitude will not be maintained as long (i.e. switching from 0 A to +12 A will only be held for 1 second before the current begins falling). A primary concern with maintaining high currents is overheating, and the servo-amp has a much lower temperature limit than the motor (65 °C vs 155 °C). This is why the peak current can not be maintained for long, the maximum temperature would be exceeded very quickly and the servo-amp's internal safeties would shut it off. As an example, in the thermal model for the motor at peak current (which is lower than the 6 A rating of the servo-amp), 65 °C was reached in only 9.8 seconds. Of course, the servo-amp has different thermal properties than the motor, so that specific value is not accurate. However, the basic idea is valid: the servo-amp is capable of passing a lot of current, and has a lower temperature limit than the motor, so there is a danger of it overheating if high currents are maintained. Luckily, the servo-amp has

built in safeties, such as a thermal shut-off.

e.

In the above sections, we showed that the motor has a continuous and peak current limits of 2.5 A and 12.74 A, and the servo-amp has current limits of 3 A and 6 A, respectively. Within our LabView controller we chose to implement saturation limits of +-6 A so that we could take advantage of the peak current capability of the servo-amp and motor. We considered setting a saturation limit based on the motor's continuous current limit of 2.5 A, but felt that might have a negative effect on the transient response of the motor. Also, we knew that the servo-amp has built in mechanisms to prevent the current from remaining at the peak value - it automatically lowers it after at most two seconds. Additionally, as we were tuning our LabView controller, we monitored a waveform graph of the attempted output of the contoller. This allowed us to adjust our gain values not just for system stability, but also to keep the attempted output within safe limits. Given more resources, we could have implemented more safety measures. For example, we could implement a slow-blow fuse rated a little under the continuous current limit of the motor. This would protect the system from high currents for prolonged times, but still allow for short bursts of higher current during transient conditions.

2.

a.

Using expressions for the back-emf of the motor with a simple model of the motor as a resistance in series with an inductance we can model the electrical characteristics of the motor. We can combine this with a simple lumped parameter model of physical dynamics of the motor to derive an overall transfer function.

$$V_s - e = iR + L\left(\frac{di}{dt}\right)$$

$$e = K_b\dot{\theta}$$

$$\Sigma_\tau = \tau_{motor} - \tau_{damping} + \tau_{coulomb} = J\ddot{\theta}$$

$$\tau_{motor} = K_T i \qquad \tau_{damping} = b\dot{\theta}$$

$$\tau_{coulomb} = \begin{cases} -\tau_{motor} & \text{if } \dot{\theta} = 0 \text{ and } |\tau_{motor}| < \tau_{friction} \\ -sgn(\tau_{motor}) \cdot \tau_{friction} & \text{if } \dot{\theta} = 0 \text{ and } |\tau_{motor}| > \tau_{friction} \\ -sgn(\dot{\theta}) \cdot \tau_{friction} & \text{if } \dot{\theta} \neq 0 \end{cases}$$

Where: K_t is the motor torque constant; J is the rotor inertia; b is the viscous damping on the rotor; K_b is the back-emf constant of the motor; R is the equivalent resistance of the motor; L is the effective motor inductance; i is the current flowing through the motor; and $\tau_{coulomb}$ is the force of coulomb friction on the rotor.

Neglecting coulomb friction:

$$V_s(s) - s\,K_b\,\theta(s) = I(s)(R + sL) \qquad \text{and} \qquad K_t\,I(s) = \theta(s)(Js^2 + bs)$$

$$\frac{\theta(s)}{V(s)} = \frac{K_T}{s\left[(Js + b)(Ls + R) + K_b K_T\right]}$$

The transfer function relating voltage and position is not particularly convenient but we can obtain a transfer function easily in terms of $I(s)$.

$$K_t\,I(s) = \theta(s)(Js^2 + bs)$$

$$\frac{\theta(s)}{I(s)} = \frac{K_T}{(Js + b)s}$$

This model relating angular position, $\theta(t)$ with motor current $i(t)$ requires the moment of inertia of the rotor, J, the viscous damping coefficient, b, and the torque constant K_T. Note that coulomb friction has been neglected.

The driver gives us current control as long as we are opperating away from the maximum speed of the motor. At maximum speed, the current required by the motor should drop to some steady state value required to overcome losses within the motor.

may need to redo picture a bit to agree with terminology in equation

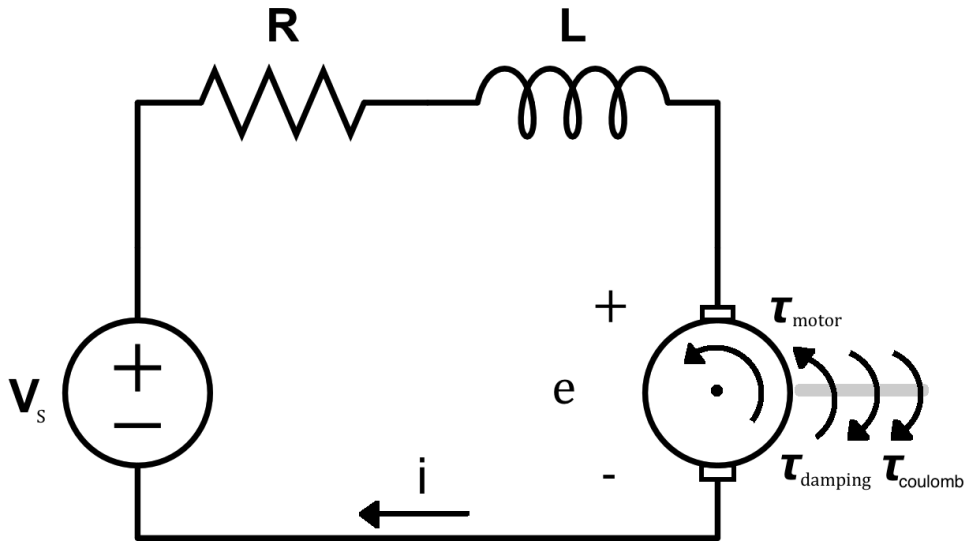

Figure 5: DC motor model

b.

The torque of friction and the viscous damping can be jointly derived in an experiment where for a given constant input voltage to the driver, V_{driver}, we read off the angular velocity. At a constant velocity the following equation holds giving a relationship between τ_{motor}, b, and $\tau_{coulomb}$.

$$\Sigma_\tau = \tau_{motor} - b\dot{\theta} + \tau_{coulomb} = J\ddot{\theta} \qquad \ddot{\theta} = 0$$

$$\tau_{motor} = K_T I_m = b\dot{\theta}_{ss} - \tau_{coulomb}$$

Assuming that we have the motor torque constant K_t, we can write an equation that relates steady state angular velocity $\theta_{steadystate}$ to motor current which is known to be directly proportional to V_{driver} for suitably small values of V_{driver}.

$$V_{driver} = I_m \qquad V_{driver} = \left(\frac{b}{K_T}\right)\dot{\theta}_{ss} - \frac{\tau_{coulomb}}{K_T}$$

From our plot in figure 6 we can obtain $({}^{b}/_{K_t})$ as the slope of each of the linear segments and ${}^{\tau_{coulomb}}/_{K_T}$ as the y intercepts shown in table 1. We can separately determine the force of static friction $\tau_{static\,coulomb}$ by measuring the smallest input voltage to the driver, and from this how much torque, is required to start the motor from rest.

$$\Sigma_\tau = \tau_{motor} - b\dot{\theta} + \tau_{coulomb} = J\ddot{\theta} \qquad \dot{\theta} = \ddot{\theta} = 0$$

$$V_{driver} = -\frac{\tau_{coulomb}}{K_T}$$

	Slope $\left(\frac{V*s}{rad}\right)$	y-intercept (V)
Forwards	5.4289e-04	0.46374
Reverse	6.72805e-04	-0.60717

Table 1: Extracted slopes and y-intercepts from current speed curve

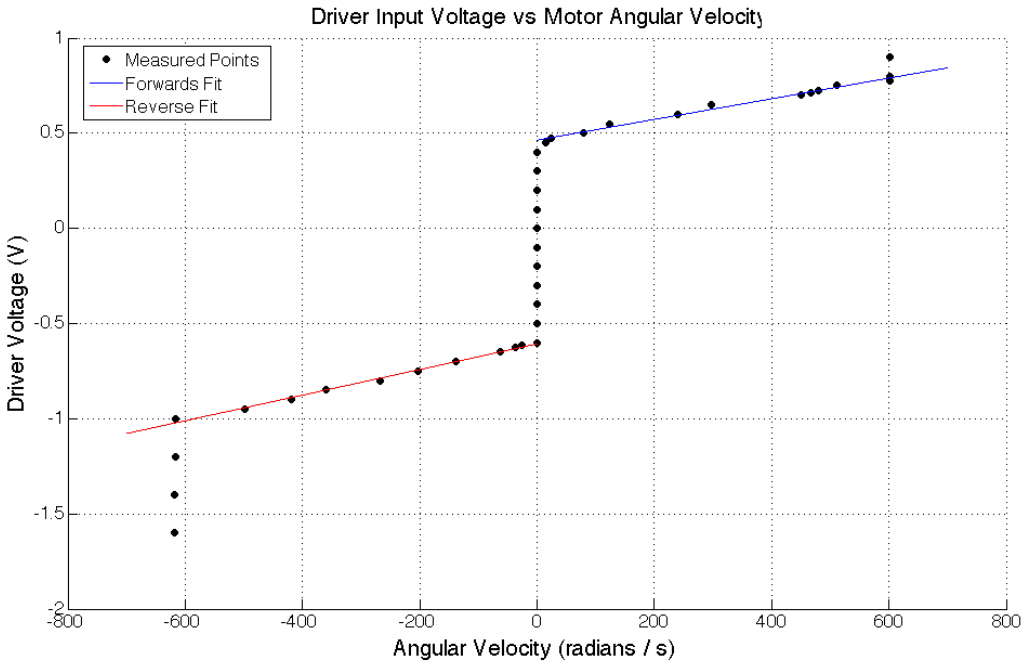

Figure 6: Relationship between angular velocity and steady state voltage applied to motor driver

	Data Sheet Values	Average Estimated Values
$b\ (N \cdot s)$	$3.7 \cdot 10^{-6}$	$2.5773 \cdot 10^{-5}$
$\tau_{coulomb}\ (N \cdot m)$	$5.6 \cdot 10^{-3}$	0.044978

Table 2: Comparison of viscous damping and friction torque estimates with values obtained from the data sheet

If we assume the motor constant on the data-sheet of $0.0424\,{}^{N \cdot m}/_{A}$ then we can arrive at the following estimates for b and $\tau_{coulomb}$ which are shown in 2 compared against the values obtained from the data sheet. Our values are within an order of magnitude of the provided values.

To compute the torque constant, K_T we can first compute the back-emf constant, K_b, which we know to be numerically equal to K_T in SI units. To calculate K_b we can measure the internal resistance of the motor, R, using an ohm-meter. Then running the system at a constant current, we can measure the source voltage, V_s that the driver outputs to maintain that current to obtain e. We can use the encoder to measure the angular velocity and use that value with e to solve for K_b:

$$\Sigma_V = V_s + V_R + V_L e = V_s + IR + e = 0$$

$$K_b = \frac{e}{\dot{\theta}}$$

We can experimentally determine the viscous damping term, b by applying a known torque to the motor. Once the angular velocity has reached steady state, b can be read off directly. The driver in current mode delivers a known current, which using the torque constant determined above yields the torque.

$$\Sigma_{tau} = \tau_{motor} - b\dot{\theta} + \tau_{coulomb} = J\ddot{\theta} \qquad \ddot{\theta} = 0$$

$$b = \frac{\tau_{motor} + \tau_{coulomb}}{\dot{\theta}} \approx \frac{\tau_{motor}}{\dot{\theta}}$$

We can determine the moment of inertia of the rotor, J by starting from rest and sending the motor a known torque while recording the position as a function of time stopping when the system has reached a constant velocity. Then by computing the 1^{st} and 2^{nd} order derivatives at each time step we can assemble a linear system to allow us to jointly solve for J, b, and $\tau_{coulomb}$.

$$A_t = \begin{bmatrix} \ddot{\theta}_t & \dot{\theta}_t & -1 \end{bmatrix} \qquad x = \begin{bmatrix} J \\ b \\ \tau_{coulomb} \end{bmatrix} \qquad z_t = \tau_{motor}$$

$$Ax = z$$

Rather than attempt the method above, we instead used MATLAB's numerical differential equation solving to simulate the trajectory of the system as a function of time, driver current, I_t and our unknown system parameters J, b, K_t, and $\tau_{coulomb}$. By then comparing the predicted angular positions with the measured angular positions over time, we can construct a non-linear regression to estimate the unknown parameters. We collected a data set where the driver was given a sinusoidal voltage, and we assume that motor current, I_m, is identical to this driver voltage and recorded the motor motion that this input caused.

Table 3 below shows the parameters for our equation of motion obtained from the data sheet. The results, in figure 7 shows the estimated trajectory using these parameter values. It is very clear that these parameter values are quite far off. Attempting to perform nonlinear optimization on our model parameters proved to be very difficult, so we resorted to parameter sweeping to to identify appropriate values. Figure 8 shows one of the best results discovered by this parameter sweep. In this case the error has been reduced, but the fit still appear inappropriate.

	Data Sheet Parameters
$J \ \left(\frac{N \cdot s^2}{m}\right)$	$8.5 \cdot 10^{-6}$
$b \ (N \cdot s)$	$3.7 \cdot 10^{-6}$
$\tau_{coulomb} \ (N \cdot m)$	$5.6 \cdot 10^{-3}$
$K_T \ \left(\frac{N \cdot m}{A}\right)$	$4.24 \cdot 10^{-2}$

Table 3: Original model parameter values obtained from the data sheet.

Manual parameter guessing revealed that it was not possible to obtain the measured curve using the model, repeated below.

$$\Sigma_{tau} = K_T I_m - b\dot{\theta} + \tau_{coulomb} = J\ddot{\theta}$$

Clearly there were some effects that this model neglected. Our previous experiment to attempt to determine b and $\tau_{coulomb}$ showed that these parameters are likely not the same in each direction, so we augmented the model to allow b and $\tau_{coulomb}$ to take on different values in the forwards direction and reverse directions yielding the following model:

$$\Sigma_{tau} = \tau_{motor} - b\dot{\theta} + \tau_{coulomb} = J\ddot{\theta}$$

$$\tau_{coulomb} = \begin{cases} -\tau_{forwards} & \text{if } \dot{\theta} > 0 \\ +\tau_{reverse} & \text{if } \dot{\theta} < 0 \\ -\tau_{forwards} & \text{if } \dot{\theta} = 0 \text{ and } \tau_{motor} > \tau_{forwards} \\ -\tau_{motor} & \text{if } \dot{\theta} = 0 \text{ and } 0 <= \tau_{motor} <= \tau_{forwards} \\ +\tau_{reverse} & \text{if } \dot{\theta} = 0 \text{ and } \tau_{motor} < -\tau_{reverse} \\ +\tau_{motor} & \text{if } \dot{\theta} = 0 \text{ and } 0 >= \tau_{motor} >= -\tau_{reverse} \end{cases}$$

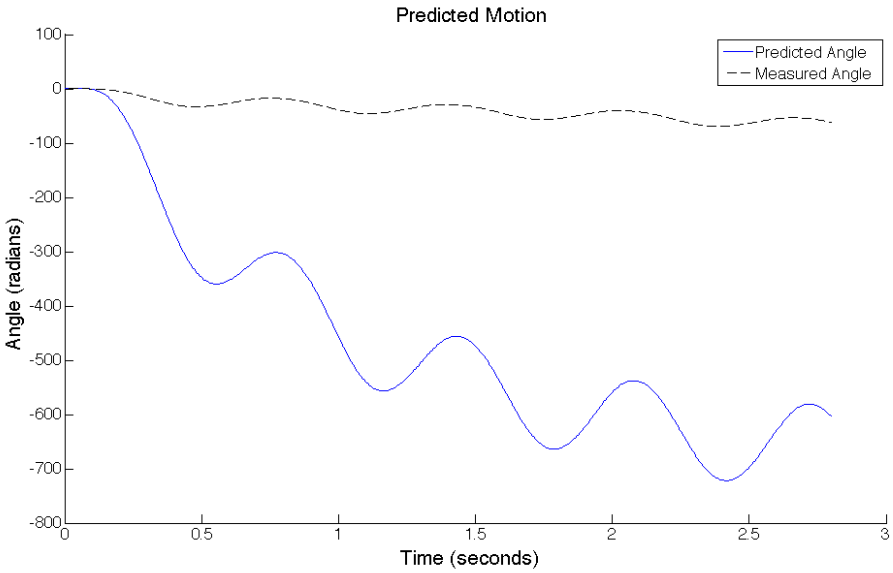

Figure 7: Predicted behavior vs Actual behavior using initial parameter values.

$$b = \begin{cases} b_{forwards} & \text{if } \dot{\theta} > 0 \\ b_{reverse} & \text{if } \dot{\theta} < 0 \end{cases}$$

With this improved model and some intelligent guessing for new parameter values, we arrived at an excellent fit shown in figure 9. The most notable feature of these values is the large difference between the torque of friction in the forwards and reverse directions.

	Data-Sheet Values	Fit Values
$J\ \left(\frac{N \cdot s^2}{m}\right)$	$8.5 \cdot 10^{-6}$	$3.50514 \cdot 10^{-5}$
$K_t\ \left(\frac{N \cdot m}{A}\right)$	0.0424	0.0314499
$b_{forwards}\ (N \cdot s)$	$3.7 \cdot 10^{-6}$	$1.08586 \cdot 10^{-5}$
$b_{reverse}\ (N \cdot s)$	$3.7 \cdot 10^{-6}$	$2.49301 \cdot 10^{-5}$
$\tau_{forwards}\ (N \cdot m)$	$5.6 \cdot 10^{-3}$	0.0224389
$\tau_{reverse}\ (N \cdot m)$	$5.6 \cdot 10^{-3}$	0.0143096

Table 4: Final Parameter Values vs Original Data Sheet Values

Table 3 below shows our final fit values alongside the original values obtained from the data sheet. This large difference in values is clearly illustrated by the large difference in behavior between the actual system and our model using the values obtained from the data sheet shown in figure 7. The results of fitting are shown in figure 9. We can also compare these fit values with values obtained from our earlier experiment. We can notice that our terms are in the same order of magnitude, however it seems as if this experiment did not accurately estimate the force of friction as the relative values seem to be flipped between the two experiments.

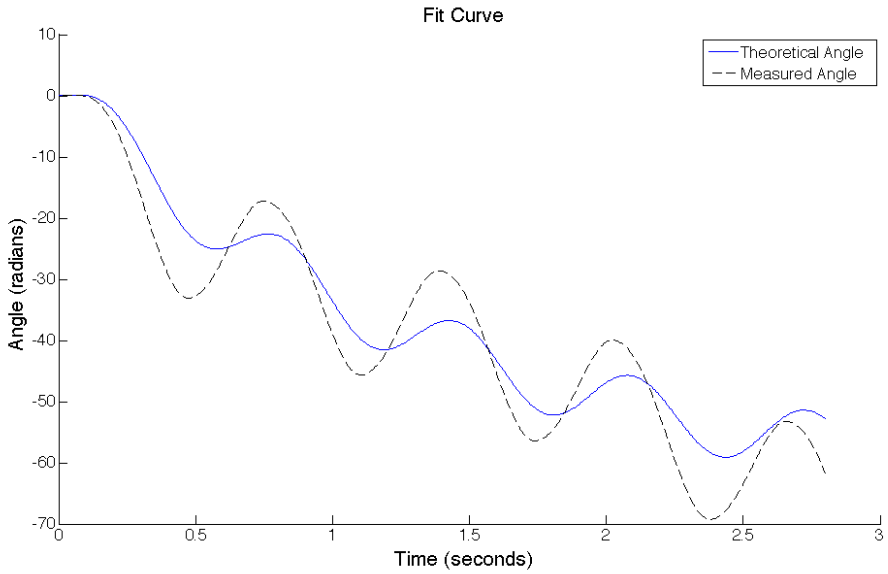

Figure 8: Improved Fit $J = 6.06724 \cdot 10^{-5}$ $B = 4.05562 \cdot 10^{-5}$ $F_f = 1.45312 \cdot 10^{-4}$ $K_t = 0.019020$

	Predicted Value	Fit Value
$B_{forwards}$	$1.707338 \cdot 10^{-5}$	$1.08586 \cdot 10^{-5}$
$B_{reverse}$	$2.11611 \cdot 10^{-5}$	$2.49301 \cdot 10^{-5}$
$\tau_{forwards}$	0.0145846	0.0224389
$\tau_{reverse}$	0.0190954	0.0143096

Table 5: Damping and friction torques calculated from earlier experiment vs fit values

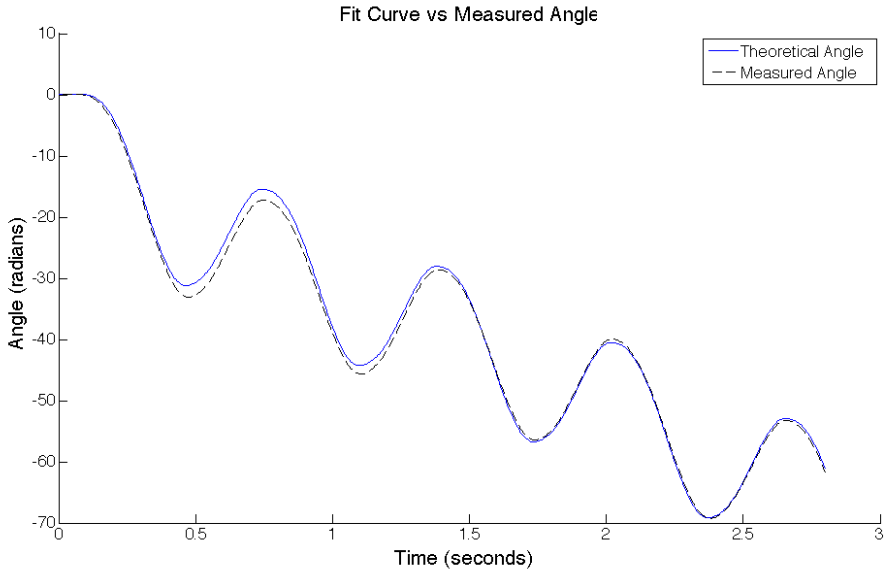

Figure 9: Predicted behavior vs Actual behavior using fit parameter values and enhanced modeling

3.

a.

b.

c.

d.

e.

f.

4.

a.

When we implemented the controller and values from our simulation in part 3, we were unable to achieve stable output in the physical system. Therefore, we kept the same general approach - a velocity controller nested within a position controller with feed forward - but we re-tuned the system with different gain values. Because we have a velocity controller inside of our position controller, we first tuned the velocity control on its own (see section 6). Once it was working, we then implemented it in the inner loop of our position controller and tuned the position control gains. In the end we were not able to meet all of the control goals with one controller. We designed one controller with good step response and noise attenuation, and then modified the gain values to improve the frequency response separately. The results for the step response and noise attenuation are given below.

Gains	P	I	D	Feed Forward	Filter τ
Position Loop	0.1	20	0	1	6.5797
Velocity Loop	22.5	22.5	1.125	-	6.5797

Table 6: Values of position controller with nested velocity controller for step response and noise attenuation

Figure 10 shows the system's response to a position step input of π radians, and figure 11 shows a zoomed in view after it had settled. The peak error here is 0.11%, which greatly exceeds the goal of 2% error. Additional results for position steps of different magnitudes are included in section 5b.

Compare to simulation results

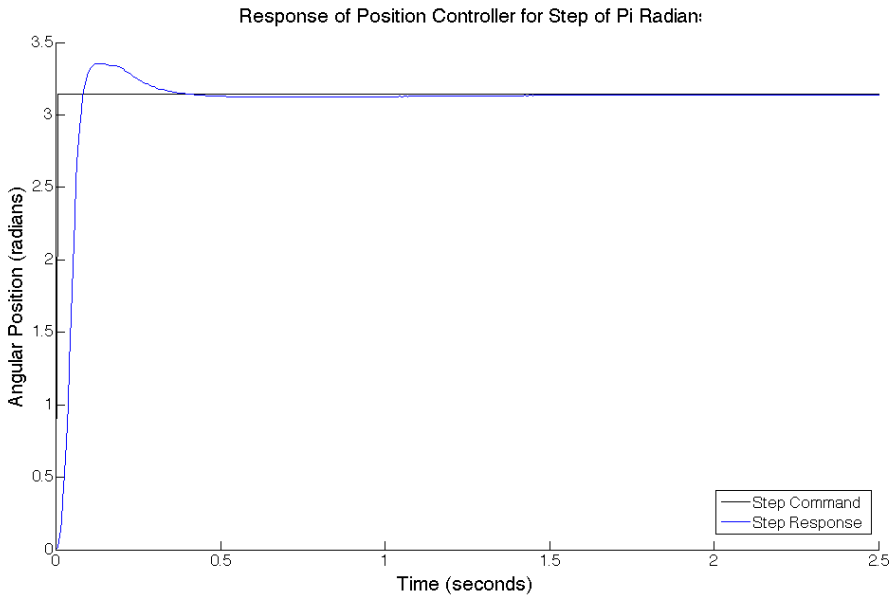

Figure 10: Step response of position controller

Figure 12 shows the noise attenuation during a step input of π radians. To simulate the noise within LabView, we added a 60 Hz sine signal with an amplitude of 1 to the output of the DAQ assistant. Figure

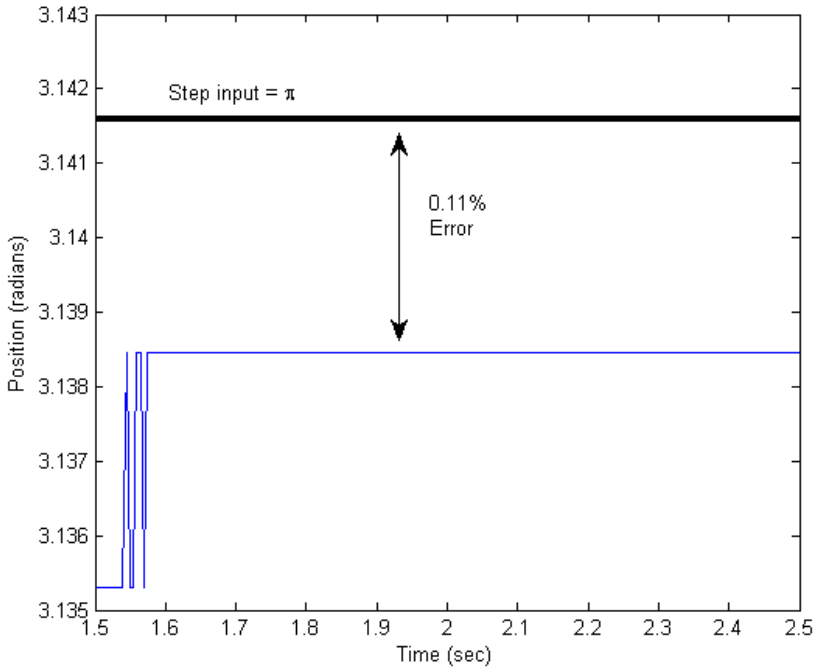

Figure 11: Steady state error for position step input of π

13 shows a zoomed in view after the step has taken place. For the step response with no noise, our steady state error was approximately 0.003, and with the noise the highest error is 0.060. Given that we were inputting a noise amplitude of one and the error only increased by 0.057, we have attenuated the noise by approximately 17.54 times. This exceeds the goal of a ten times reduction.

Compare to simulation results

The position controller we were using for the step response and noise attenuation did not not perform well with sinusoidal inputs, therefore we modified to gains while keeping the overall architecture the same. The relevant values are given in the table 7, and the following figures show the response at various frequencies with an amplitude of π. Figure 14 shows the error as a function of input frequency. During our tests the position output remained stable up to 5 Hz, but the output amplitude error exceeded 5% just beyond 4 Hz. This does not meet the goal of less than 5% error up to 5 Hz.

Compare to simulation

Gains	P	I	D	Feed Forward	Filter τ
Position Loop	0	20	0	1	6.5797
Velocity Loop	3	7	0.4	-	6.5797

Table 7: Values of position controller with nested velocity controller for frequency response

To determine the closed-loop bandwidth of the system, we slowly increased the frequency of the input signal until the output amplitude just crossed -3 dB (for a input of π this corresponded to an output of 2.22). We found that the bandwidth of our position control system was 5.27 Hz, as shown in figure 15.

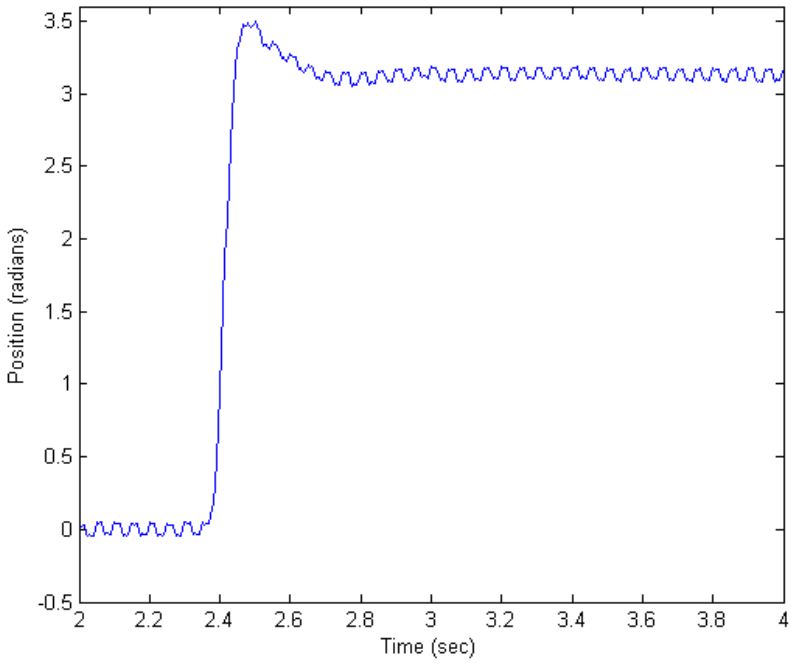

Figure 12: Step response for input of π with a 60 Hz noise signal (amplitude = 1)

Frequency (Hz)	Input Amplitude	Output Amplitude	Magnitude (dB)	%Error
1	π	3.13845	-0.008693172	-0.100033771
2	π	3.02221	-0.336504689	-3.800067888
3	π	3.079209	-0.174214105	-1.985733367
4	π	3.030152	-0.313709167	-3.547266176
5	π	2.473084	-2.078220099	-21.27929134
5.27	π	2.181877	-3.166392169	-30.54869805

Table 8: Velocity control data.

Even though we included many nonlinear factors in our simulation, we still faced a lot of trouble meeting all of the performance specifications, particularly trying to do so with just one controller. In general, we found that when we adjusted gains to meet the step requirement, the frequency response usually got worse, and vice-versa. It's possible that some of the values used in our simulation are incorrect (even after testing and regression), or that there are other factors we did not account for, such as the dynamics of the servo-amp. In the end we were able to meet the steady state error for step input and noise attenuation requirements, and were very close to the frequency response requirement.

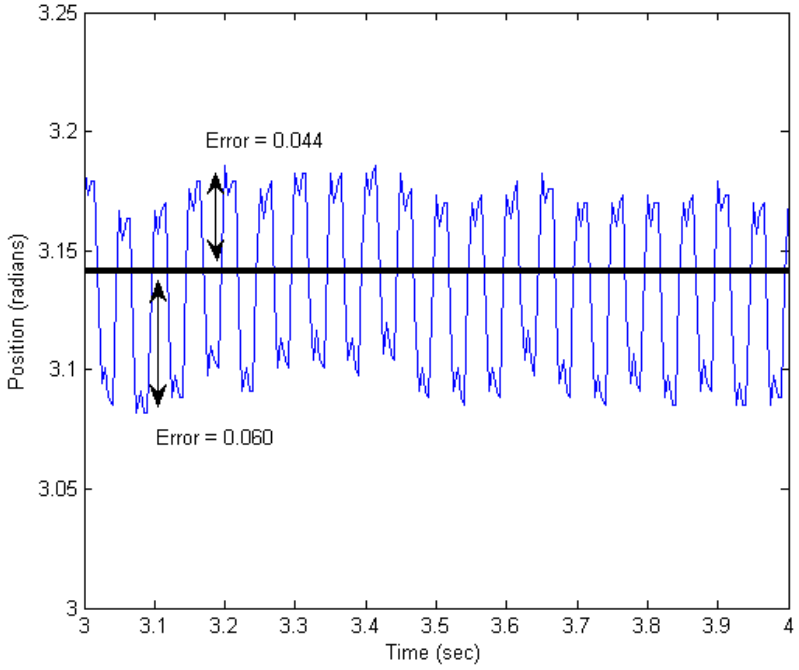

Figure 13: Steady state step response for input of π with a 60 Hz noise signal (amplitude = 1)

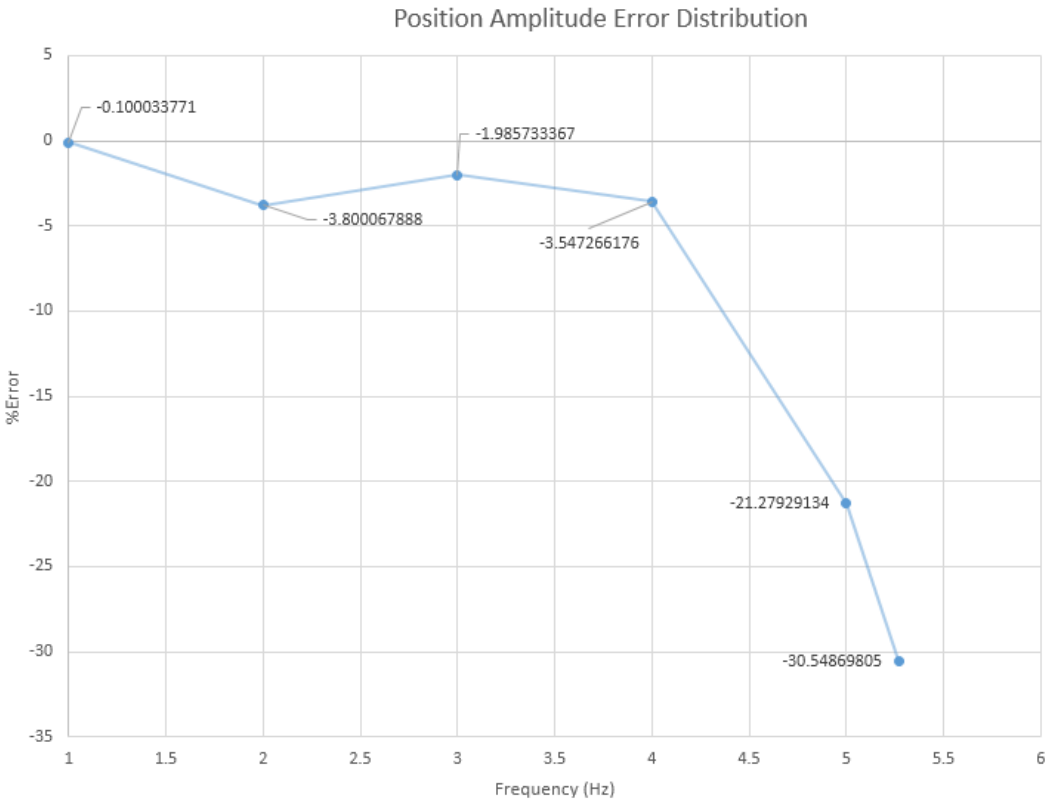

Figure 14: Position error vs. frequency for an input amplitude of π radians

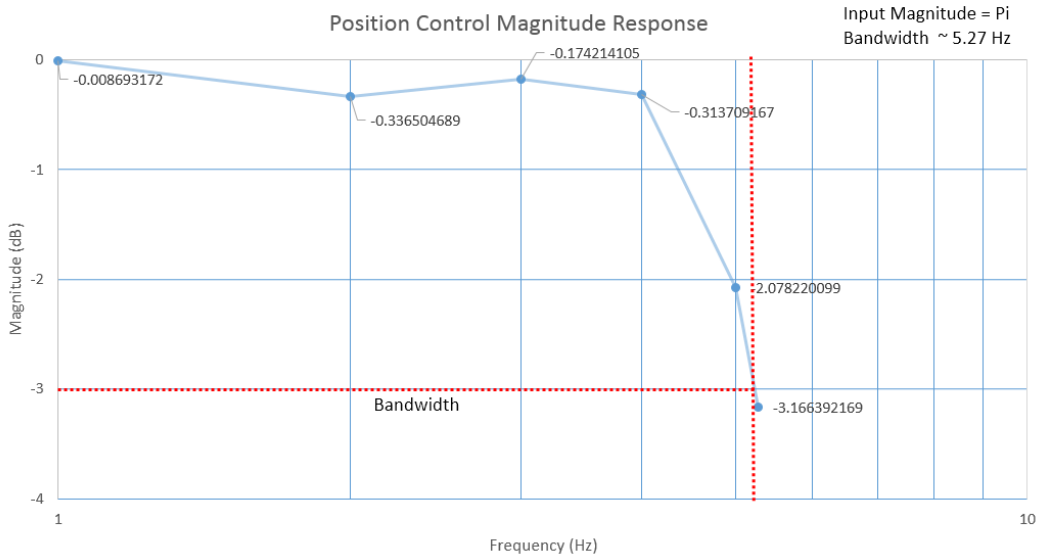

Figure 15: Output magnitude vs. frequency used to determine bandwidth of position controller

5.

a. Coulomb Friction

The bearings in the motor are the primary source of coulomb friction in the motor encoder system. This could be especially true in the current configuration of the motor because Pittman's website only specifies that they use ball or sleeve bearings in their products. If these are only radial bearings with no thrust capability, then the motor's performance could be affected by the vertical mounting we are using. Additionally, there is friction between the brushes and commutator. In section 2b we discussed and carried out experiments and modeling to determine the friction value.

Setting aside the earlier parameter identification, there are some simple experiments that could be run to estimate the friction in the system. If we assume that the torque constant given in the data-sheet is correct and that the gain of the servo-amp is exact and known, then for a given command voltage, the motor should be capable of a specific torque output. By slowly increasing the voltage from zero until the motor just begins to turn. From this, the torque caused by static friction would be equal to the product of the torque constant and the input voltage (because there is a 1:1 relationship between command voltage and applied current).

A similar method could be used to determine sliding friction. By applying a constant voltage to the system, the motor would come to a steady-state velocity. If we the coefficient of damping is known, then the torque caused by sliding friction is equal to the motor torque (K_t * $V_{command}$) minus the torque from damping (the product of the damping coefficient and velocity). This was carried out in part 2b.

Friction was included in our Simulink model from the beginning of our modeling, as presented in part 3. At the beginning of our tests we used approximations based on the data-sheet values, but as we performed more regression analysis, we obtained new values that should be closer to the true values of the physical system.

b. Saturation

Position Saturation

comment on actuation saturation

Step Input Amplitude (Radians)	1π	1.5π	2π	3π	4π
Rise Time (seconds)	0.0700002	0.0800000	0.0850000	0.110000	0.125
Settling Time (seconds)	0.290000	0.320000	0.28000	0.54500	1.45500
Percent Overshoot (%)	6.79997	16.8000	31.8500	65.0331	91.0003
Steady State Error (%)	0.100034	0.594022	0.150008	0.13335	0.399713

Table 9: Step Response for a variety of input positions. Where rise time is defined as reaching 90% of the final value while settling time is the time after which the function remains within 2% of its final value

c. Quantization

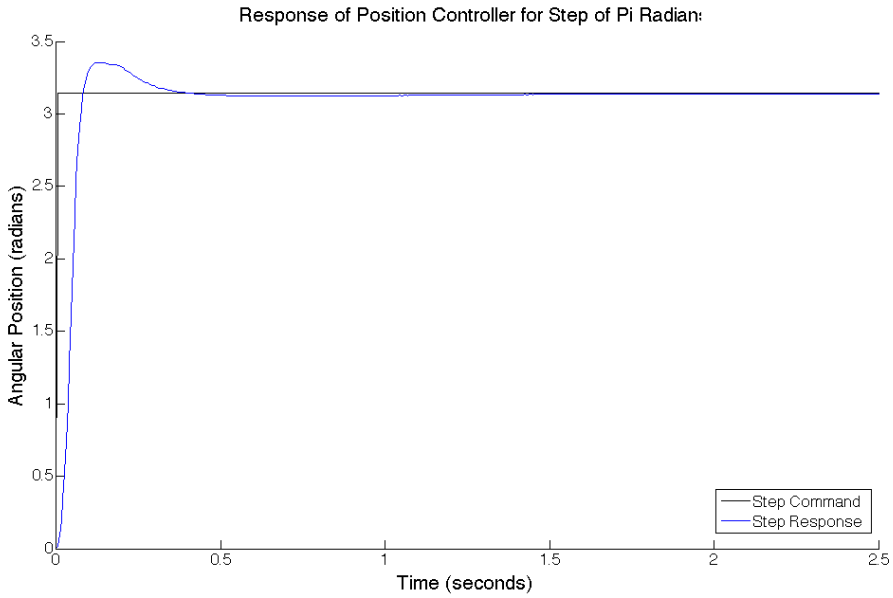

Figure 16: Step response of position controller

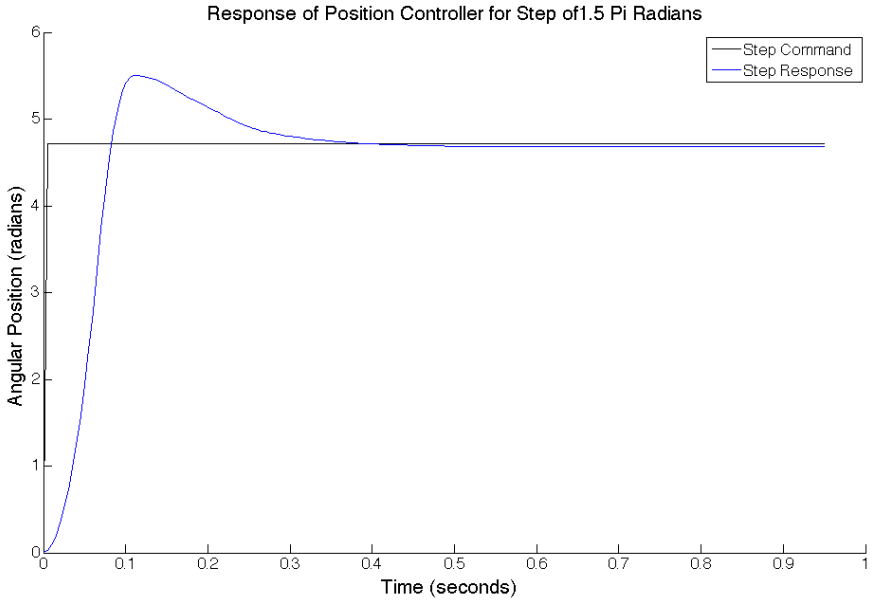

Figure 17: Step response of position controller

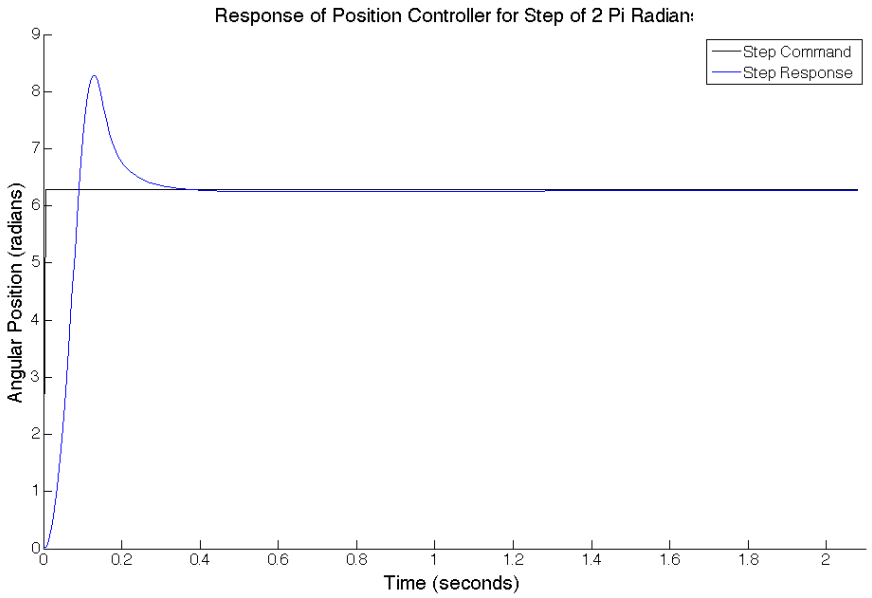

Figure 18: Step response of position controller

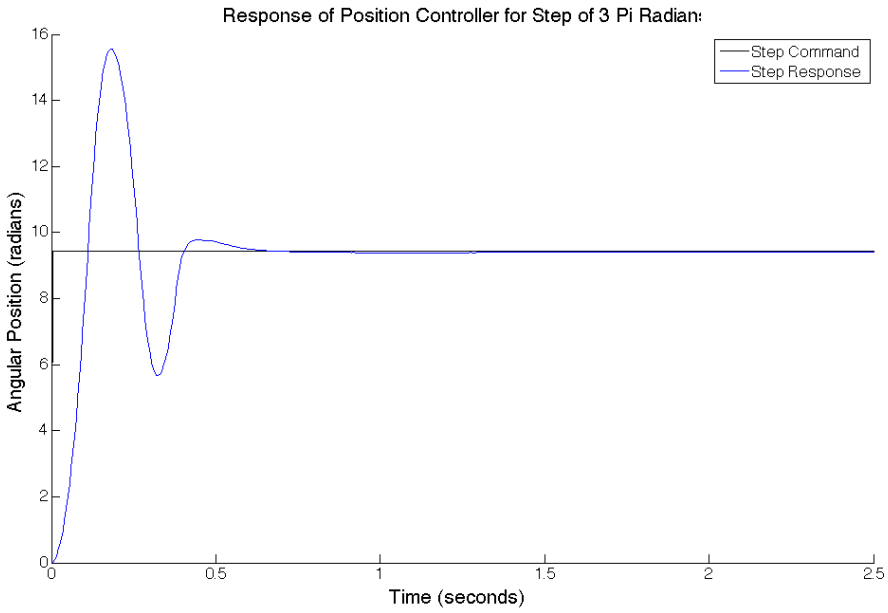

Figure 19: Step response of position controller

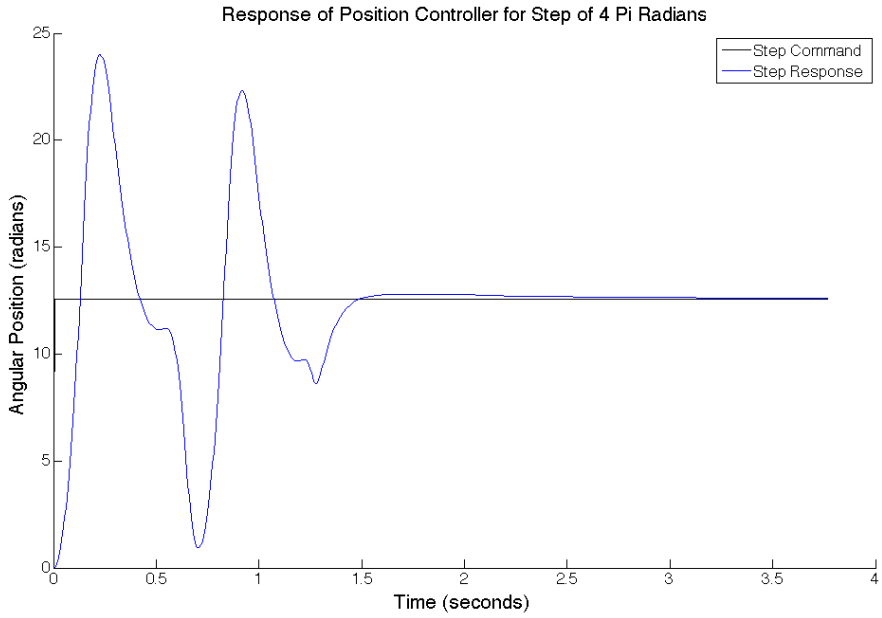

Figure 20: Step response of position controller

6.

As was discussed in section 4, work on the velocity controller for the real system was performed before the position controller, because we needed to nest the velocity loop within the position loop. The gains we used in the simulation were not able to stabilize the system with the performance we wanted, so we re-tuned the gain values. As with position control, we were not able to use one controller to meet all of the requirements: we designed one controller that could meet the step and noise requirements, and then modified the gain values to improve the frequency response separately. The results for the step response and noise attenuation are given below. The gains used here are the same as in the step position controller's nested velocity loop.

Gains	P	I	D	Filter τ
Velocity Controller	22.5	22.5	1.125	6.5797

Table 10: Values of velocity controller for step response and noise attenuation

Figure 21 shows the system's response to a velocity step input of $2{\cdot}\pi$ radians, and figure 22 shows a zoomed in view after it had settled. The peak error here is approximately 0.16% (and on average is much less), which greatly exceeds the goal of 2% error. Additional results for velocity steps of different magnitudes are also shown in the following tables and figures.

Compare to simulation results

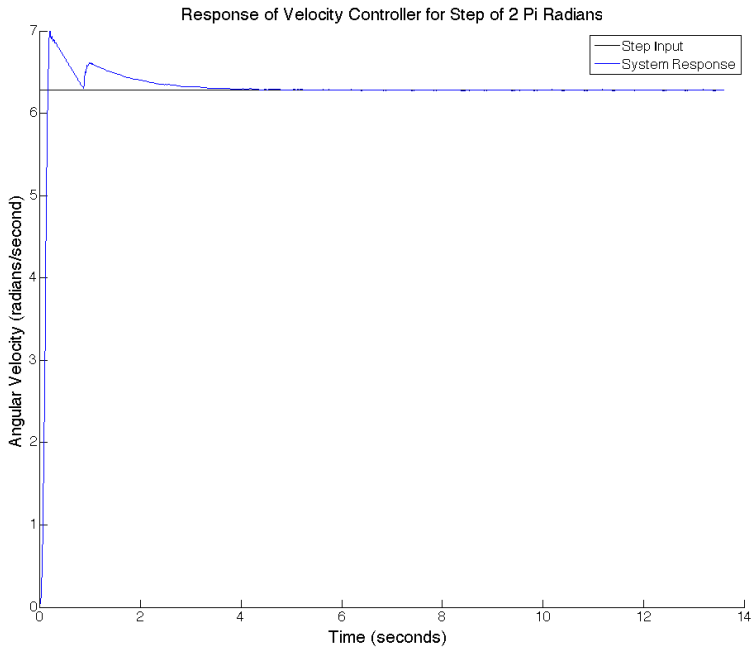

Figure 21: Velocity Controller Step Response

Figure 28 shows the noise attenuation during a step input of $2{\cdot}\pi$ radians/sec. To simulate the noise within LabView, we added a 60 Hz sine signal with an amplitude of 1 to the output of the DAQ assistant, just as with the position controller. Figure 29 shows a zoomed in view after the step has taken place.

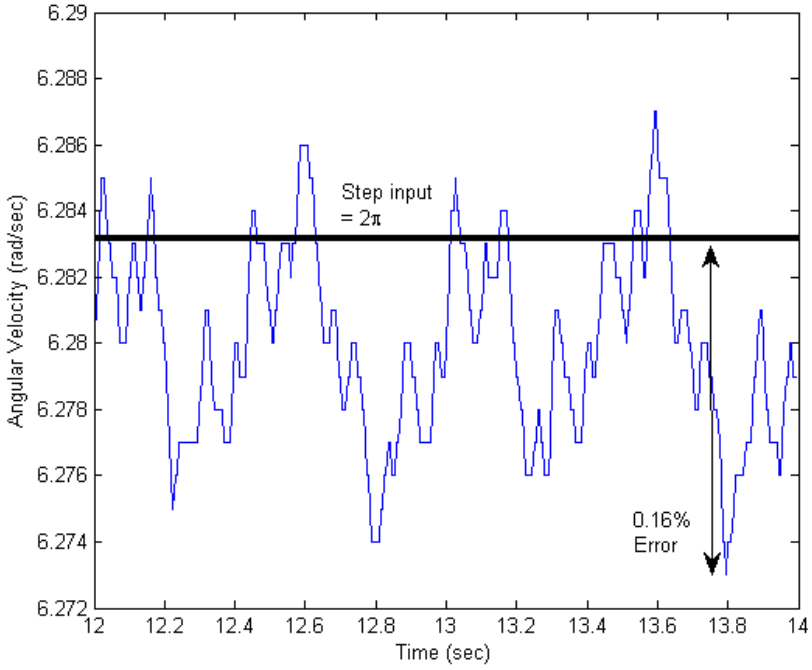

Figure 22: Steady state error for step input of $2 \cdot \pi$ rad/sec

Step Input Amplitude (Radians)	$0.5 \cdot \pi$	$1 \cdot \pi$	$1.5 \cdot \pi$	$2 \cdot \pi$	$5 \cdot \pi$	$10 \cdot \pi$
Rise Time (seconds)	0.11	0.115	0.125	0.14	0.23	0.385
Settling Time (seconds)	1.5700	1.6700	1.95500	1.9700	2.3050	2.8550
Percent Overshoot (%)	4.3415	6.1564	7.9919	11.3925	21.091	27.1553
Steady State Error (%)	0.11779	0.041862	0.053597	0.058333	0.050267	0.050634

For the step response with no noise, our peak steady state error was 0.010, and with the noise the highest error is 0.20. Given that we were inputting a noise amplitude of one and the error only increased by 0.19, we have attenuated the noise by approximately 5.26 times. This meets the goal of a five times reduction.

Compare to simulation results

Step Input Amplitude (Radians)	$2 \cdot \pi$	$2 \cdot \pi$ with Noise
Rise Time (seconds)	0.14	0.135
Settling Time (seconds)	1.9700	-
Percent Overshoot (%)	11.3925	29.6000
Steady State Error (%)	0.058328	0.054649

The velocity controller we were using for the step response and noise attenuation did not not perform well with sinusoidal inputs, therefore we modified to gains to try to improve the frequency response. The relevant values are given in the table 11, and the following figures show the response at various frequencies with an amplitude of $\pi/2$. Figure 30 shows the error as a function of input frequency. During our tests the velocity output remained stable up to 5 Hz, but the output amplitude error increased greatly. The error

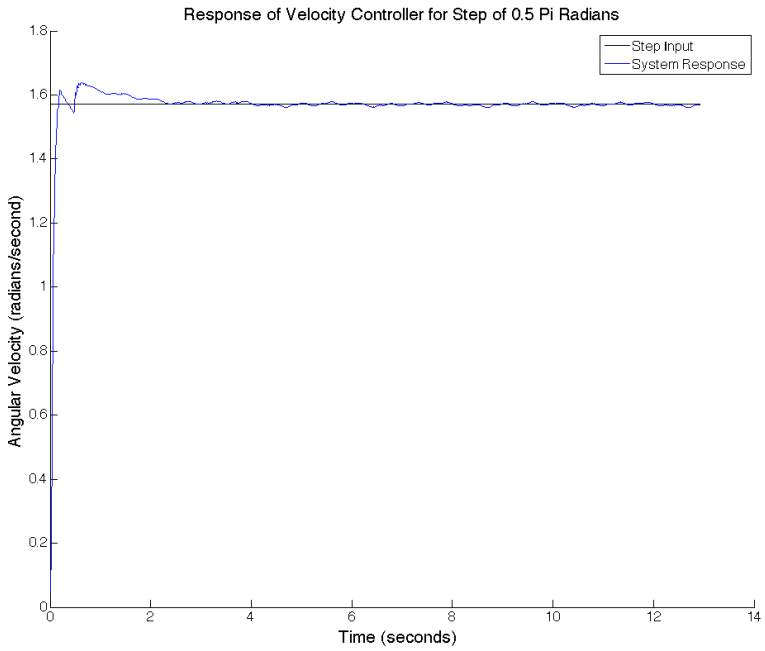

Figure 23: Velocity Controller Step Response

exceeded 5% just beyond 2.2 Hz. This does not meet the goal of less than 5% error up to 5 Hz.

Compare to simulation

Gains	P	I	D	Filter τ
Velocity Loop	2	9	0.3	6.5797

Table 11: Values of velocity controller for frequency response

$$\text{Predicted Bandwidth } = SOMETHING$$

$$\text{Experimental Bandwidth } = 2.9Hz$$

To determine the closed-loop bandwidth of the system, we slowly increased the frequency of the input signal until the output amplitude just crossed -3 dB (for a input of $\pi/2$ this corresponded to an output of 1.11), just as we did in section 4 to determine the bandwidth of the position controller. We found that the bandwidth of our position control system was 2.9 Hz, as shown in figure 31.

Just like with the position controller, we faced a lot of trouble meeting all of the performance specifications with just one controller. Again, we found that when we adjusted gains to meet the step requirement, the frequency response usually got worse, and vice-versa. As we mentioned in section 4 ,it's possible that some of the values used in our simulation are incorrect, or that there are other factors we did not account for. In the end we were able to meet the steady state error for step input and noise attenuation requirements, but were not very close to the frequency response requirement.

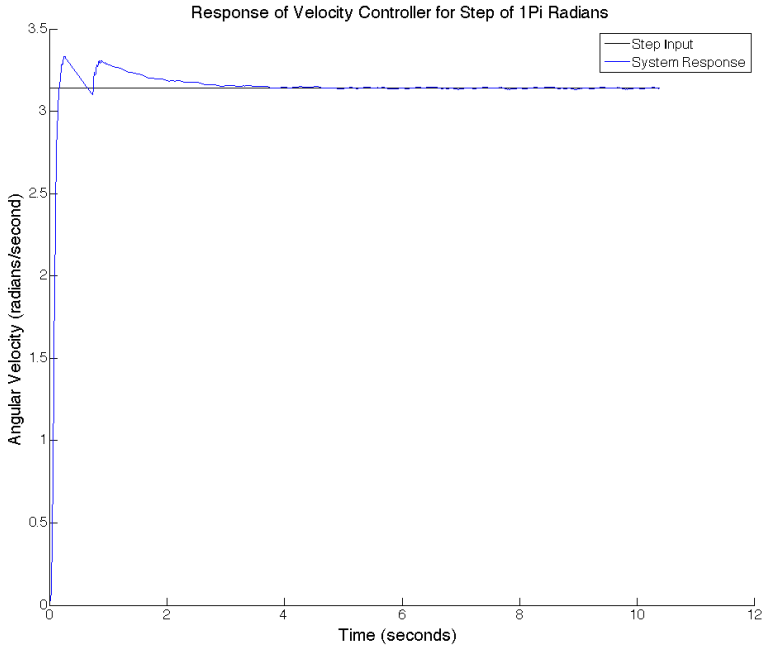

Figure 24: Velocity Controller Step Response

Frequency (Hz)	Input Amplitude	Output Amplitude	Magnitude (dB)	%Error
1	$\pi/2$	1.649	0.422015572	4.978600463
2	$\pi/2$	1.632143	0.332766594	3.905450513
2.9	$\pi/2$	1.11778	-2.955270847	-28.83991508
3	$\pi/2$	1.06516	-3.374100561	-32.18980833
4	$\pi/2$	0.581875	-8.625803574	-62.956687
5	$\pi/2$	0.47833	-10.32784514	-69.54856643

Table 12: Numerical results of velocity control.

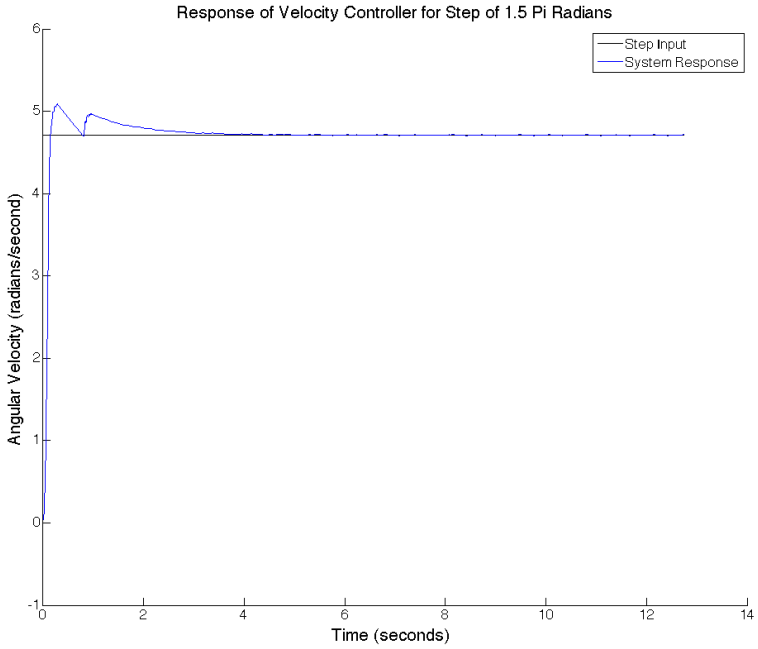

Figure 25: Velocity Controller Step Response

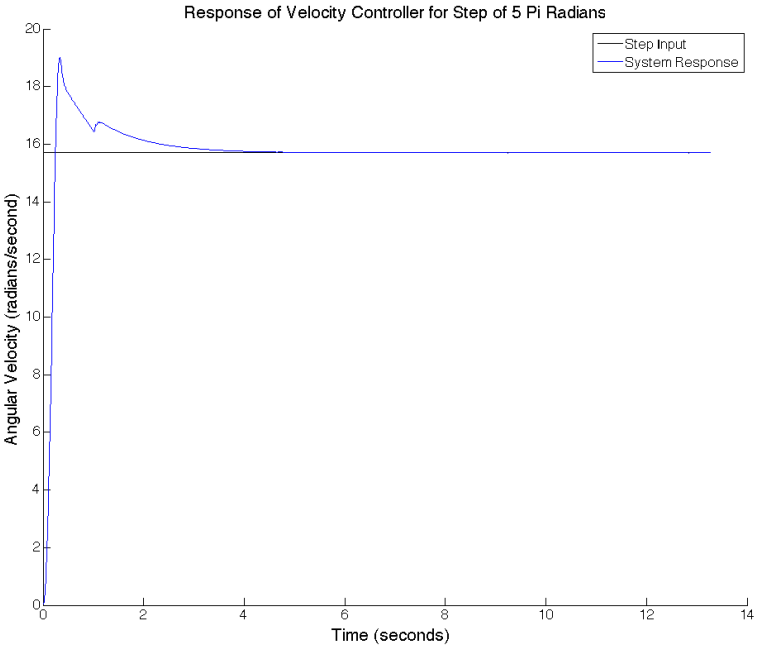

Figure 26: Velocity Controller Step Response

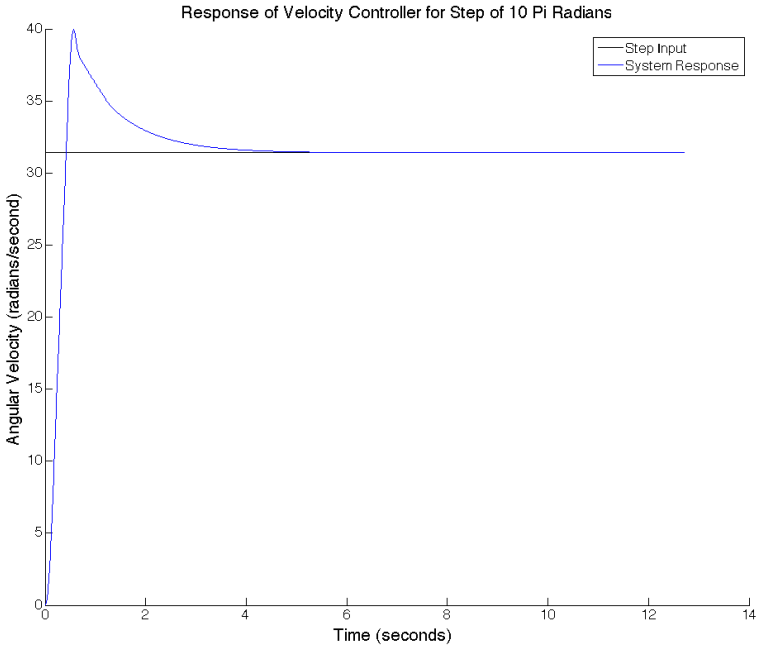

Figure 27: Velocity Controller Step Response

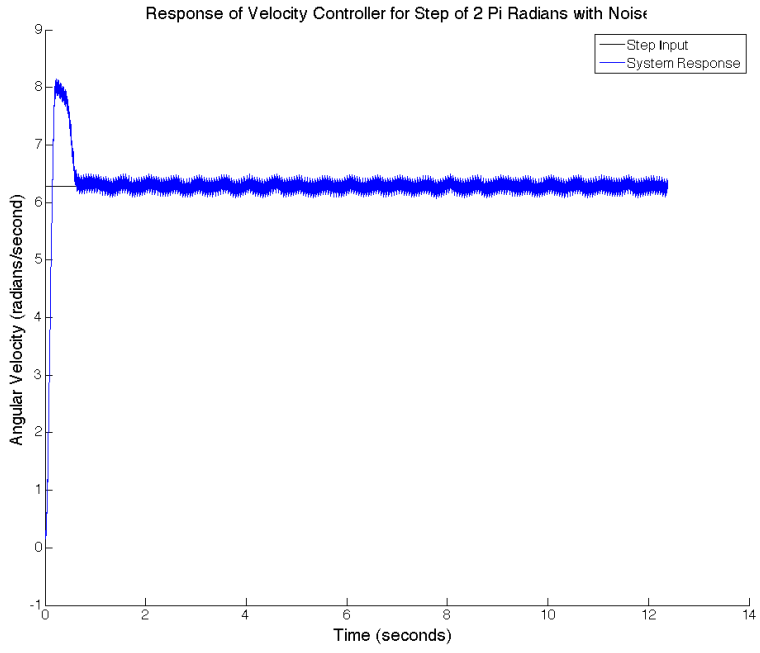

Figure 28: Velocity step response for input of $2 \cdot \pi$ with a 60 Hz noise signal (amplitude = 1)

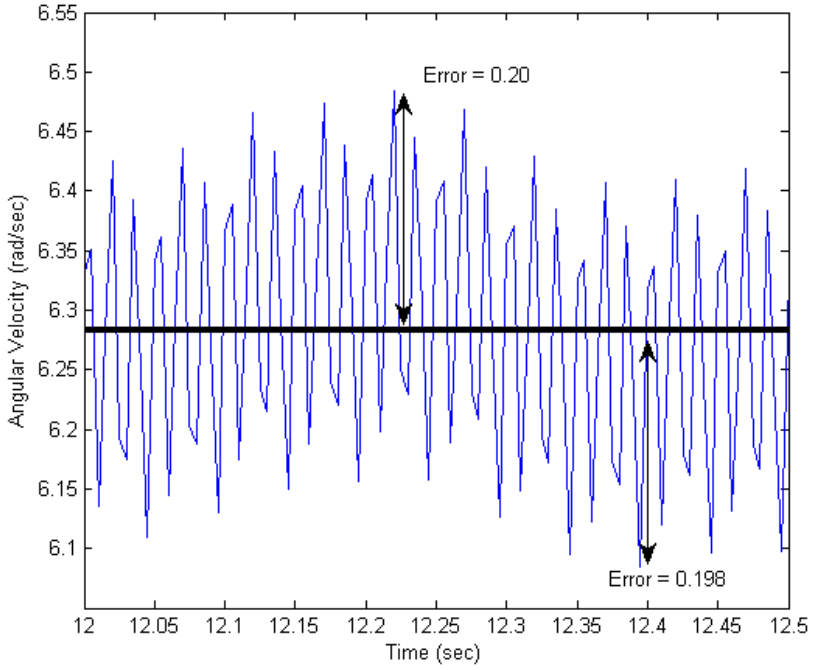

Figure 29: Steady state velocity step response for input of $2 \cdot \pi$ with a 60 Hz noise signal (amplitude = 1)

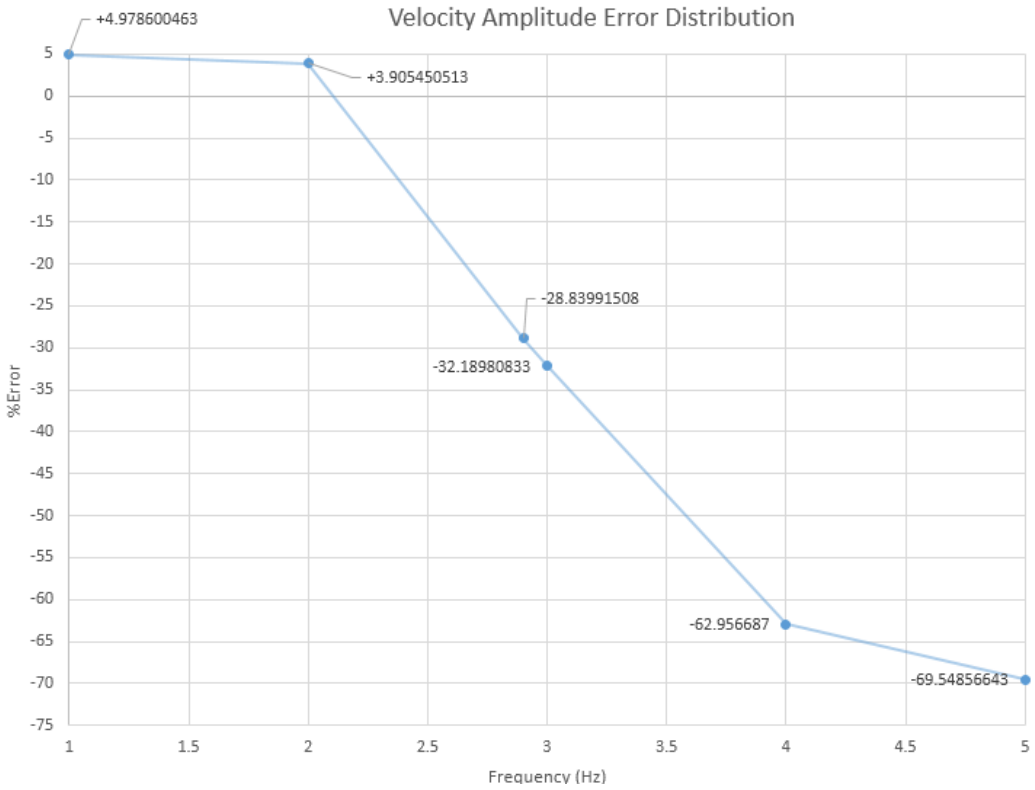

Figure 30: Velocity error vs. frequency for an input amplitude of $\pi/2$ rad/sec

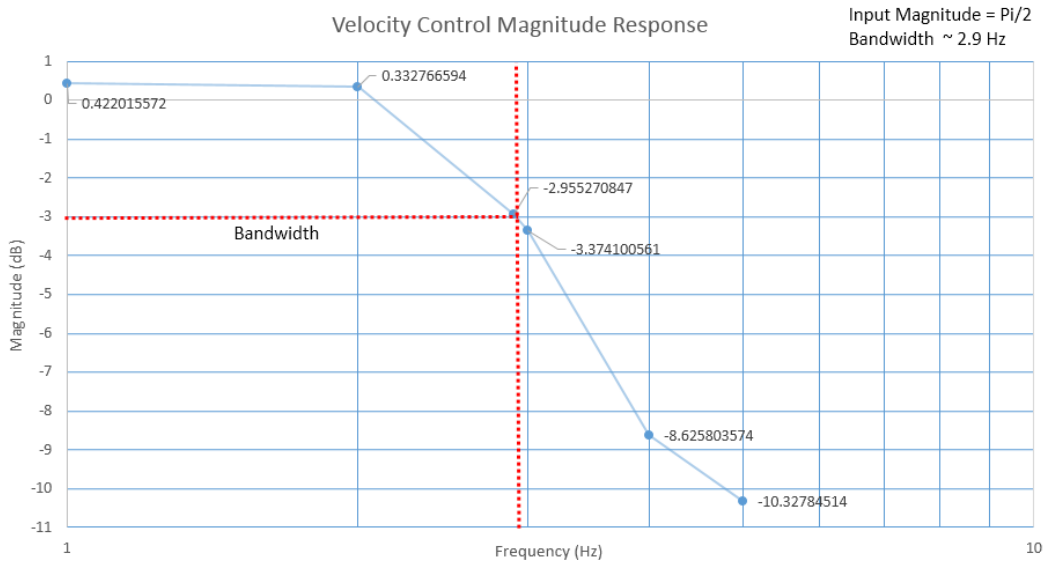

Figure 31: Output magnitude vs. frequency used to determine bandwidth of velocity controller