

# WEBWIND

## Web Controlled Wind Tunnel System

Final Project Report

May 11, 2010

By:

Ben Morrison

Mike Firman

Advisors:

Dr. Aleksander Malinowski

Dr. Scott Post

# Web Controlled Wind Tunnel System

## Contents

Abstract.....	2
Project Summary.....	3
Introduction .....	3
Wind Tunnel Description .....	3
Power Up and Speed Control.....	3
Testing Bed and Linear Actuators .....	3
Sensors.....	4
Overall System Description.....	4
Wind Tunnel, Webcams, and Router .....	4
Microcontroller, GUI, and Sensor Data.....	5
Linear Actuator Position Control and Wind Speed Controller.....	5
Progress.....	5
Linear Actuator Model .....	5
Closed Loop Model Description .....	7
Signal Processing Block .....	8
Gain .....	8
PWM Subsystem .....	8
H-Bridge Subsystem .....	9
Model Input .....	9
Simulation/Gain Tuning .....	9
Simulation After Tuning .....	12
H-Bridge Circuitry.....	14
Suggested Future Work .....	15

# Web Controlled Wind Tunnel System

## **Abstract**

The Bradley University wind tunnel system was upgraded to make it more modern. The wind tunnel utilizes a variety of sensors, actuators, and manual switches. The system is being updated from dated manual control to more accurate computer aided control. The system is also to be updated in such a way that the entire wind tunnel system can be accessed via the internet.

# Web Controlled Wind Tunnel System

## **Project Summary**

### **Introduction**

Currently, the Bradley University Department of Mechanical Engineering has a small research wind tunnel. The original system was completely manually controlled and operated. Since all functions needed to be adjusted by hand and there were no digital control mechanisms in place, it was very difficult to get accurate, repeatable results. In addition, this also increased the amount of time required to take measurements with the wind tunnel, and increased the work load for the operator. The mechanical engineering department also requested adding the ability to operate the wind tunnel system remotely. This not only allows for easier data acquisition onto a personal computer or laptop for later analysis, but allows other institutions which do not have access to a wind tunnel the educational experience of operating and collecting data from a real world wind tunnel.

To accomplish this task we updated the current wind tunnel platform with an embedded microcontroller capable of implementing a simple closed-loop controller which allowed for precise actuation of the wind tunnel components. This embedded microcontroller also collects data from the various sensors and streams it to a host computer, which is running a webserver. Clients will connect through a Java interface and be able to view the current state of the wind tunnel and operate all functions as if they were in the room with the wind tunnel.

## **Wind Tunnel Description**

### **Power Up and Speed Control**

The first major component of the wind tunnel is the blower turbine. This blower turbine is what causes the air flow. This turbine runs at a constant speed once engaged. To change the airspeed inside the system a damper is opened or closed via a universal motor. This universal motor contains a field winding and an armature winding, bidirectionality is achieved by changing the position of the connections of these windings.

### **Testing Bed and Linear Actuators**

The testing bed is the area where the component on which the current experiment is being run is located. This area currently contains a single Firghelli Automation linear actuator used to change the angle of the component. The angle is set via visual alignment to a protractor taped to the back wall of the test bed. Eventually an additional Firghelli Automation linear actuator will be added which is to have a hotwire wind speed sensor attached to it. This actuator/sensor combination will allow for data to be taken at various points above and below the component being tested.

# Web Controlled Wind Tunnel System

## Sensors

The wind tunnel system has a variety of sensors which need to be interfaced to the microcontroller. The test bed contains a lift sensor and a drag sensor. These two sensors are load cells which output a low voltage, bipolar signal. Each of the linear actuators contains a built in potentiometer which gives feedback on the current position of the arm. There is also a pressure transducer in place which gives the airspeed inside the wind tunnel on a zero to ten volt range.

## Overall System Description

The overall system is comprised of eight major parts: the wind tunnel, webcams, router, microcontroller, graphical user interface (GUI), sensor data, linear actuator position control and the wind speed controller. This section will take a deeper look at each of these systems and how they work together. A simple block diagram is shown below.

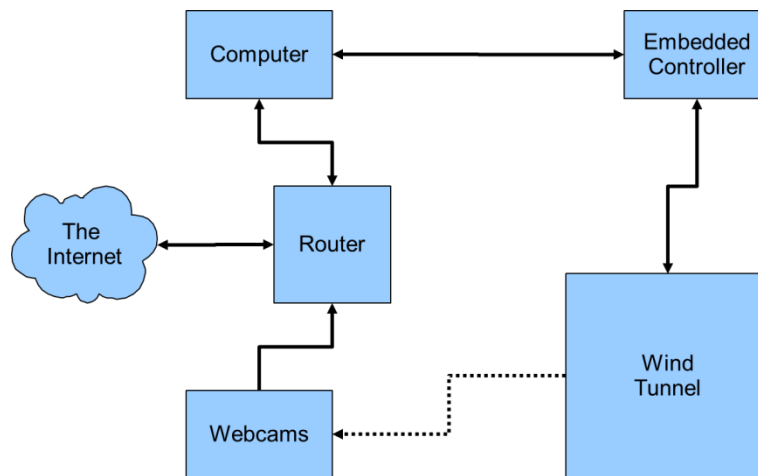


Figure 1. System Block Diagram

## Wind Tunnel, Webcams, and Router

The wind tunnel section of the overall system refers to all components which are used to operate the wind tunnel currently. The wind tunnel operates via a large constant speed turbine connected to a damper which is used to control the airspeed. This damper is opened or closed with a universal motor to increase and decrease the speed in the test bed. The test bed contains a linear actuator, used to adjust the angle of the object being tested, two load cells, used to measure lift and drag, and a protractor on the back wall of the test bed to set the angle. Finally, there is a pressure transducer which is used to measure the airspeed in the test bed.

The webcams, D-Link DCS-910s, are situated around the wind tunnel system. These will be placed to allow the user to see how the part being tested is performing in the wind tunnel as well as allow them to see the status of the surrounding area of the wind tunnel. The webcam handling will be modular such that more webcams can be added to the system without any reconfiguration. They will merely need to be plugged into the router beside the current three webcams.

# Web Controlled Wind Tunnel System

The router, a Netgear FVS-318 VPN, will enable the webcams and computer host to be internet connected. It will also handle the port forwarding necessary to allow the multiple camera streams to be embedded into the web-interface page.

## **Microcontroller, GUI, and Sensor Data**

To be filled in by Mike later.

## **Linear Actuator Position Control and Wind Speed Controller**

Another major section of the overall system is the linear actuator position controller. The linear actuators are controlled via a closed loop system implemented on the SI-Labs F-120 microcontroller explained earlier. The actuator controller is a PI, or proportional and integral, control scheme. More detail on the development of this controller will be given later.

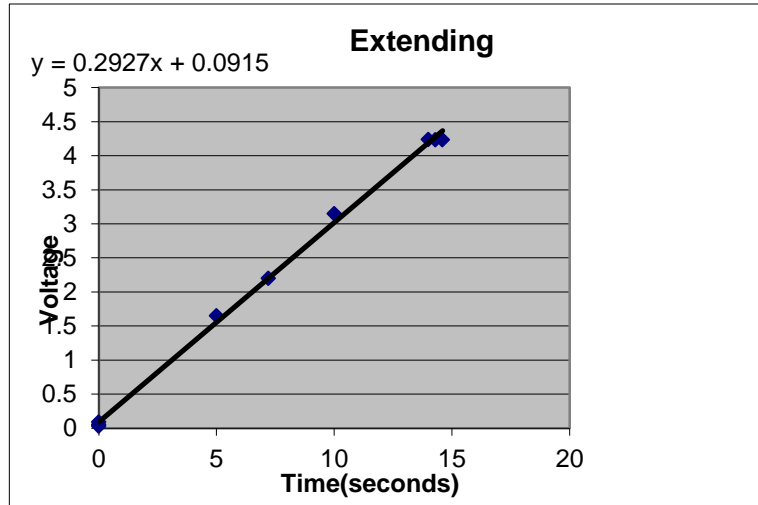
Lastly, the wind speed controller was developed to alter the airspeed in the testing bed. The airspeed is changed, as described earlier, by opening and closing a damper which is controlled by a universal motor. This motor is a 120 volt A/C motor with separate armature and field windings. The connection of these windings determines which direction the motor turns. The control that was decided on for this system is an open loop method which uses a solid state relay to give the 120 volt power and a double pole double throw(DPDT) relay to form the armature and field connections, thus retaining the bidirectionality of the system.

## **Progress**

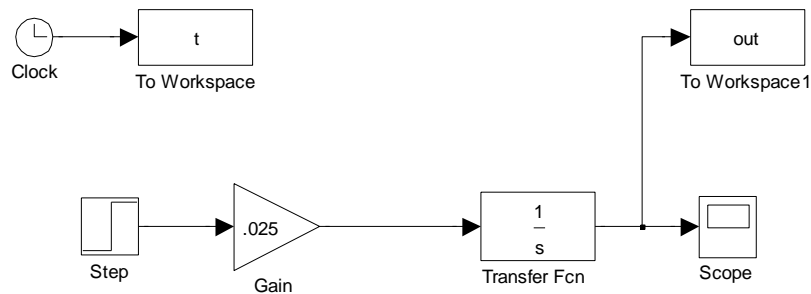
### **Linear Actuator Model**

In order to create a model for the closed loop system in order to do testing before implementing the controller in hardware the first step was to make a model for the linear actuator. Data was taken by applying a 12 volt step input to the actuator and recording the position of the arm at time intervals by measuring the current voltage of the wiper on the potentiometer. The results of this plot can be seen in the figure below.

## Web Controlled Wind Tunnel System

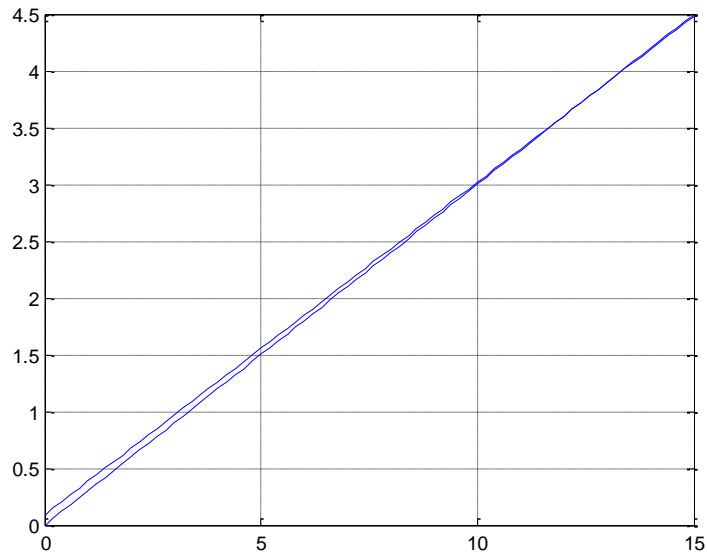


This plot shows a best fit line from three test runs. The best fit line,  $y=0.2927x+0.0915$ , was used to test the accuracy of the model later. This plot clearly shows that the linear actuator acts as an integrator: a step input gives a ramp output. The Simulink model for the actuator therefore is an integrator with a gain. Using gain tuning to achieve a result that closely matches the experiment data a gain value of 0.025 was chosen to represent the actuator with a 5 volt signal across the potentiometer. The Simulink model of the actuator is shown in the figure below.



Notice the clock with goes to workspace as well as the final output. This was done to be able to plot time vs output in MATLAB. By using MATLAB it was possible to obtain a graph with both the model output and the best fit line on the same plot. This was done to allow for an easy comparison of the Simulink model with the experimental results. This combination plot is shown in the figure below. This plot was obtained using the command `plot(t,out)` in the MATLAB command prompt after running the simulation. The X-axis is time and the Y-axis is voltage.

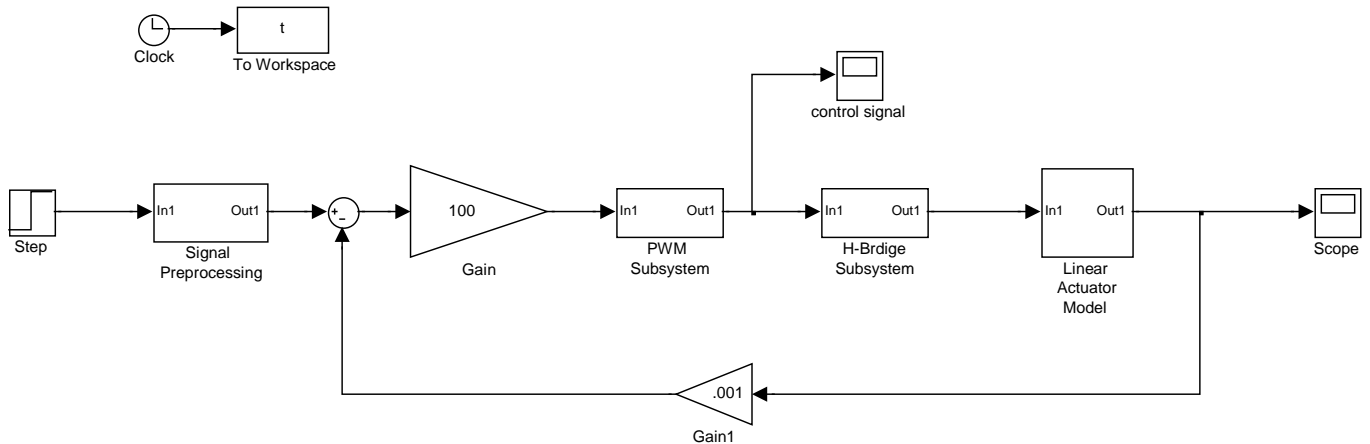
# Web Controlled Wind Tunnel System



There is a slight discrepancy between the two plots at the beginning, but the largest discrepancy is less than 0.1 volts. Based on these results it was determined the Simulink model of the linear actuator was sufficient for the project.

## Closed Loop Model Description

The next step was to continue working on the Simulink model. The loop was closed models for the PWM system and the H-Bridge. These models were developed from an earlier example from Dr. Dempsey. Below in Figure 22-1 the complete block diagram can be seen. Each block will be expanded later with a description of how it was derived. At this point a sample time of 1ms is being used. The gain values and interrupt time will most likely need to be tuned later. The feedback gain is set to the sample period.





# Web Controlled Wind Tunnel System

Figure 22-1 Simulink Model of Closed Loop System

## Signal Processing Block

The signal processing block is used to adjust the magnitude of the signal based on sample times and gains. This block will make the input to the Simulink model match the system input more closely. Currently only the sample time gain is included in this block. The contents of the Signal Processing block are shown below in Figure 22-2.

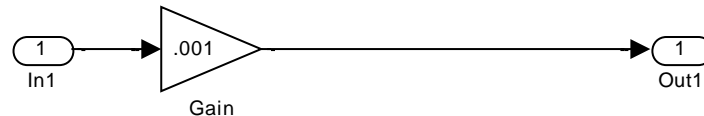


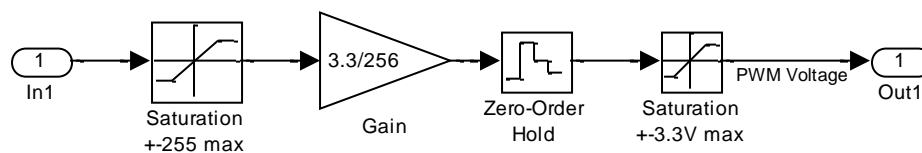
Figure 22-2 Signal Preprocessing Block

## Gain

Inside the loop the next block is the gain block. Since the system plant (the actuator) is an integrator PI control is obtained simply by adjusting the gain. This Gain block is the gain value of the controller. This was tuned to give an appropriate settling time for the system.

## PWM Subsystem

The PWM Subsystem was adapted from a file from Dr. Dempsey used in EE 450, and can be seen in Figure 5-1 below. The first saturation block is to model an 8 bit PWM, which is why the bounds are +255 and -255, this assumes unsigned numbers. The next gain block models the conversion to voltage steps, 256 being  $2^8$  and the 3.3 being the max pin output voltage of the microcontroller. The next block is the Zero-Order Hold, or ZOH. The ZOH is part of a D/A converter model, it will hold the value between interrupts. The final block is another saturation block. This block is not required for a functioning model, but increases the accuracy of the model. Because this block feeds the H-Bridge which will be bidirectional the 3.3V max output is modeled as both +3.3 and -3.3 volts.



# Web Controlled Wind Tunnel System

Figure 5-1 PWM Subsystem Block

The PWM voltage is read by a scope at this point, immediately following the PWM Subsystem block. The signal on the scope here will be the "Control Voltage."

## *H-Bridge Subsystem*

The H-Bridge Subsystem was also adapted from Dr. Dempsey's EE 450 course. The model used here can be seen in Figure 6-1 below. The H-Bridge gain block models the step up from the 3.3 volt microcontroller output to the 12 volt H-Bridge output (i.e.  $12/3.3=3.6364$ ). The saturation block used here models the H-Bridge limits of +12 and -12 volts, thereby also modeling the bidirectionality.

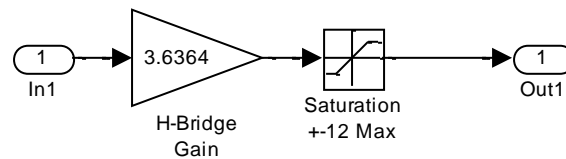


Figure 6-1 H-Bridge Subsystem Block

## *Model Input*

The input to the system will be a step function. The magnitude of this step will model the voltage difference across the actuator's potentiometer. In other words, a step with a magnitude of 2.5 signifies moving the full range of motion of the actuator (2.5 volts across the potentiometer). Typical angle adjustments are more likely to be in the tenths of volts range.

## *Simulation/Gain Tuning*

Using a step input with a magnitude of 2 (nearly full range of motion, around 13 seconds to 4 volts (80% of peak value, 5 volts) in Figure 20-2) and the gain value of 100 the following scope plot(Figure 24-1) is obtained from the output, only increasing to .003 volts in 25 seconds. The controller gain needs to be increased dramatically.

## Web Controlled Wind Tunnel System

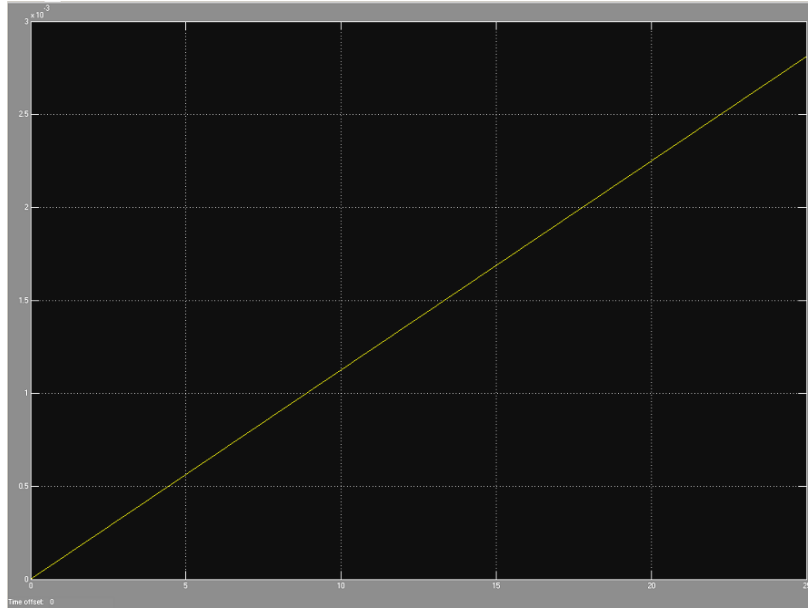


Figure 23-1 Output From 2V Input, Gain=100

Increasing the value of the controller gain to 10000000 gives a reasonable output, shown below in Figure 23-2 (note: plot was obtained by using the sink to workspace Simulink block, then using the command `plot(t,out)` in the MATLAB console). With the gain at 10000000 the final value is reached in roughly 19 seconds, a much more acceptable time.

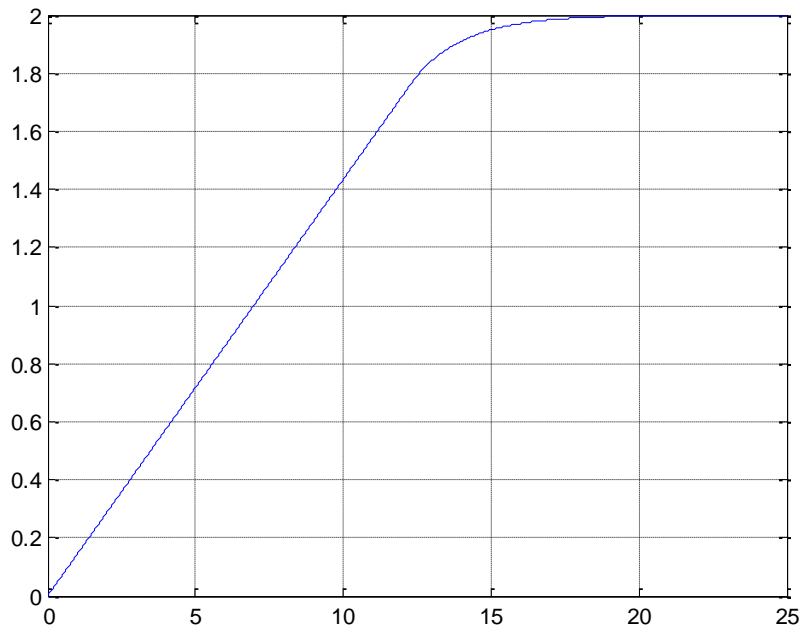


Figure 23-2 Output From 2V Input Gain=10000000

# Web Controlled Wind Tunnel System

While having the gain value at 10000000 results in steady state being reached in an acceptable amount of time the gain value is too high to code. Using block diagram algebra reduced the controller gain was reduced by dividing by an amount, then moving the amount which was divided to the signal preprocessing and the feedback loop. In order to decrease the gain value further the sampling time was increased to .1 seconds from .001 seconds. The resulting Simulink model after tuning is shown below in Figure 24-1.

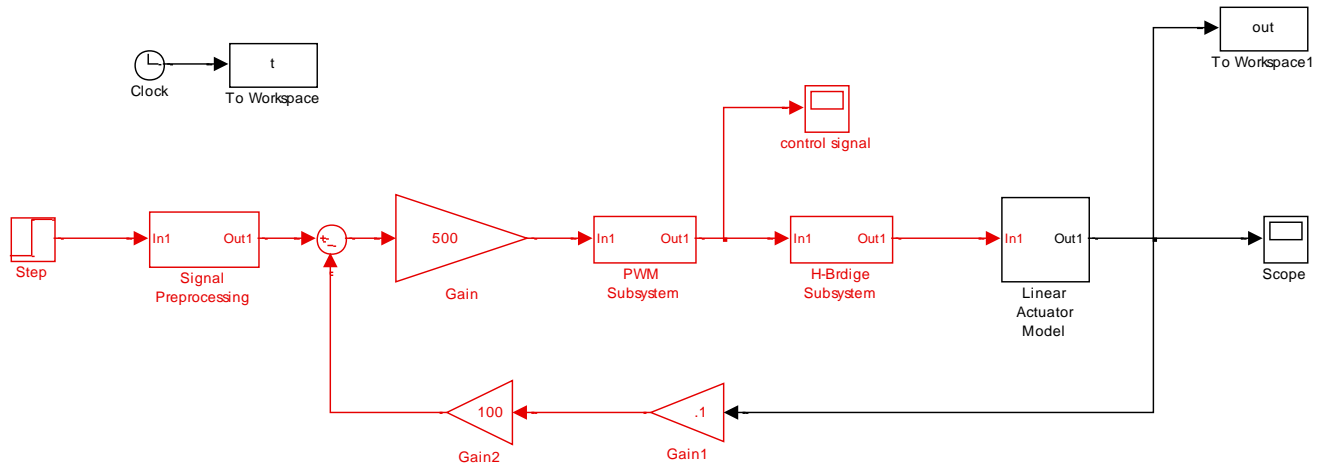


Figure 24-1 Final Block Diagram

The decreased controller gain, to 500, and the addition of the feedback gain, of 100, are visible in Figure 24-1. The simulated sampling gain has also been changed from .001 to .1. The new Signal Preprocessing block is shown below in Figure 24-2. Note the addition of the gain value of 100, relating to the conditioning of the controller gain.



Figure 24-2 Signal Preprocessing Block

# Web Controlled Wind Tunnel System

## *Simulation After Tuning*

Using the previous step input magnitude of 2 volts the following output plot (Figure 25-1) is obtained. This figure shows a settling time of 14 seconds. This can be compared to Figure 20-2 and Table 191 which shows full range achieved in approximately 14 seconds open loop.

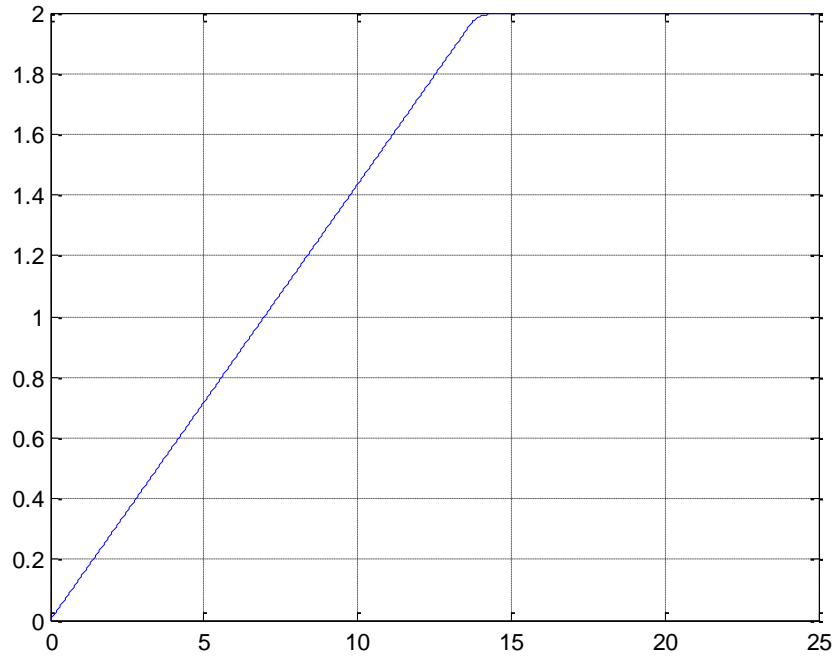


Figure 25-1: Output for 2V Input With Feedback Gain

The following plot (Figure 25-2) shows the command signal, i.e. the PWM output. From this plot it is clear that the majority of the time the PWM will be running at 100% duty cycle, outputting the full 3.3 V. The PI control effect can be seen from 13-16 seconds where the PWM is no longer running at max speed.

## Web Controlled Wind Tunnel System

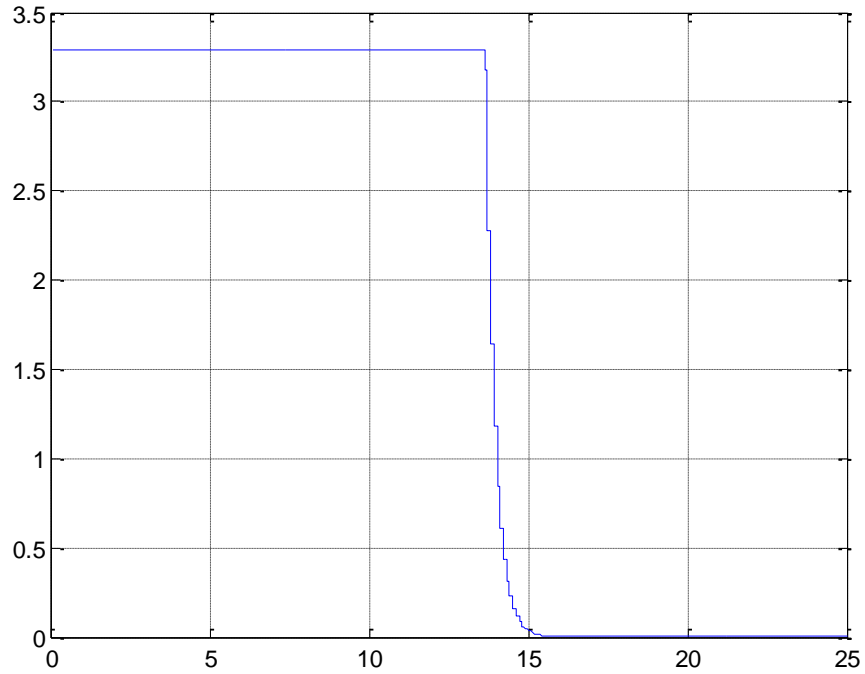


Figure 25-2: Control Voltage for 2V Input With Feedback Gain

Changing the step input magnitude to a more typical value of 0.2 volts gives the following output (Figure 26-1) and command (Figure 26-2) plots. The settling time for this input is around 3 seconds. The control is only at max PWM for a little more than 1 second

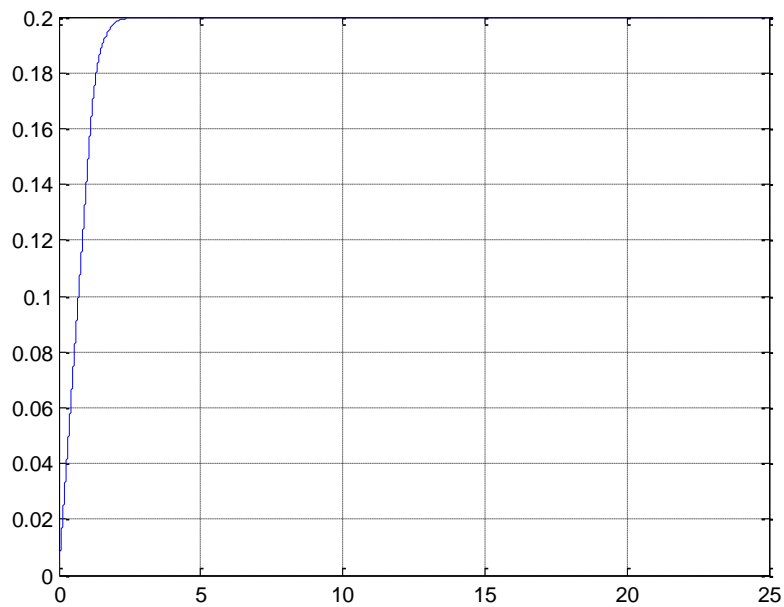


Figure 26-1 Output Voltage 0.2V Input With Feedback Gain

# Web Controlled Wind Tunnel System

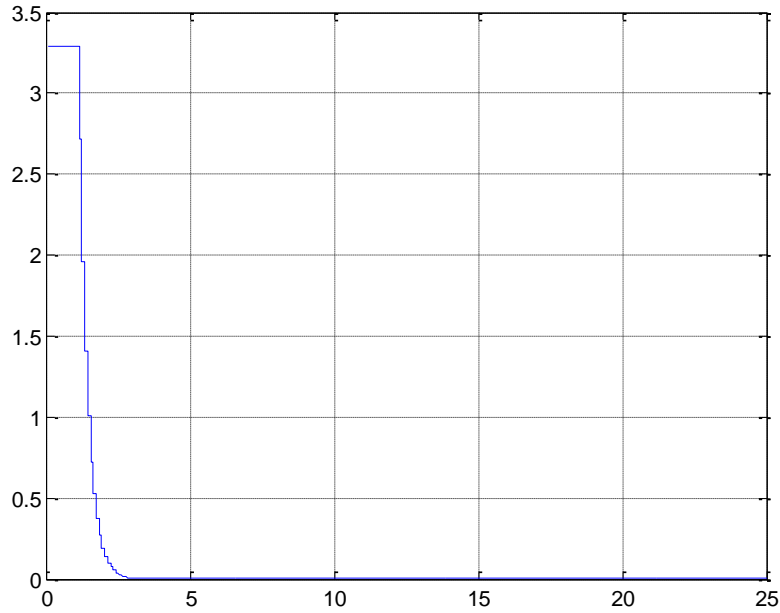


Figure 26-2 Control Voltage 0.2V Input with Feedback Gain

## H-Bridge Circuitry

After confirming it was possible to receive results consistent with the project demands from the Simulink simulations the next step was to build the H-Bridge circuitry which was used to move the linear actuator. It was decided to use PWM from the microcontroller with a port pin to control the direction input on the H-Bridge. The National LMD18200 H-Bridge was chosen for this application because of familiarity and previous experiments with it. The pin out and circuit diagram are shown in the two figures below.

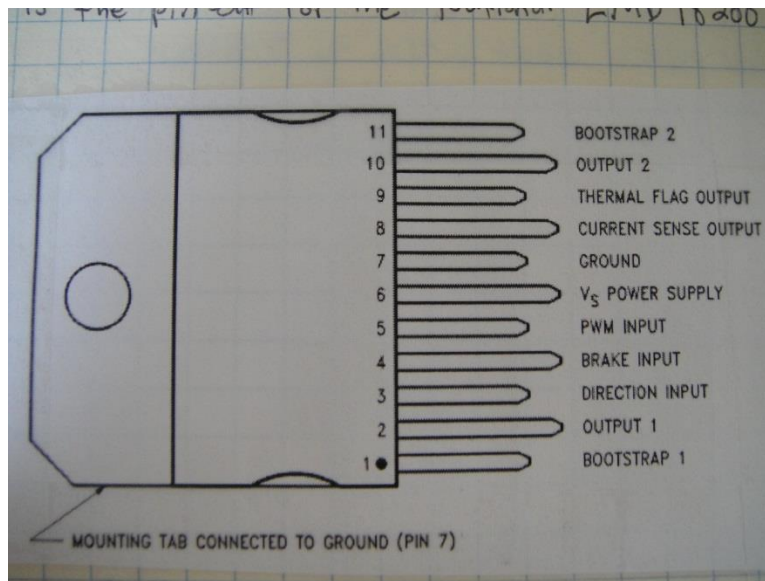


Figure X. H-Bridge Pinout

## Web Controlled Wind Tunnel System

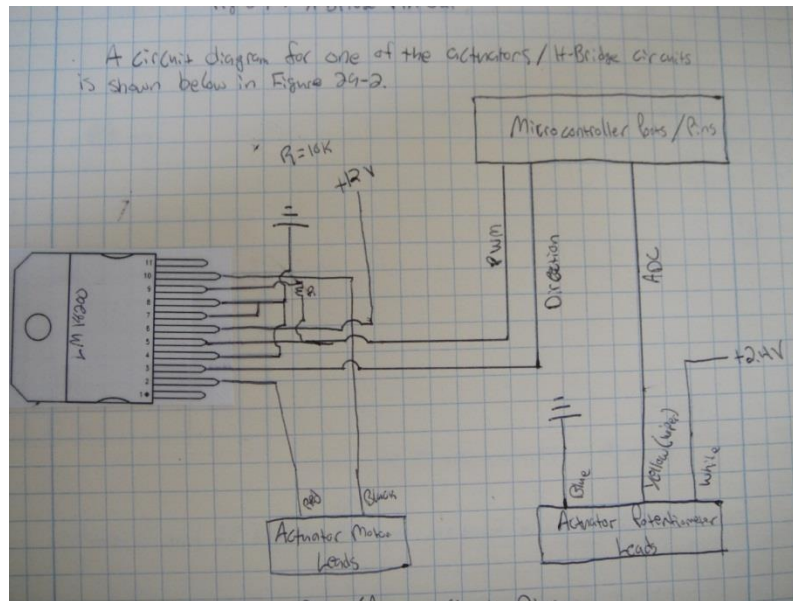


Figure X. H-Bridge Circuitry

### Suggested Future Work

- Continue the design and implementation of a windspeed control system.
- Modify windspeed control system to operate in a closed loop, reducing the error.
- Continue/modify the Java GUI to stream line its function.
- Advance the networking capabilities to the system can be accessed off campus.