# Observer-Based Robot Arm Control System

Final Project Report

By:
Alex Certa
Ron Gayles
Nick Vogel

Advisor:
Dr. Dempsey

May 9, 2012

## Table of Contents

## Abstract

This report covers system ID, classical control, and observer based control methods for two robot arms. All data collection and design was done in the Matlab / Simulink software. The purpose of this project was to investigate the differences between classical and observer based controls on these two systems. The control parameters that are used for comparison are percent overshoot, settling time, phase margin, gain margin, and steady state error.

## Project Summary

The goal of this project is to use Ellis's method of observer-based control to control complex configurations of a robot arm system. Observer-based control simulates the system and fuses any available sensors to estimate the state of the system. Ellis's method is based largely on PID-style controllers, making it a relatively simple style of control for someone with experience in PID tuning or for someone without advanced control knowledge. Two systems are being used for this project. One is a horizontal arm configuration with two degrees of freedom, with a spring system associating the second degree of freedom to the first. The other configuration is the pendulum configuration. This configuration is vertical and thus has to deal with force of gravity.

The systems have both been tested with traditional control schemes, and models have been developed to approximate the response of the system. The actual observer and observer-based controller will be constructed in Simulink and run via Quarc from Quanser Consulting to control the system. The two systems should quickly and consistently move to commanded positions, and the systems should be able to consistently hand objects between them.

## Project Description

### Background Information

Control theory is applied in a vast variety of different fields including heating & cooling systems, cruise control, assembly line automation, and nuclear reactor control. Most applications still use Proportional-Integral-Differential(PID) control because it is relatively simple to use and generally provides sufficient results for many applications while being easy to understand.

PID control consists essentially of three parallel paths summed in the forward path. The first, the proportional path, simply is a constant times the error. The second is integral, which is a constant times the integral of the error. The third term is the derivative term, which consists of a constant times the derivative of the error. The three paths can be seen in Figure 2-1.

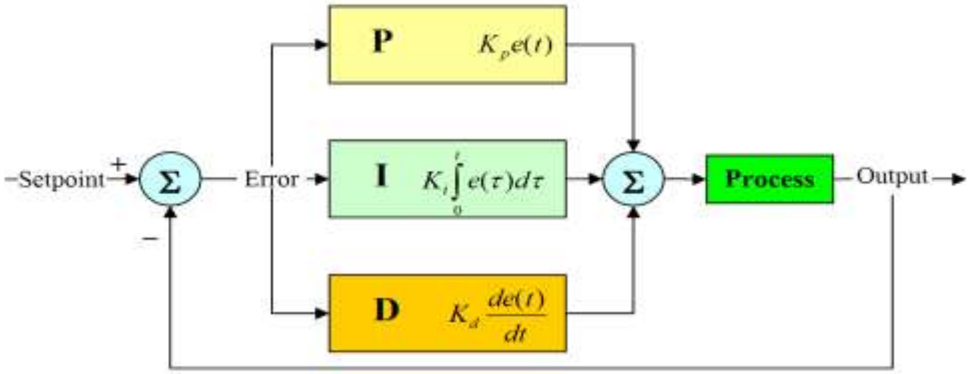

**Figure 2-1 PID Control Diagram.**

In practice, the PID controller cannot perfectly replicate the theory. The derivative term is not realizable in practice and so instead it is approximated by a first order high pass filter with a relatively high corner frequency. The transfer function for the derivative term is kp*s/s+p1, where p1 is large enough for it to approximate a differentiator for the specific application.

Observer-based control is one of the newer, more advanced concepts in control. Observers simulate a system based on available data, including sensor readings and command input and controller output. The method of simulating and controlling using this concept varies from observer to observer, but one of the most common types of observers used is based on linear algebra.

One method of observer-based control that stands out in particular is the Kalman Filter. This filter uses knowledge of the variance and covariance of the noise of different sensors in order to minimize the mean squared error of the measurement. This method, while being in some sense 'optimal', requires knowledge of statistics, linear algebra, and much time spent studying the sensor outputs.

George Ellis, however, proposes a system that is largely based on PID control. The general control system set up can be seen in Figure 2-2.

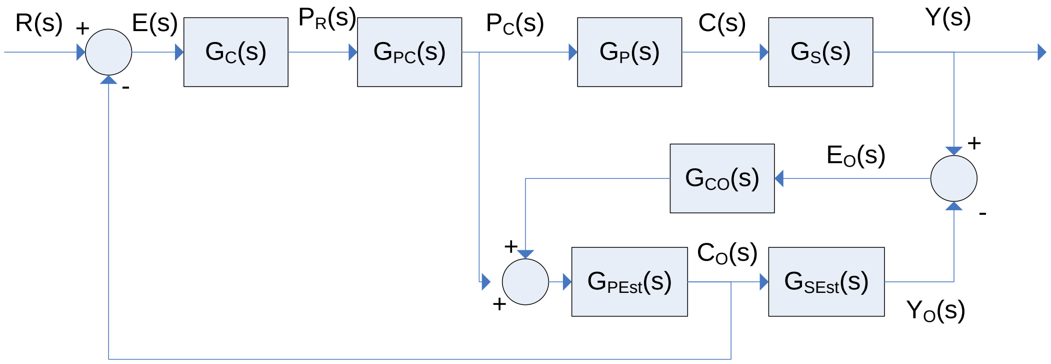

**Figure 2-2 Ellis's Method of Observer-based Control**

R(s) is the set point input of the system. E(s) is the estimated error of the system, which is then fed into a PID controller Gc(s), and through the power converter Gpc(s). This signal is sent both to the actual system Gp(s) and to the observer. The system is then monitored by the sensors, which have a transfer function Gs(s). The output is then compared to the output of the observer, and the difference is sent through a PID controller Gco(s) and added to the input to help drive the error of the system to zero. The observer then estimates Gp(s) in Gpest(s), and this signal is then used to control the system. The observer also estimates the sensors in Gsest(s) in order to provide an output without significant phase lag.

## Functional Description and Block Diagrams

With that background the motive of our project is clear: evaluate the usefulness of Ellis's method of observer-based control as a simple alternative for complicated robot arm systems.

The overall goals of the project are:

- Learn the Quanser software package and real time control via Simulink.
- Obtain a mathematical model for the pendulum arm and the horizontal arm.
- Design controllers for each system using classical control methods.
- Design a controller for each system that uses observers to predict the plant’s response.
- Evaluate the performance of the two control methods and compare the result.

The workstation for each robot arm consists of the following components:

- PC with Matlab and Simulink
- Motor with Quanser Control System
- Linear Power Amplifier
- Robot arm with Gripper
- SRV-02 Rotary Servo Plant
  - One robot arm will also contain a SRV-02 Rotary Flexible Joint to add another degree of freedom.
  - The pendulum robot arm contains a rotary encoder, and the level robot arm contains a potentiometer to measure position.

One robot arm will be configured vertically in a pendulum-like fashion to incorporate the effects of gravity on the arm shown in Figure 4-1. The other robot arm will be placed horizontally and will have a flexible joint to add a second degree of freedom that is independent from the base of the system, shown in Figure 4-2. A closed-loop PID control system will be implemented in Simulink and will use Quanser software to allow real-time control of the robot arms through Simulink.

System Inputs:

- Internal Commands (position and velocity)
- Potentiometer Position Feedback (2 DOF arm configuration)
- Rotary Encoder (pendulum arm configuration)

System Outputs:

- Position
- Velocity

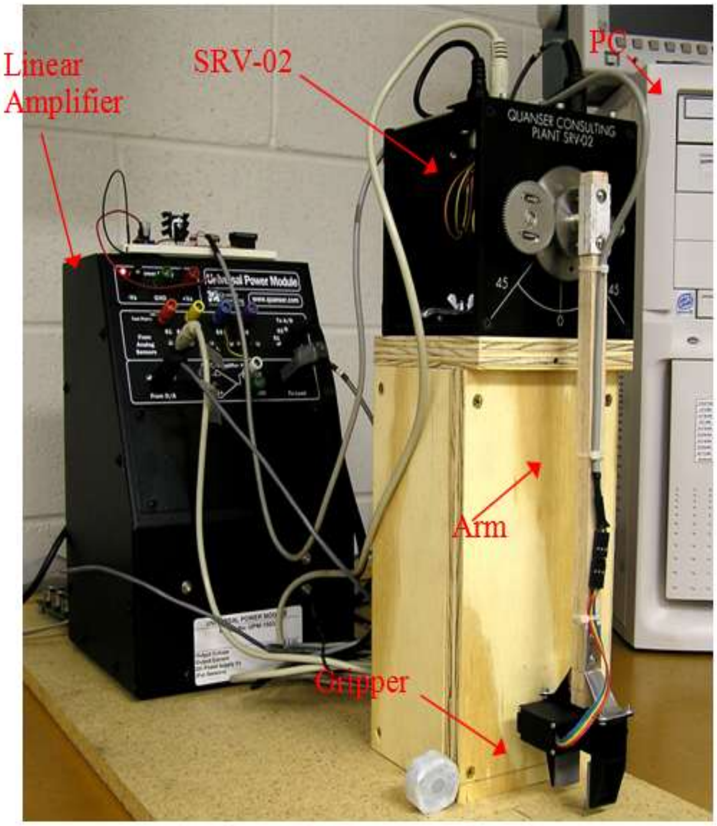

**Figure 4-1 Pendulum Robot Arm Configuration**

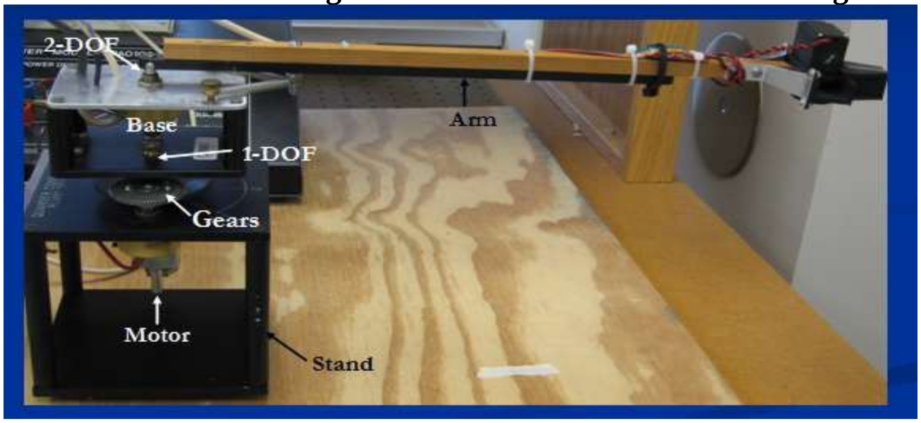

**Figure 4-2 2-Degree of Freedom Robot Arm**

The high level block diagram for the project is shown in Figure 5-1. The command signal is set in Simulink, which then sends a signal to the implemented arm controller, which then sends the signal to the arm. Sensors connected to the arm then send feedback to the controller allowing closed loop control. The power electronics involved with the robot arm controller, the robot arm itself and the sensors all introduce external disturbances including power supply noise, changes in load, friction, and quantization error.

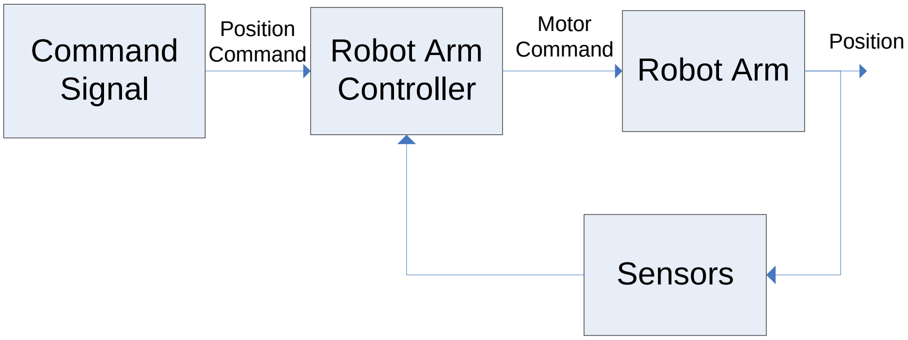

**Figure 5-1 Overall System Block Diagram**

Figures 5-2 and 6-1 show detailed views of the robot arm systems, the sensors, and the power electronics involved in the control of the arms.

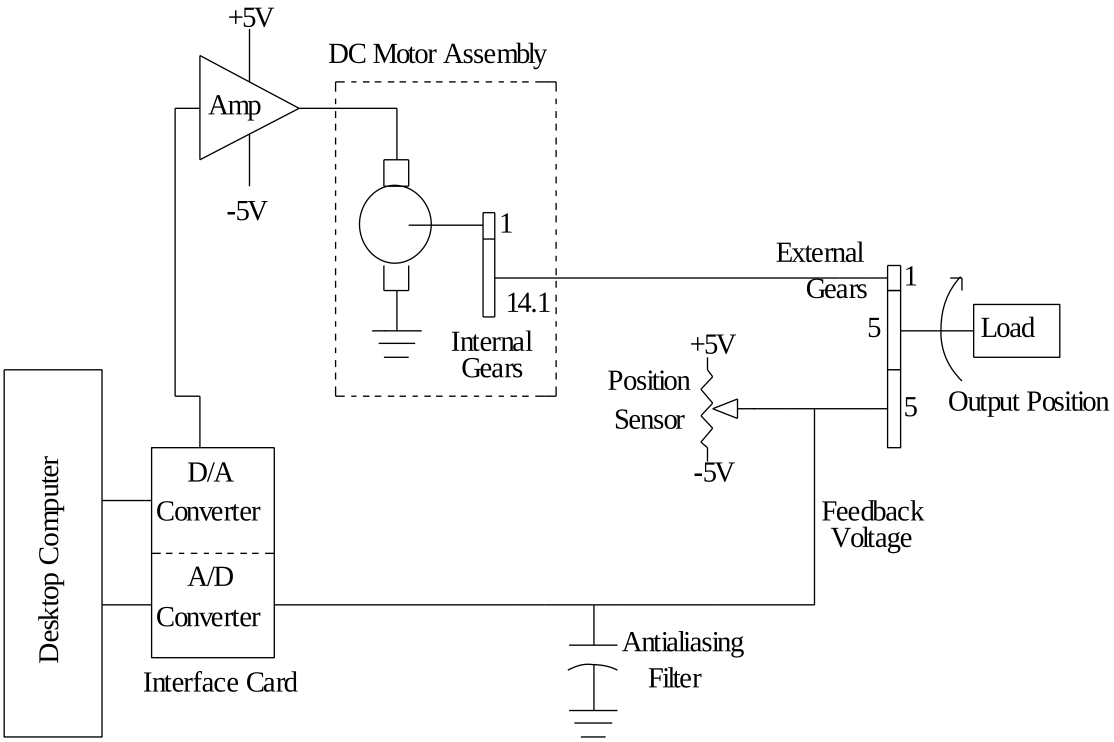

**Figure 5-2. Quanser Electromechanical Plant with Potentiometer**

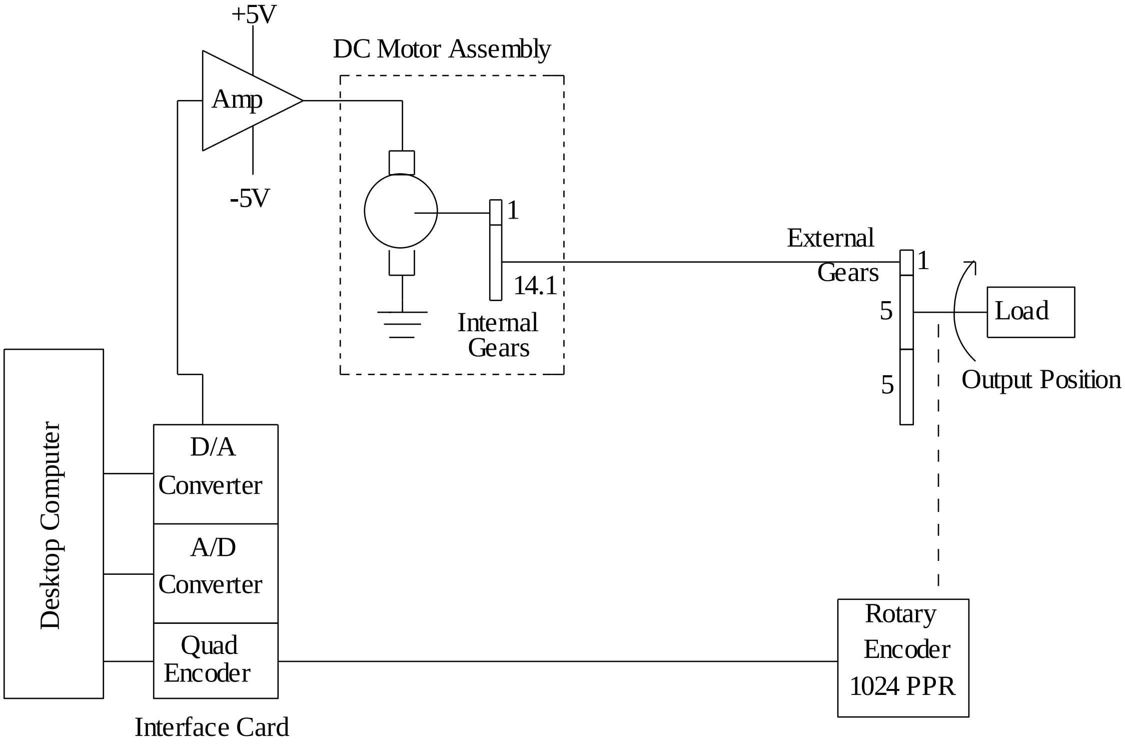

**Figure 6-1. Quanser Electromechanical Plant with Rotary Encoder**

The first, simple type of controller used is a single loop controller shown in Figure 6-2. Gc represents a controller transfer function, which varies from simply a gain, representing a proportional controller, to a PID controller or even more complicated system with more poles and zeroes. Gp is the transfer function of the system being controlled, and H is the transfer function of the sensors used. The more complicated final system is shown in Figure 2-2.

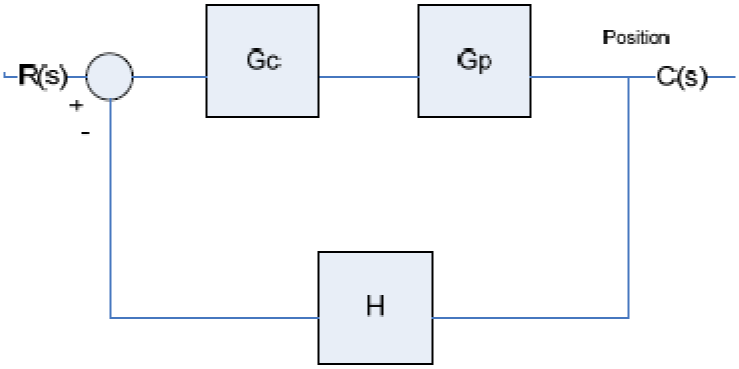

**Figure 6-2 Single Loop Controller**

## Functional Requirements & Performance Specifications

The high level block diagram for the project is shown in Figure 5-1. The command signal to the system will be a value that is assigned in Simulink. This value will be limited to plus or minus 90 degrees. The position command will be passed to the controller via Simulink. The controller will generate a digital control signal, which will be converted to an analog signal in the range of ±5 volts via a D/A converter. The position of the arm will be measured two different ways in the two platforms. The 2-DOF platform will use a potentiometer to measure position. The analog position signal will be fed into an A/D converter. The digital signal will then be compared to the reference signal to generate an error signal. The pendulum arm platform will use a rotary encoder to measure position, which is fed into the quad encoder interface to the computer. The signal is then compared with the reference signal to generate an error signal to drive the controller.

The level 2-DOF configuration shall be controlled using different types of classical controllers and observer-based controllers. The system shall also perform disturbance rejection for a load. The specifications for the performance of this system for a step command of 90 degrees are as follows:

- The overshoot of the arm shall be less than or equal to **15%**
- The settling time of the arm shall be less than or equal to **2s**
- The phase margin shall be at least **50 degress**
- The gain margin shall be at least **3.5**
- The sample time shall be **10 ms**
- The steady state error of the system shall be less than **2 degrees**

The pendulum arm configuration will go through the same design process as above. The system shall perform disturbance rejection for a load. The specifications for this configuration given a 90 degree step command are as follows:

- The overshoot of the arm shall be less than or equal to **15%**
- The settling time of the arm shall be less than or equal to **2s**
- The phase margin shall be at least **50 degrees**
- The gain margin shall be at least **3.5**
- The sample time shall be **10 ms**
- The steady state error of the system shall be less than **1 degree**

For both of these systems, the specifications shall hold for loaded conditions. The controllers will be designed to work with the existing robot arm system, and A/D and D/A converters.

**Design Equations and Calculations**

| | |
|---|---|
| $T_s = 4/(\zeta * \omega n)$ | Settling Time Equation |
| $\omega_n = x\pi/T$ | Natural frequency equation |
| $\%OS = 100 * e^{(-\pi * \zeta / \surd\ (1-\zeta\wedge 2))}$ | Percent overshoot equation |
| %OS=[Peak value-Final Value]/Final Value | |
| $PM = 100\zeta = 180 - \lvert angle\ Gp \rvert$ | Phase Margin Equation |
| $GM = 1/\lvert Gp*Gc\ @\ 180 \rvert$ | Gain Margin Equation |

## Completed Work for Pendulum Configuration

The work completed was done by two groups working separately on the pendulum robot arm system and the two degree of freedom robot arm system.

The preliminary research on the pendulum robot arm system was done in this order: first, the motor system was identified and controlled; second, the robot arm system was identified and a linear model constructed; third, proportional control was implemented; finally, PID control was implemented, then a non-linear model for the motor was constructed.

The first step taken was to control a relatively simple system: the Quanser station with the arm not attached. The system was identified using knowledge of the workings of motors along with data from the data sheet to construct a system block diagram. The model is shown in Figure 8-1

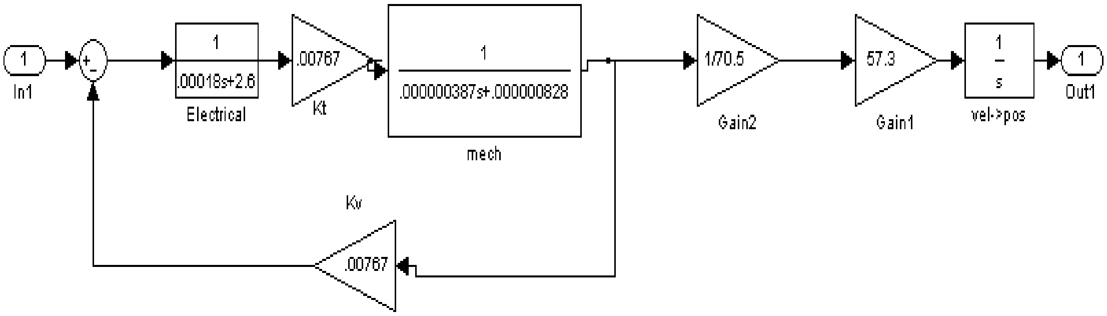

**Figure 8-1 motor and gear train linear model**

This model closely matched the response seen in the actual system. Based on this model, a proportional controller was constructed. The maximum velocity of the system appeared to be approximately 450 degrees per second. With a 180 degree input, there is a 0.25 degree steady state error, and it took 0.42 seconds to reach 90% of the way to 180 degrees. A feed forward model was then constructed. The poles for the model are at 0, -13000 and -60. Since the pole at -13000 is much further than the other poles, it can be ignored for this design.

A feed forward network attempts to cancel the low frequency effects of the plant. In this case at low frequencies the transfer function approaches 102/s. Thus for our feed forward design, the network should be s/102. This is a pure differentiator which cannot be implemented in practice, and so a pole is placed more than a decade from the relevant poles. The gain was then tuned for optimum results.

For a 180 degree input, the time to reach 90% of the value was the same, however, it reached steady state 7% faster and only had 0.1 degree of error, which is much smaller than the potential gear backlash and thus can be ignored.

The proportional and feed forward controller transient responses can be seen in Figure 9-1 and 9-2.

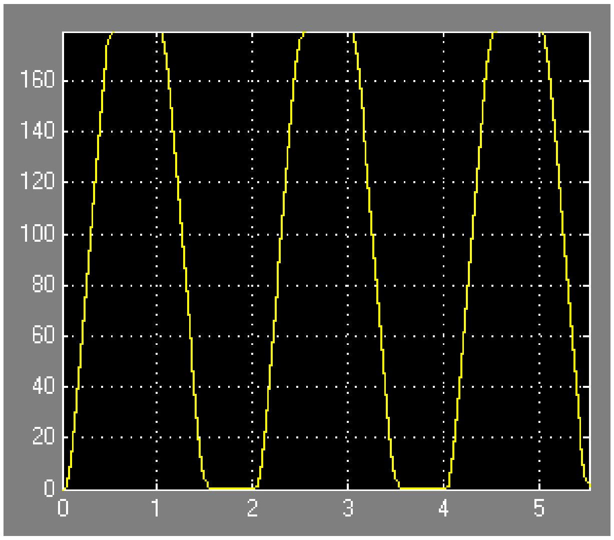

**Figure 9-1 Transient Response of Proportional Controller for Motor System**

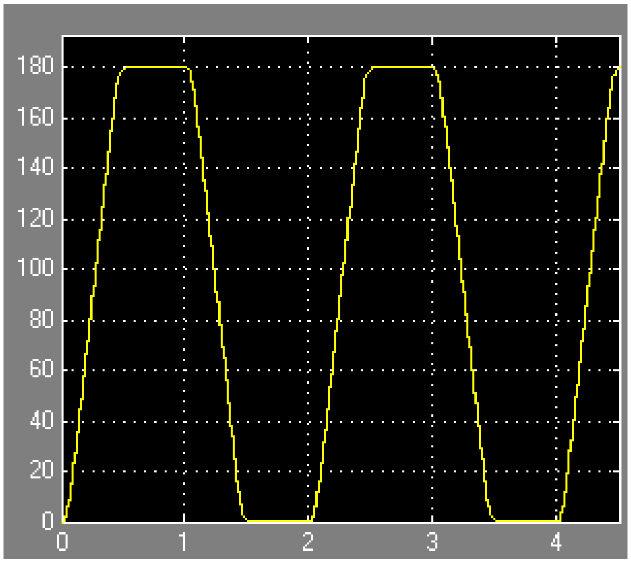

**Figure 9-2 Transient Response for Feed Forward Controller for Motor System**

After completing these controllers for the motor system, the arm was attached and the system including the arm was identified. Three primary methods were used to help identify the robot arm system. First, the steady state voltage response was measured to help determine the DC gain over the linear region. Second, a proportional controller with high gain was implemented and the $2^{nd}$ order transients were measured, then the exact second order equations were used to estimate the open loop pole locations. Third, the frequency response was measured to compare the results of the model with the results of the system.

The voltage response of the arm was measured and is shown in Figure 10-1. The next step taken was to divide by the angle to get the DC gain at each DC value. The response is shown in Figure 10-2. Small angle approximation guarantees that the DC gain due to gravity should be a constant over small angles. Our DC gain as shown in Figure 10-2 is not constant but appears to be linear over a large portion or the range. This disparity can be accounted for by Coulomb friction. Through trial and error, a 0.13 constant reduction in voltage was found to account for the slope of the DC gain. The modified DC gain graph can be seen in Figure 11-1.

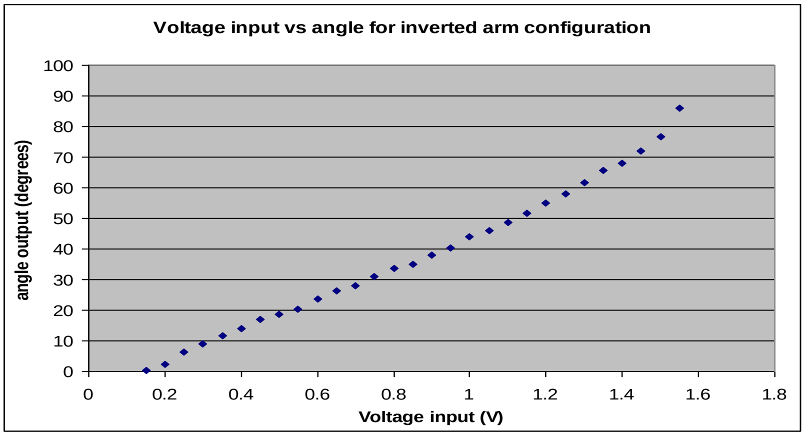

**Figure 10-1 DC Voltage Response of Robot Arm System**

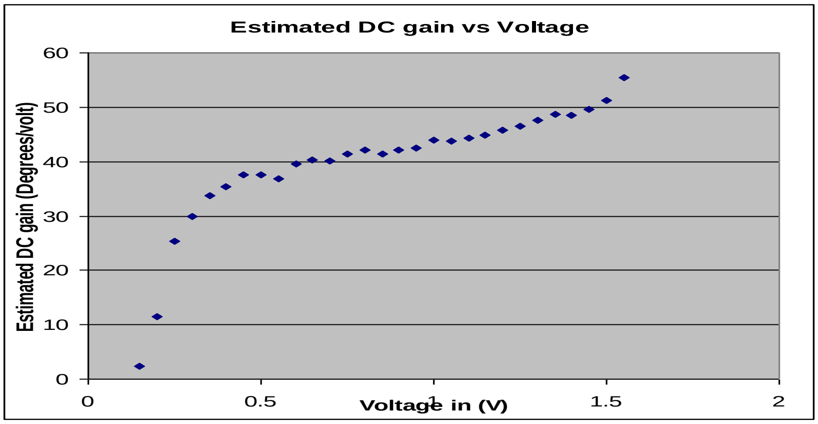

**Figure 10-2 DC Gain Response of Robot Arm System**

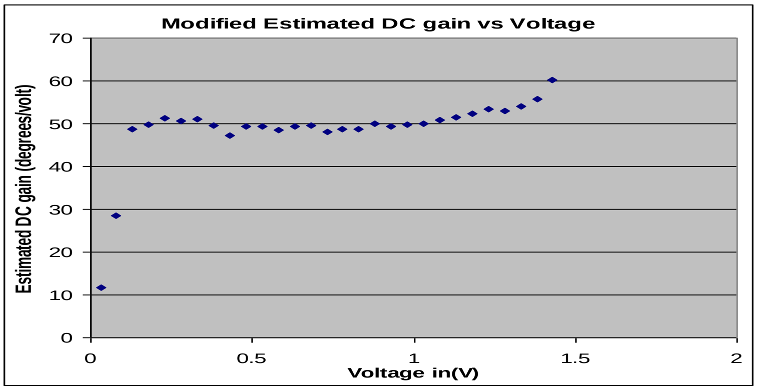

**Figure 11-1 Modified DC Gain Response of Robot Arm System.**

The second order step response was then generated using a proportional controller with a gain of 0.45. This gain provided sufficient overshoot so that the second order specifications of percent overshoot, settling time, rise time, and time to first peak could be accurately measured. The percent overshoot was 46%. The rise time was 0.06 seconds. The settling time was 0.58 seconds. The time to first peak was 0.14 seconds. The graph of the transient response can be seen in Figure 11-2.

The robot arm system was assumed to be a second order system in the form k/(s/p1+1)(s/p2+1). Based on the equations for an exact second order system, which should prove a good approximation, the poles were found to be at 2.6 and 11. The results of the system with the proportional gain and the results of the model with the same gain are shown in Figures 11-2 and 11-3.

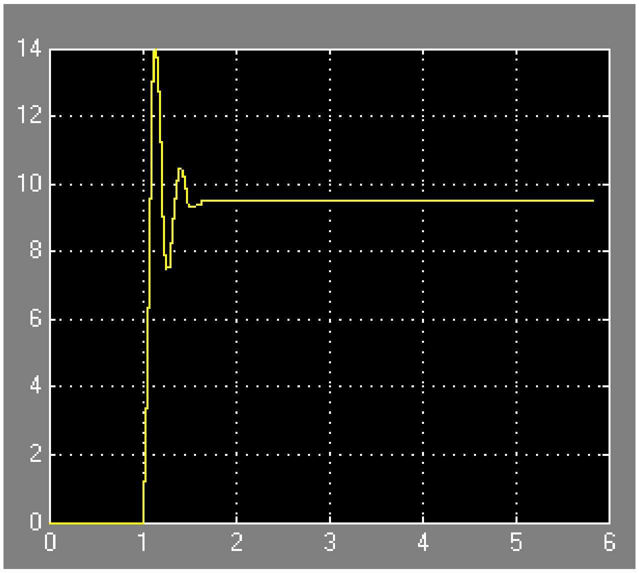

**Figure 11-2 Transient Response of High Gain Proportional Controller for Robot Arm System**

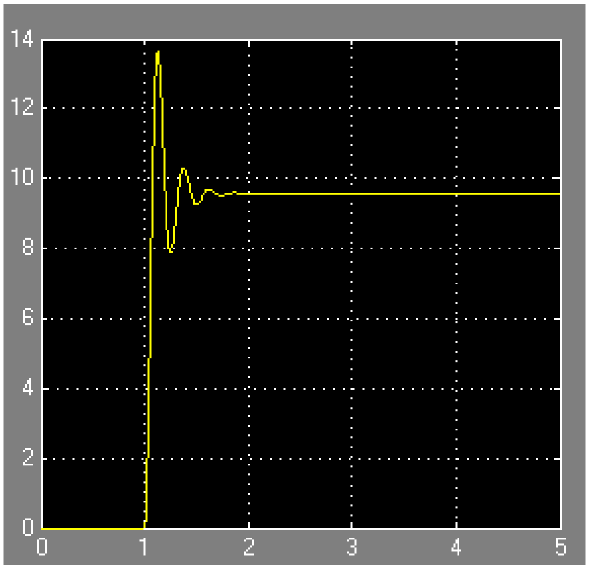

**Figure 11-3 Transient Response of High Gain Proportional Controller for Robot Arm Linear Model**

These responses are very similar, with only slight errors in the values at different times. The next step taken was to compare the system and the model open loop. The results are shown in Figure 12-1 and Figure 12-2. It is clear that the model and actual systems still have major differences, but these differences will be reduced when placing them in closed loop systems. The model has a much more gradual transition into steady state, but the actual system stops dead at a certain point. This difference can most likely be accounted for by static friction effects so that once the robot arm stops, it is much more difficult to get moving again.

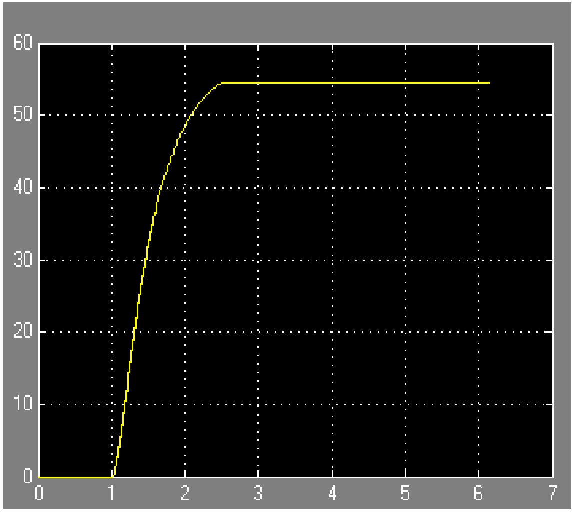

**Figure 12-1 Open Loop Transient Response of Robot Arm System**

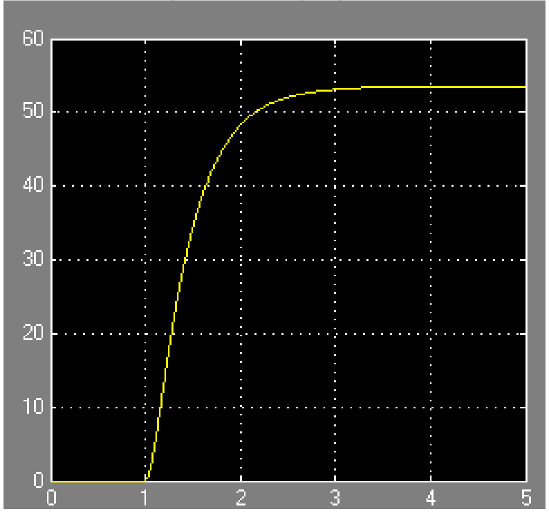

**Figure 12-2 Open Loop Transient Response of Robot Arm Linear Model**

Next, the magnitude frequency response of the robot arm system and the linear model were measured. The results are shown in Figure 12-3. The results were normalized to a nominal value of the DC gain. The system response is shown in blue, while the model response is shown in yellow. The system with a slight change to the DC gain is shown in pink.

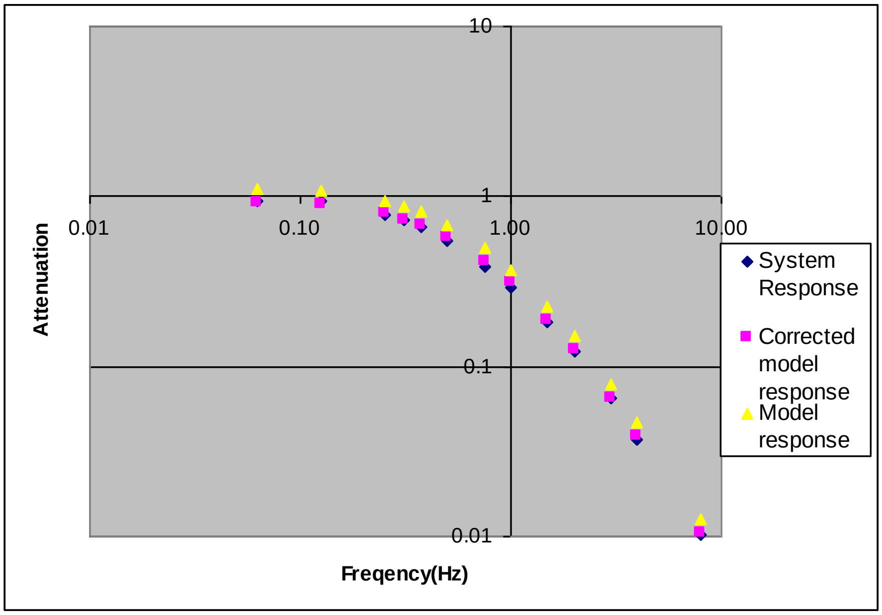

**Figure 12-3 Frequency Response of System and Model**

A proportional controller was then constructed with 15% overshoot for a 20 degree input. The steady state error was 2.5 degrees, the rise time was 0.12 seconds, and the settling time was

0.41 seconds. The transient response can be seen in Figure 13-1.

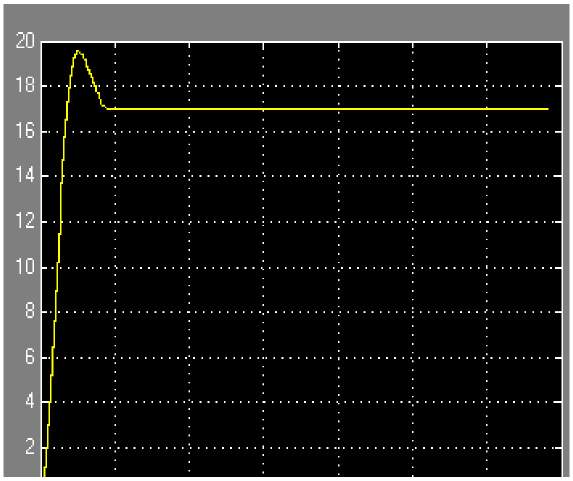

**Figure 13-1 Transient Response for Proportional Control of Robot Arm**

After the proportional controller was working, a PID controller was implemented to create an exact 2nd order system. The poles were cancelled with zeroes and a pole at the origin and another on the real axis were placed. Having the pole placed further out in theory makes the system faster, but with the D/A converter limitations the input has to be rate limited to avoid saturation. The system was tested with no saturation (with a 1 degree input), to see how the system would work without rate limitation. As expected, with the pole locations further out the settling time is lower for the same percent overshoot. However, for a 180 degree input the time spent in rate limitation dominates the actual transients, so the system is faster with a lower pole location that allows a larger rate limit. The results of this testing can be seen in Table 13-1. The final results of the PID controller are show in Figure 14-1.

**Table 13-1 PID Pole Location Determination Data**

| Pole Location | Gain value | overshoot % | Settling time | Rate limitation | Rate limited settling time |
|---|---|---|---|---|---|
| -40 | 0.75 | 14.9 | 0.20 | 155 | 1.16 |
| -80 | 1.5 | 15 | 0.10 | 148 | 1.20 |
| -60 | 1.1 | 14.9 | 0.14 | 151 | 1.18 |
| Rad/s | | | s | deg/s | |
| | | | 1 deg input | | 180 deg input |

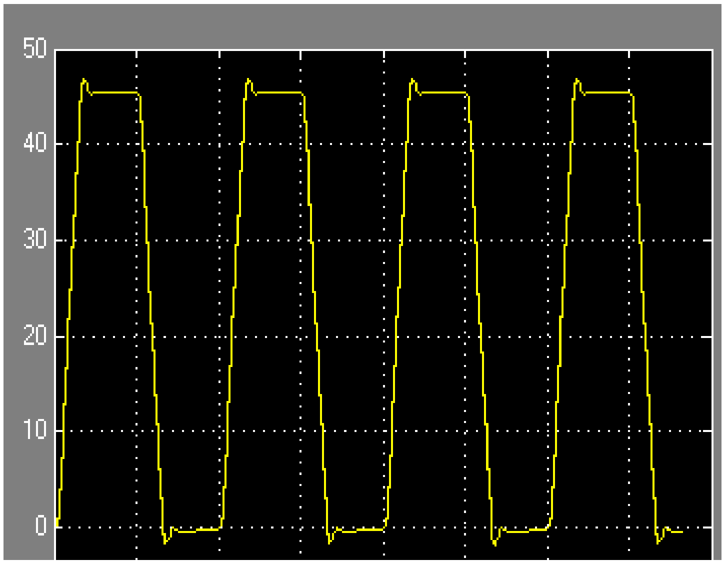

**Figure 14-1 Transient response for PID controller**

After working on the PID Controller, the model for the system under the loaded case was tested and compared to a DC gain adjusted model of the system. The results are shown in Table 15-1, and the graph is shown in Figure 14-2. The model is within four dB for frequencies under four Hertz. This was deemed as an acceptable model for initial testing of the system.

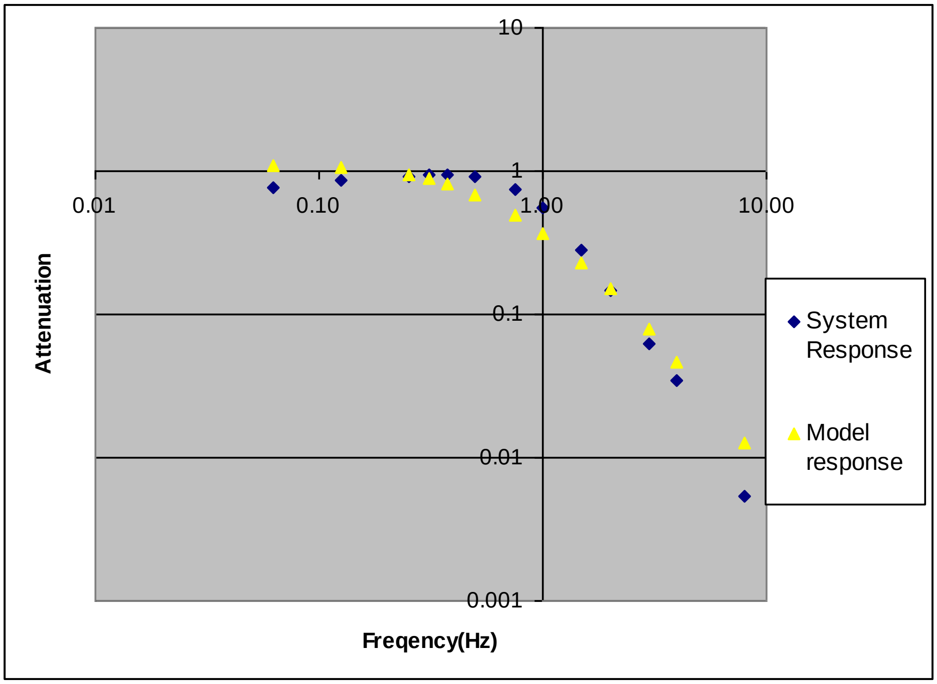

**Figure 14-2 Frequency response of loaded system vs. DC gain adjusted system model**

**Table 15-1 Frequency Response of Loaded System vs. DC Gain Adjusted Unloaded Model**

Frequency Response of System

| Freqency(Hz) | Response(V) | Adjusted Gain (V) | Input(V) | corrected inupt | estimated mag | % | DB |
|---|---|---|---|---|---|---|---|
| 0.06 | 37.4 | | 2 | 1.85 | 49.395 | 0.757162 | -2.41623 |
| 0.125 | 43 | | 2 | 1.85 | 49.395 | 0.870533 | -1.20429 |
| 0.25 | 45.5 | | 2 | 1.85 | 49.395 | 0.921146 | -0.71343 |
| 0.3125 | 46.4 | | 2 | 1.85 | 49.395 | 0.939366 | -0.5433 |
| 0.375 | 46.7 | | 2 | 1.85 | 49.395 | 0.94544 | -0.48732 |
| 0.5 | 44.9 | | 2 | 1.85 | 49.395 | 0.908999 | -0.82873 |
| 0.75 | 37.2 | | 2 | 1.85 | 49.395 | 0.753113 | -2.4628 |
| 1 | 27.6 | | 2 | 1.85 | 49.395 | 0.558761 | -5.05548 |
| 1.5 | 13.7 | | 2 | 1.85 | 49.395 | 0.277356 | -11.1392 |
| 2 | 7.2 | | 2 | 1.85 | 49.395 | 0.145764 | -16.727 |
| 3 | 4.8 | | 3 | 2.85 | 76.095 | 0.063079 | -24.0023 |
| 4 | 3.5 | | 4 | 3.85 | 102.795 | 0.034048 | -29.3581 |
| 8 | 0.7 | | 5 | 4.85 | 129.495 | 0.005406 | -45.3431 |
| Frequency Response of Gain Adjusted model | | | | | | | |
| 0.06 | 59.2 | 31.6128 | 1.2 | 1.07 | 28.569 | 1.106542 | 0.879358 |
| 0.125 | 57.5 | 30.705 | 1.2 | 1.07 | 28.569 | 1.074766 | 0.626281 |
| 0.25 | 50.65 | 27.0471 | 1.2 | 1.07 | 28.569 | 0.946729 | -0.47549 |
| 0.3125 | 47 | 25.098 | 1.2 | 1.07 | 28.569 | 0.878505 | -1.12512 |
| 0.375 | 43.4 | 23.1756 | 1.2 | 1.07 | 28.569 | 0.811215 | -1.81728 |
| 0.5 | 36.6 | 19.5444 | 1.2 | 1.07 | 28.569 | 0.684112 | -3.29745 |
| 0.75 | 26.5 | 14.151 | 1.2 | 1.07 | 28.569 | 0.495327 | -6.10216 |
| 1 | 19.9 | 10.6266 | 1.2 | 1.07 | 28.569 | 0.371963 | -8.59001 |
| 1.5 | 12.1 | 6.4614 | 1.2 | 1.07 | 28.569 | 0.226168 | -12.9114 |
| 2 | 8 | 4.272 | 1.2 | 1.07 | 28.569 | 0.149533 | -16.5053 |
| 3 | 4.2 | 2.2428 | 1.2 | 1.07 | 28.569 | 0.078505 | -22.1021 |
| 4 | 2.5 | 1.335 | 1.2 | 1.07 | 28.569 | 0.046729 | -26.6083 |
| 8 | 0.67 | 0.35778 | 1.2 | 1.07 | 28.569 | 0.012523 | -38.0456 |

After work on the PID controller was completed, the next step was designing the observer. The top level block diagram is shown in Figure 16-1. The controller used was the PID controller shown above with the added feedback path for disturbance rejection. The controller block diagram is shown in Figure 16-2. The observer uses Ellis's method of observer based control; the gain of the model of the plant was reduced in order to make the observer better over a wide variety of loads, but otherwise the observer used the linear model that was developed earlier. The observer block diagram is shown in Figure 17-1. The observer uses a parallel PI structure to determine the amount to modify the input by to help the model stay synchronized with the system. The advanced disturbance observer algorithm uses this value added to the controller output to drive the system to appear more like the model.

The specifications for the observer with this disturbance rejection algorithm are shown below.

- Phase Margin = 50 degrees

- Gain Margin = 3.5
- Steady state error < 1 degree
- Rise Time = 1.17 s
- % Overshoot = 3%

The time domain results were then recorded for the proportional, PID, observer, and observer with disturbance rejection controllers in the unloaded and loaded cases. These results are shown in Figure 17-2 and Figure 18-1 respectively.

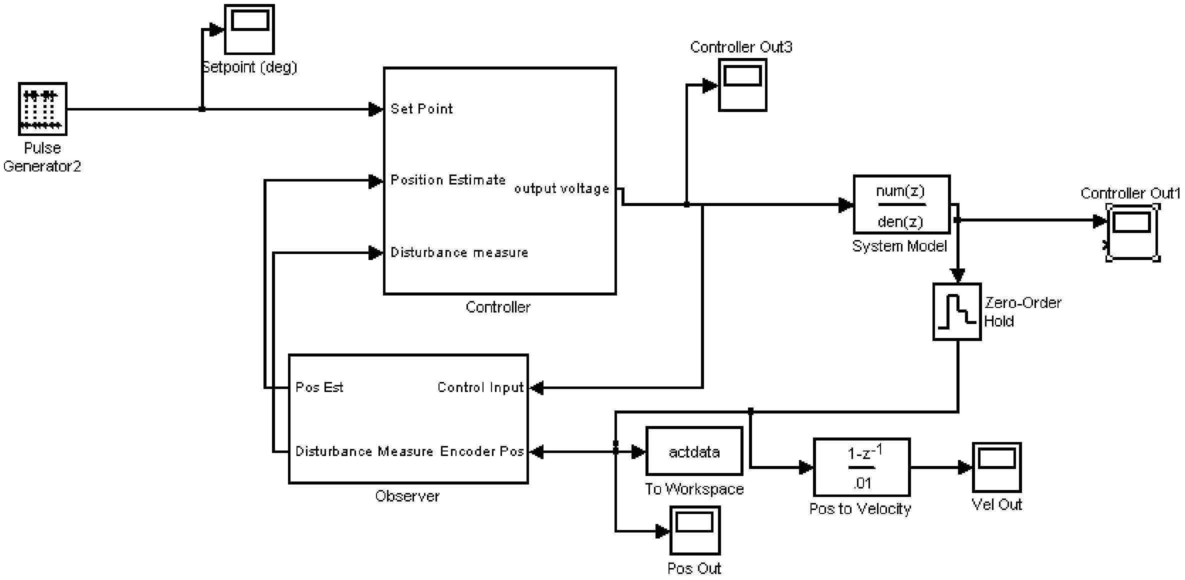

**Figure 16-1 Top Level Block Diagram for Observer Control**

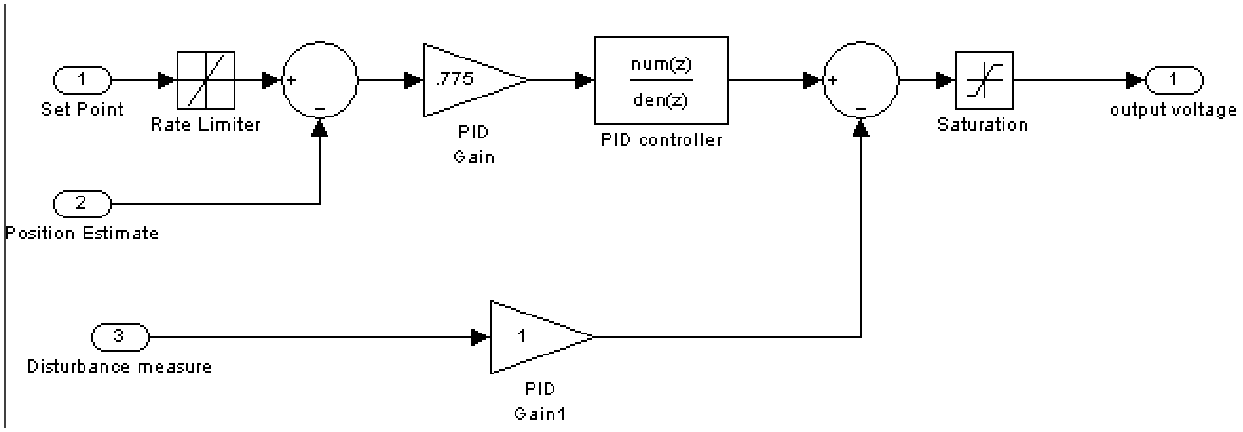

**Figure 16-2 Observer Controller Block Diagram**

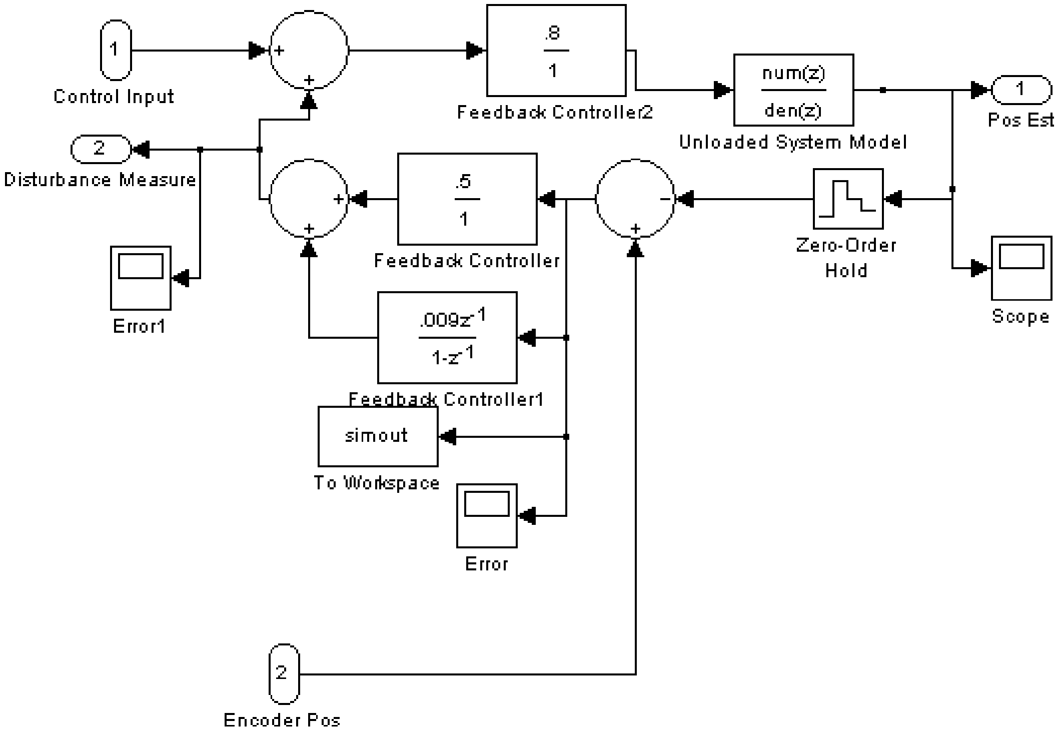

**Figure 17-1 Observer Block Diagram**

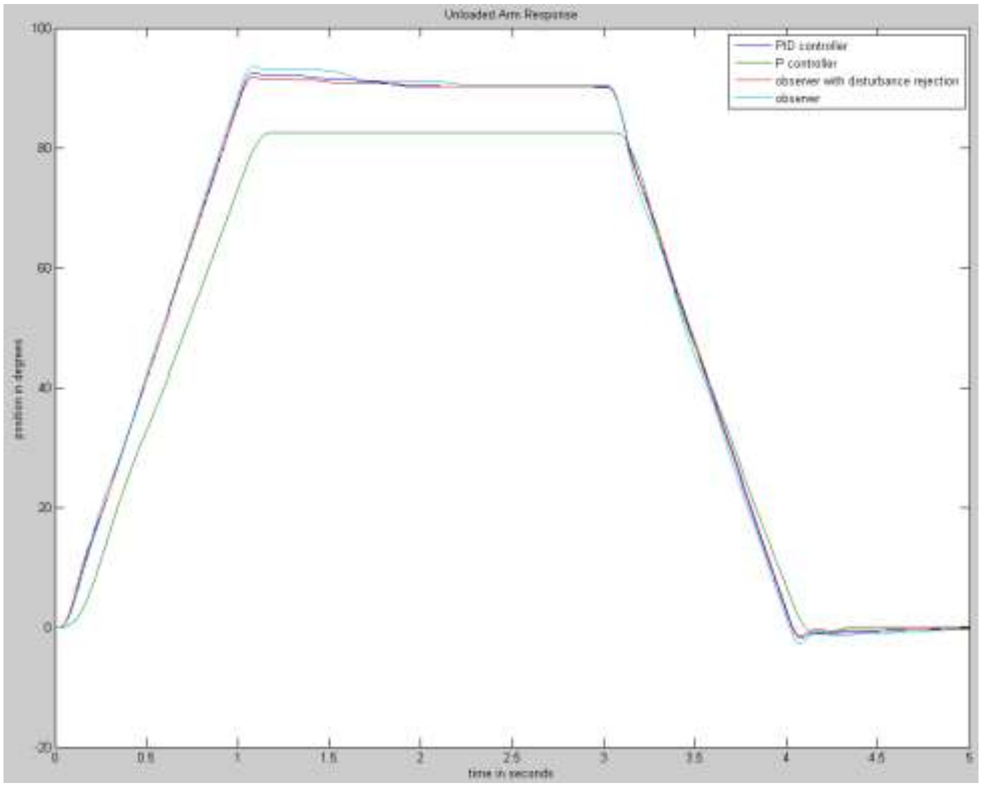

**Figure 17-2 Time Domain response of different controllers**

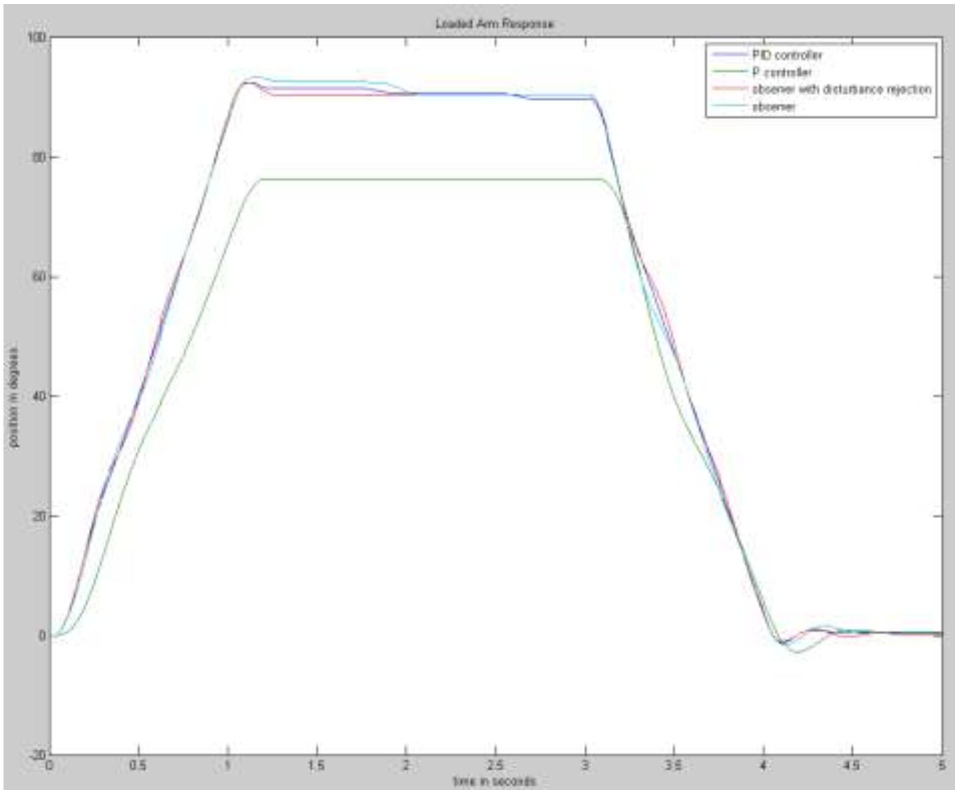

**Figure 18-1** Loaded Arm Results for Implemented Controllers

As shown in Figure 17-2 the observer without the disturbance rejection term is slightly slower, with higher overshoot than the PID controller without the observer. With the disturbance rejection though, the observer with PID controller performs slightly better than the PID controller by itself. In the loaded case on the upswing the same information applies, however on the downswing all of the models are very similar.

## Conclusion – Pendulum Arm

In the case of the pendulum arm the observer with disturbance rejection provides a slight improvement in settling time, overshoot and phase margin compared to the other tested methods. This improvement comes at the cost of computational complexity and engineering time. If computational complexity is at a premium or if the slight increase in performance is not critical this method of control is not recommended.

The observer without disturbance rejection does not provide any advantages to this system since the sensor on the robot arm is already precise enough to get a very good measure of the position of the arm. With worse sensor capabilities observer based control might provide more benefit.

# Completed Work for Level Arm Configuration

## System ID calculations

### Pole Locations

When performing system ID, the overshoot and settling time of the base was measured and the closed loop pole locations were calculated from that information. The overshoot was measured to be 24.2% and settling time was 0.78s. Using the percent overshoot equation, the damping ratio was calculated to be 0.41. From the settling time equation, the ζ*ωn factor was calculated to be 5 so the pole locations were calculated to be at the origin and at -10 rad/s in the s-domain. Next the poles associated with the spring action were identified. Using the displacement of the arm relative to the base, and the settling time and frequency of oscillation were measured. The settling time was measured to be 0.9s and frequency of oscillation was measured to be 2.7Hz. The settling time gives the ζ*ωn factor through the settling time equation and the 2.7Hz gives the imaginary part of the poles. That was found by multiplying the 2.7Hz by 2 pi. From these calculations, the locations of the spring poles were determined. These locations were calculated to be at -4 ± 17j.

### Model Development

It was decided to treat the spring motion as a disturbance to the arm base position. This allows the base transfer function to be the forward path of the plant and the spring transfer function to be a loop disturbance with positive feedback.

The horizontal 2-DOF robot arm underwent similar design stages to the pendulum configuration robot arm. The system identification was broken down into two steps. First the system was modeled without the effect of the springs and then modeled with the effect of the springs included. The two results were combined to build an accurate model of the system.

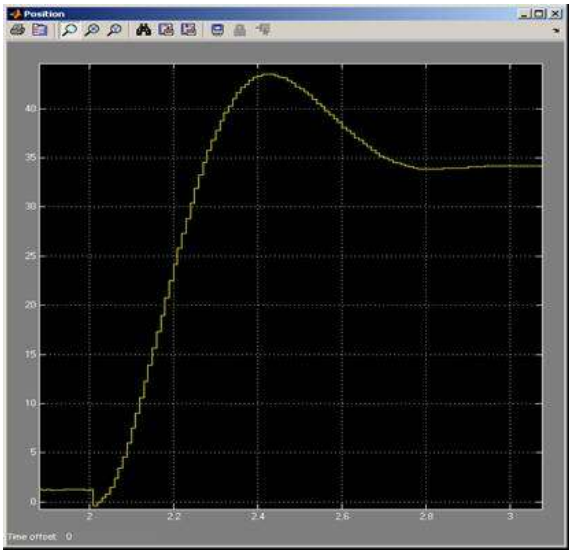

**Figure 19-1 Step Response of Horizontal Configuration Without Springs**

From the graph shown in Figure 19-1, the pole locations are determined to be 0 and -10 rad/s. The DC gain was found in Simulink to be 1500deg/s. This gives an overall transfer function of **Gp=1500/(s2+10s).** Next the springs were modeled in the same way; the step response of the system was measured and from the displacement of the arm relative to the base, the poles associated with the springs were determined. These turned out to be -4±17j. The DC gain was found in Simulink to be 0.42 giving an overall transfer function of GD=0.42s/(s2+8s+289). The spring disturbance was modeled as a minor loop disturbance because it continually affects the output until steady state. The block diagram of the entire system is shown in Figure 20-1 below.

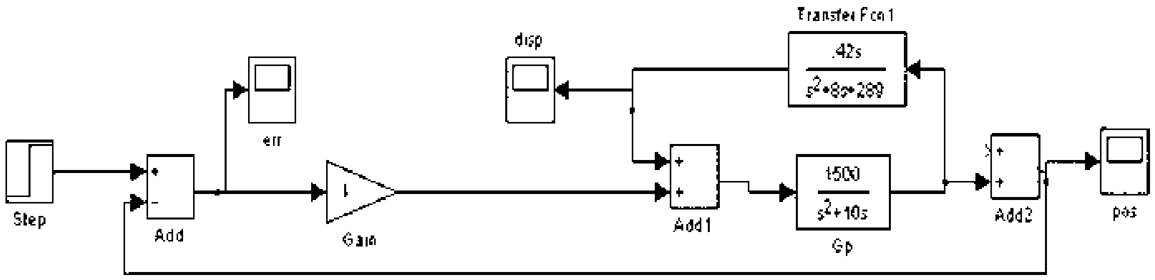

**Figure 20-1 Block Diagram for the Arm with Spring Disturbance Modeling**

After getting an accurate model of the plant, it was digitized in order to implement digital controllers for the plant. The digital model was then analyzed via root locus and the sample frequency was adjusted to give a better response. This sample period was found to be 0.1s. At this sample period, the phase margin specification as well as the %OS specification was met with just proportional control.

## Control Methods

### Proportional Controller

After determining the sample period, a system was created using only proportional control. The gain was tuned to the fastest possible settling time, which resulted in a gain of 0.0231, a settling time of 1.1 seconds, and an overshoot of 0.27%. The phase margin was measured in MATLAB to be 70.5 degrees, and the gain margin was 20.5 dB. The performance of our system was also measured with a load of 69.2g, and the results we obtained were as follows: OS = 6.3%, $T_s$ = 2s, $e_{ss}$ = 2.1 deg. The system with proportional control is shown in Figure 20-2.

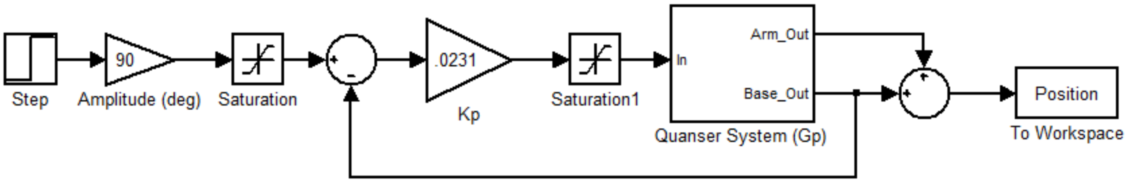

**Figure 20-2 System with Proportional Controller**

### PID Controller

A PID controller was designed where $K_p = 0.023$, $K_i = 0.01$, and $K_d = 0.01$. This achieved a settling time of 1.1 seconds with 0% overshoot, which was not much of an improvement over the proportional controller.

When a load was added to the system with the PID controller, the following results were obtained: OS = 4.2%, $T_s = 1.9s$. The loaded testing yielded slightly better results than with just proportional control. The PID control system is shown in Figure 21-1.

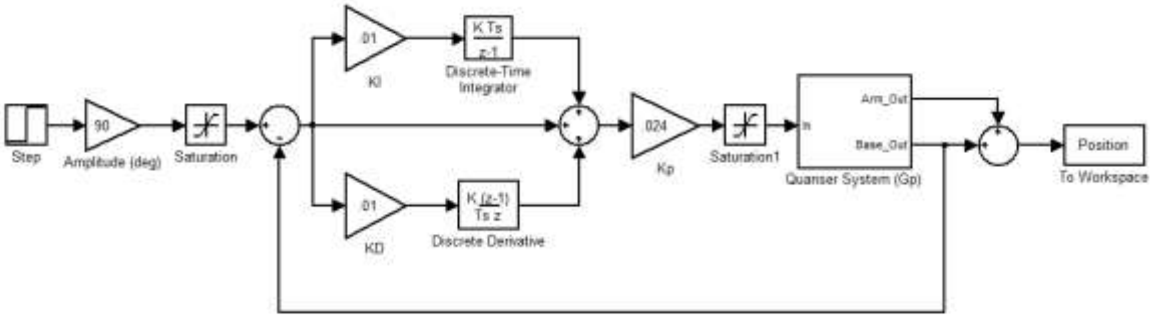

**Figure 21-1 System with PID Controller**

### Lead-Network Controller

A lead-network controller was designed which cancelled out a pole at z = 0.458 and changed $K_p$ to 0.048. The system had a settling time of 0.9 seconds with no overshoot. The loaded system had a settling time of 1.1 seconds with an overshoot of 0.4%, which was a huge improvement over the other loaded systems. The system with the lead-network controller is shown in Figure 21-2.

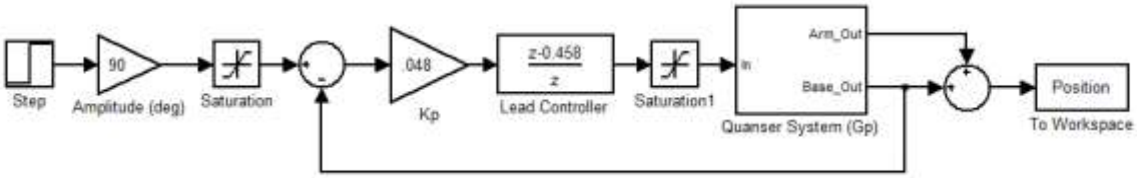

**Figure 21-2 System with Lead-Network Controller**

### Minor-Loop Control

The next classical control method designed was minor-loop control. The parameters were $K_p = 6.0$ and $K_{ml} = 0.01$. A settling time of 1.4 seconds with 15.5% overshoot was obtained. Experimentation found that adding a PI controller with the minor-loop control, as shown in Figure 22-1, yielded much better results, with a settling time of 1.0 seconds and an overshoot of 7%. When the load was added, the system had a settling time of 1.4 seconds with 10% overshoot.

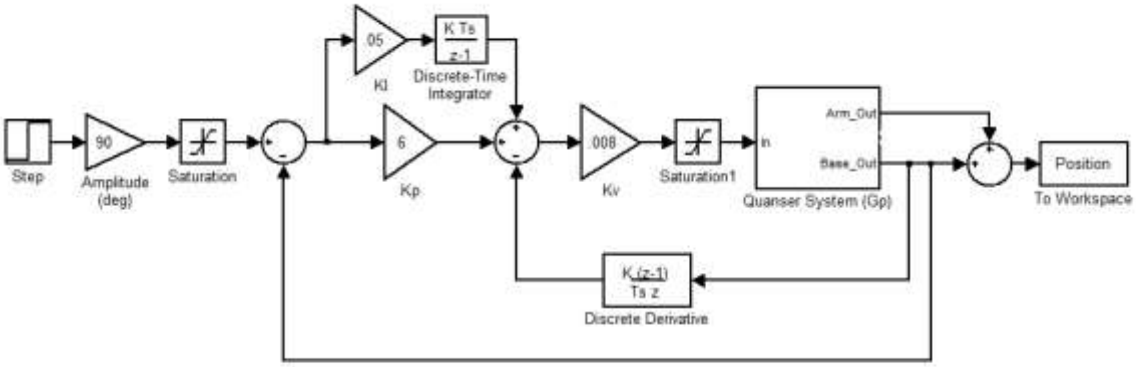

**Figure 22-1 System with Minor Loop Controller**

### Observer-Based Control

The observer was based on George Ellis's model, shown in Figure 22-2. The $G_C(s)$ is the same lead controller that was used in the lead-network design, with a gain of 0.2. The values for $G_{PC}(s)$, $G_S(s)$, and $G_{SEst}(s)$ are all 1.0, since they have virtually no effect on the system. $G_{CO}(s)$ is 0.01, which is the highest possible value with a stable system. $G_{PEst}(s)$ is the model of the plant base and was tuned by setting $G_{CO}(s)$ equal to 0, then ignoring the output drift and adjusting the DC gain until a reasonable output around 90 degrees was obtained, as shown in Figure 23-1.

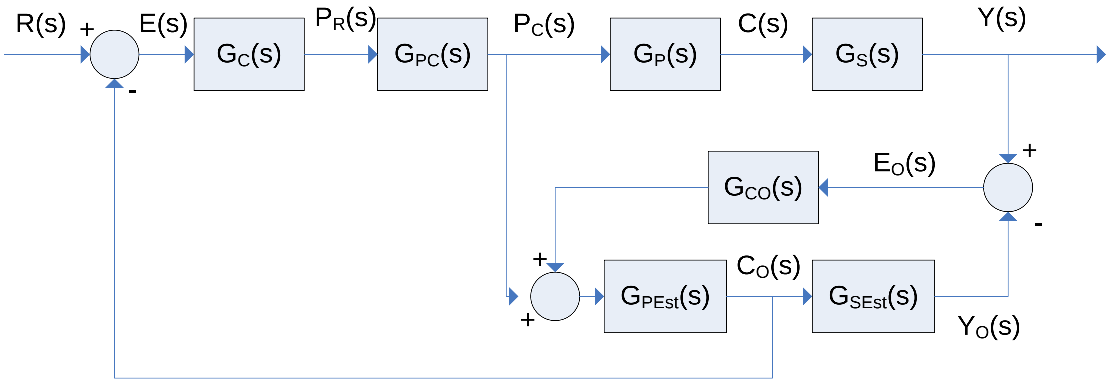

**Figure 22-2 George Ellis's Observer Method**

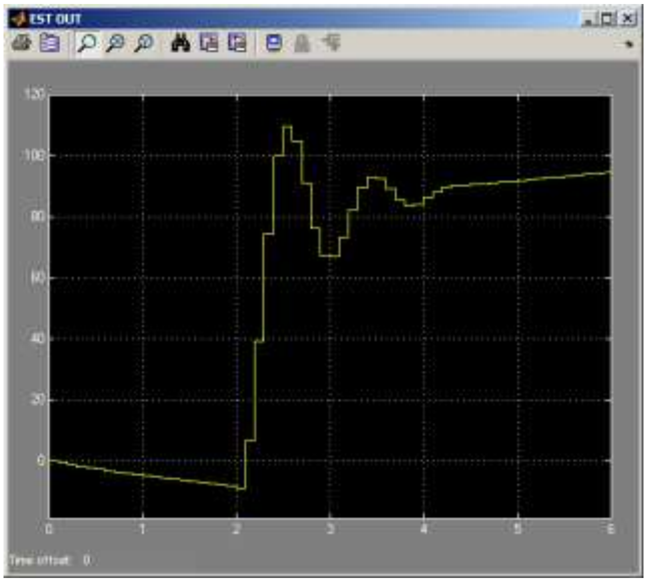

**Figure 23-1 Drifting Output of the Observer Estimator**

The observer gave a settling time of 0.9 seconds with 0% overshoot, which was a very impressive system. However, it could not handle a disturbance, and if the arm was physically moved from the final point, it would not correct itself. A lead controller was added in the observer path to speed up the disturbance rejection, but it did not make a significant difference. The issue was finally fixed by feeding the observer controller directly to the plant, as shown in Figure 23-2. A gain of $K_{DD}$=2 was added to the error signal, and the gain and controller in the observer were adjusted to give the fastest possible system. The final observer had a settling time of 1.2 seconds with an overshoot of 9% without a load, and a settling time of 1.4 seconds with an overshoot of 12% with a load applied.

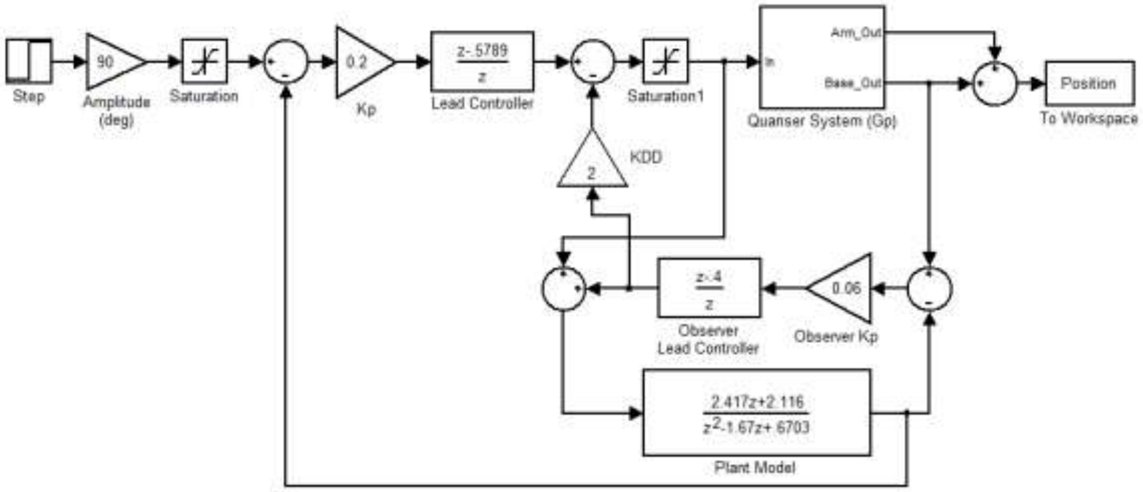

**Figure 23-2 System with Observer-based Controller**

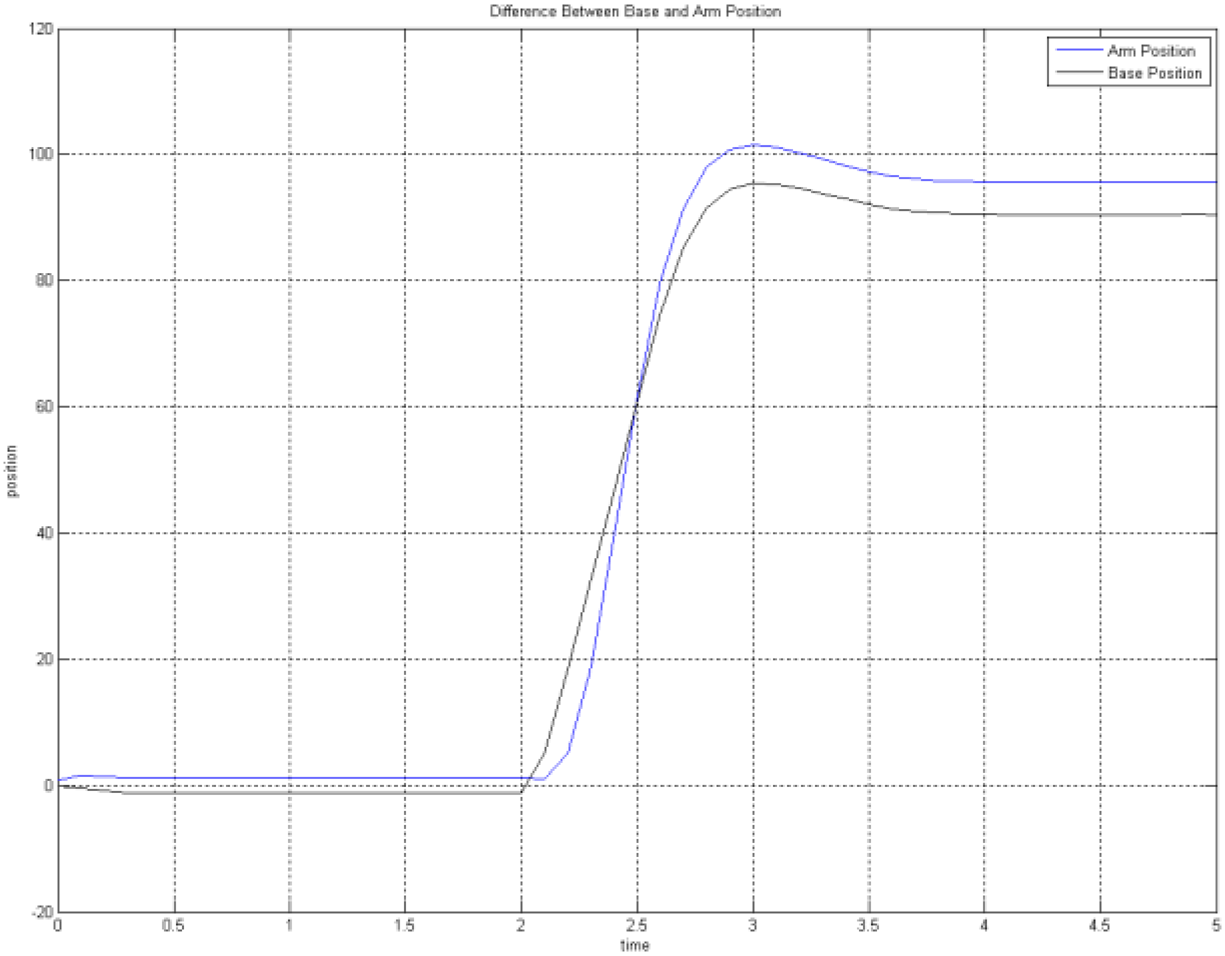

**Figure 24-1 Output of Base Position and Arm Position**

As shown by Figure 24-1, there are inconsistent measurements due to the springs in the system. The arm and the base of the system will appear to be at different places, sometimes over a 5-degree difference. With this, it is nearly impossible to meet the initial specification of a steady-state error less than 2 degrees. This is also the reason why the results in Figures 25-1 and 25-2 have a wide range of final values, especially after a load is applied to the system.

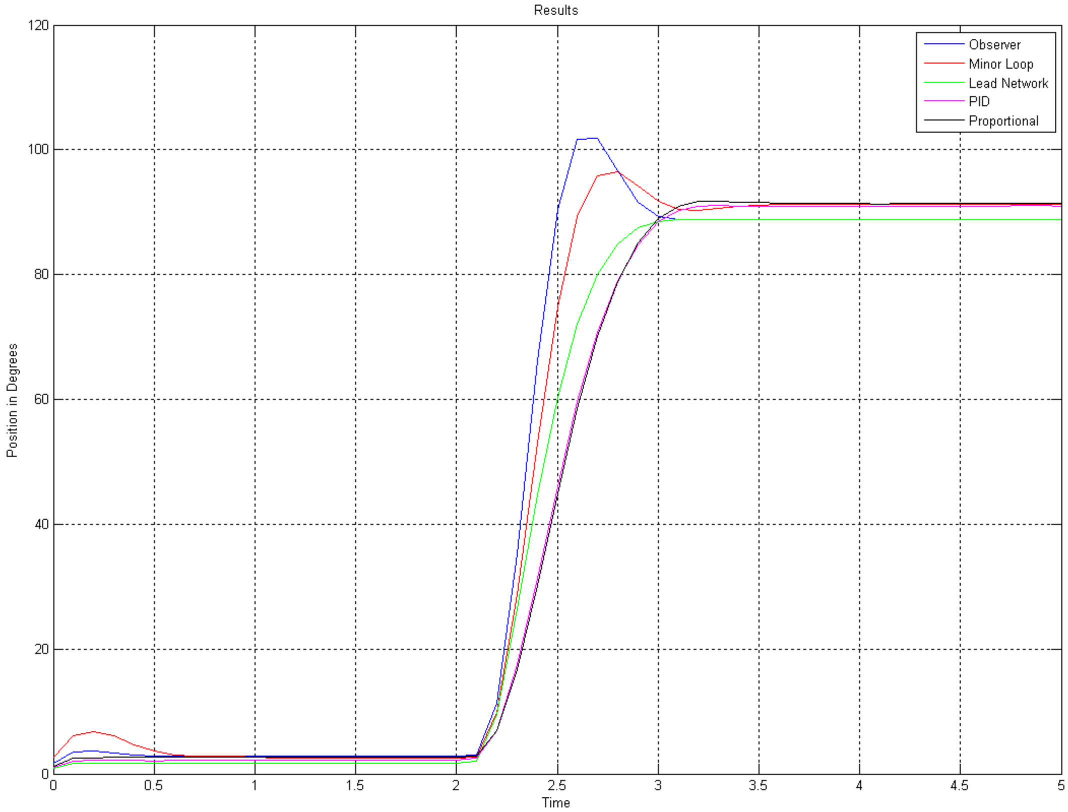

**Figure 25-1 Results of Unloaded Systems with Different Controllers**

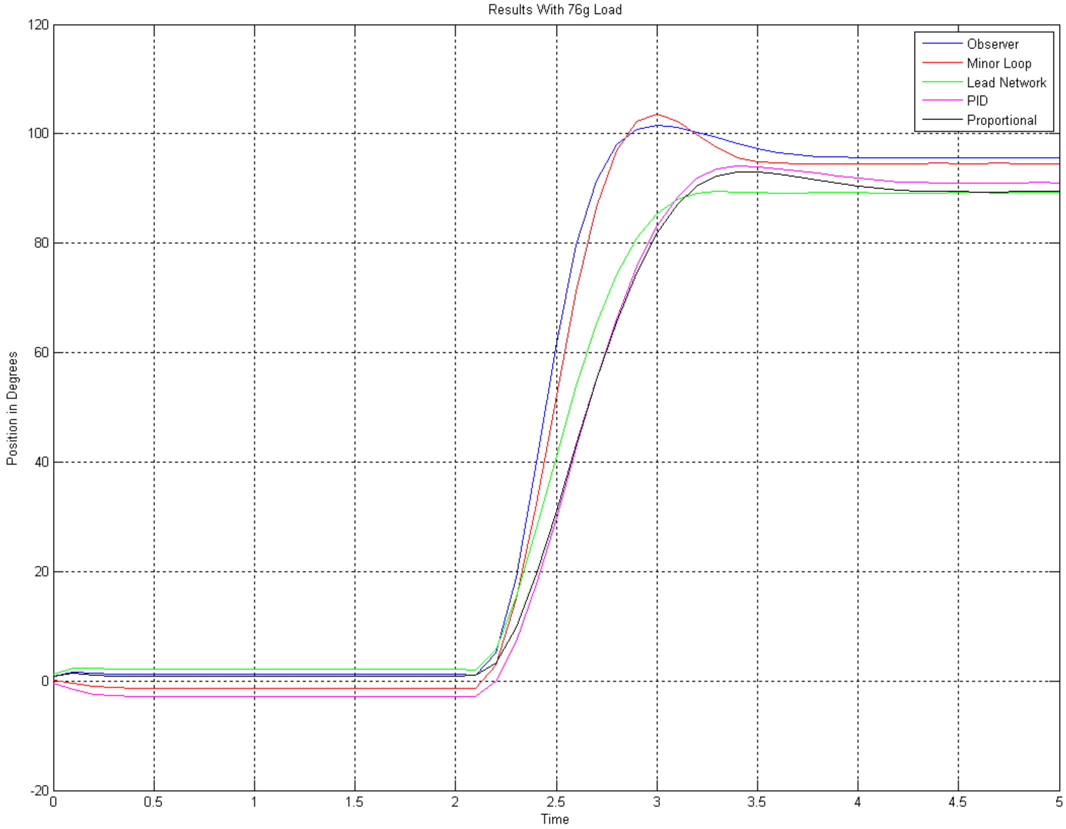

**Figure 25-2 Results of Systems with 76.6g Load Attached**

## **Conclusion – 2-DOF Arm**

The Ellis method of observer-based control provided both advantages and disadvantages compared to classical control methods. The results are shown in Figures 25-1 and 25-2. For the 2-DOF robot arm, proportional control was inadequate for the desired system. The results varied greatly between loaded and unloaded testing, and the settling time was slow and the steady-state error was large. The PID controller provided a slight improvement on the system, both loaded and unloaded. However, the improvement was not significant, and although the system met the specifications, a much faster system was preferred.

The minor-loop controller yielded great results, but perhaps the best classical control method was the lead-network controller. It produced fast settling times with little to no overshoot for both loaded and unloaded systems. It seemed to perform even better than the system with observer-based control. The observer provided the fastest response without disturbance rejection; however, a system with disturbance rejection was more desirable. After adding disturbance rejection, the observer still worked well, as it had a settling time that was quicker than the PID controller, and the system was not affected much by a load. However, despite its decent speed and its powerful disturbance rejection, the system’s overshoot rose higher than the rest of the controller methods, and it was still not the fastest system.

Overall, the Ellis method of observer-based control proved to be very effective when disturbance rejection was ignored, but only moderately effective when disturbance rejection was included. Unfortunately, the springs in our system cause the steady-state error to often be outside of the specification, which is a problem that we are unable to correct. However, if the error from the springs is ignored, the observer causes the steady-state error to be very close to zero, which proves that it is an accurate, reliable controller method for the robot arm.

# Bibliography

[1] G. Ellis, *Observers in Control Systems*, San Diego, CA: Academic Press, 2002.
[2] G. Dempsey, *EE 432- Control System Theory Lecture Workbook*, Peoria, IL: Unpublished, 2011.
[3] A. Fouts and K. Liggett, “Observer-based Engine Cooling Control System (OBCOOL) Final Report,” B.U., Peoria, IL, 2011.