Controller Design for a Linearly Actuated Active Suspension System

**(cdlaass)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**EE 452 Senior Project Final Report**

**Students:**

Dan Altman

Tim Reilley

Joseph Sholl

**Advisors:**

Dr. Anakwa

Prof. Gutschlag

**Date:**

4/15/2014

**Abstract**

Active suspension systems control the vertical movement of a vehicle using software and hardware with the aim of reducing or eliminating disturbances.  The controller design for a linearly actuated active suspension system project approaches the problem of limiting vertical movement of a vehicle using a linear actuator and hardware/software from National InstrumentsTM.  A disturbance is introduced to the system via an elliptical cam shaft attached to an AC motor.  The frequency of the disturbance is determined by a frequency drive controlling the AC motor.  As the controller software processes the vertical displacement of the vehicle by way of a potentiometer, the software outputs a signal(s) to circuitry driving the linear actuator lowering or raising the vehicle in response to the disturbance.

# **Table of Contents**

# Abstract i

Introduction 1

Project Summary 1

Project Description 1

Block Diagram 2

Functional Requirements 3

Controller Flow Chart 4

Preliminary Experimental Work 5

Design Equations and Calculations 6

Related Circuitry 7

Equipment List 12

Computer Simulations and Software 13

Analysis and Results 17

Recommendation for Future Work 18

References 19

**Introduction**

In the late 1980’s, a few automobile companies began introducing active suspension systems into some of their high-end automobile models. The biggest drawback to active suspension systems is the high cost. Some of the heavy machine manufacturers, including John Deere, have incorporated an active suspension system into the seats in the cab of their tractors.

We have designed a controller for an electric linear actuator-based active suspension system. A position sensor was used to determine the location of the “vehicle,” relative to the “wheel” position. The controller uses this information to engage the linear actuator to keep the mass at a relatively constant position. The addition of an accelerometer to the system will eventually be investigated to control the acceleration levels experienced throughout the range of available “wheel” displacement. LabVIEW will be used throughout the project as the controller platform. An additional deliverable of the project will be the creation of a tutorial (or guide) on the use of LabVIEW in controller design and implementation.

# **Project Summary**

The goal of this project was to design and construct a controller for an electric linear actuator-based active suspension system. Initially, a position sensor was used to determine the location of the “vehicle,” relative to the “wheel” position. The controller uses this information to engage the linear actuator to keep the mass at a relatively constant position. LabVIEW was used throughout the project as the controller and data acquisition platform.

An additional project deliverable was the creation of a tutorial (or guide) on the use of LabVIEW for data acquisition and controller design and implementation.

**Project Description**

This project involved focused efforts in power electronics design, system modeling and simulation, and feedback controller design. After the system and controller were simulated successfully utilizing Simulink, National Instrument hardware and software was used to implement the feedback controller and provide control signals to the power electronics driving the linear actuator of the active suspension system.

**Block Diagram**

# 

# **FIGURE 1: Complete System Block Diagram**

A high-level block diagram of the process is shown in Figure 1. Disturbances will be introduced to the system by a camshaft attached to an AC motor with a variable speed drive designed to provide external inputs to the suspension system. The controller will be developed using National Instrument’s hardware and software (LabViewTM). Sensors will provide data to the National Instrument’s module to provide feedback to the system controller.

**FIGURE 2: Complete Physical System**

**Functional Requirements**

* The controller shall drive the linear actuator to maintain a midpoint level, yet to be determined, and minimize displacement for a disturbance input with a maximum frequency of 5 Hz.
* The system shall minimize displacement of the cab from the midpoint to ± ⅛” (3.175 mm) with no load weight applied to the platform.
* The system shall minimize displacement of the cab from the midpoint to ± ¼” (6.35 mm) with a load.

**National Instruments Hardware**

* NI cDAQ-9174 NI CompactDAQ 4-slot USB 2.0 Chassis, 9 V - 30 V Input Voltage Range
* NI 9401 Digital Input/Output Module
* NI 9221 8-Channel Voltage A/D Module (12-bit resolution), 800 kS/s/ch sample rate, ± 60 V

**H-Bridge and Gate Driver Hardware**

* One (1) IGBT Full-Bridge Power Module (Fuji 7MBR30NF060)
* Four(4) HCPL-3120 Gate Driver Optocouplers

**Controller Flow Chart**

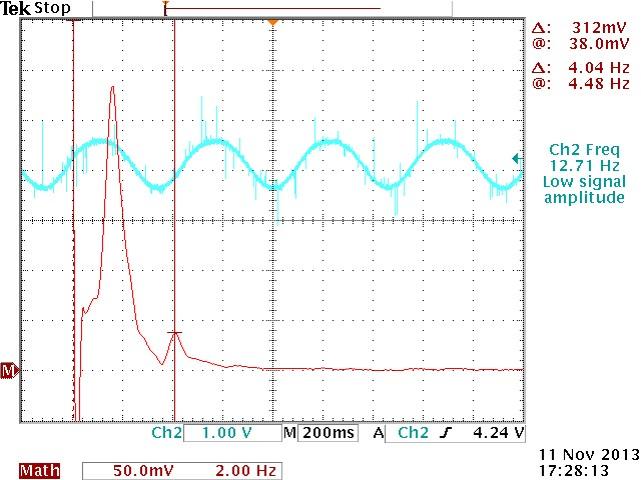
****

**FIGURE 3: Controller Flow Chart**

As disturbances are introduced to the system, a signal from the potentiometer is fed through an A/D converter in order to be processed by the controller software.  The controller sends signals to the linear actuator to respond to the disturbance.

**Preliminary Experimental Work**

The first test conducted in the laboratory was to observe the disturbance on an oscilloscope. To ensure that the disturbance applied to the system was a pure sinusoidal signal. The results are pictured in Figure 3 below. An oscilloscope Fast Fourier Transform (FFT) analysis on the waveform obtained from the disturbance indicated there was a second harmonic present in the signal. This means that the disturbance isn’t a pure sinusoid, but it is still a viable input signal for the system.



**FIGURE 4: Disturbance Analysis**

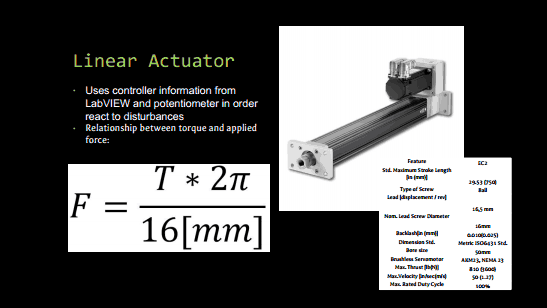
FFT analysis of the output from the position sensor (potentiometer) reveals that the disturbance is not a true sinusoid.

Disturbance equations [1]: r(t)=0.0127sin(ωt) [m]

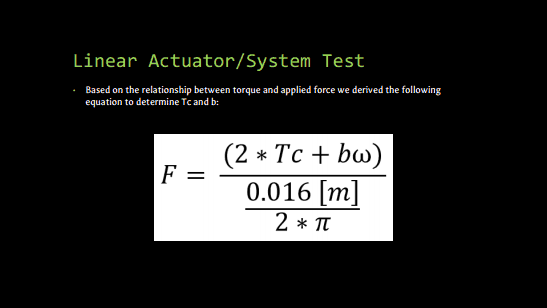
or

r(t)=0.5sin(ωt) [in]

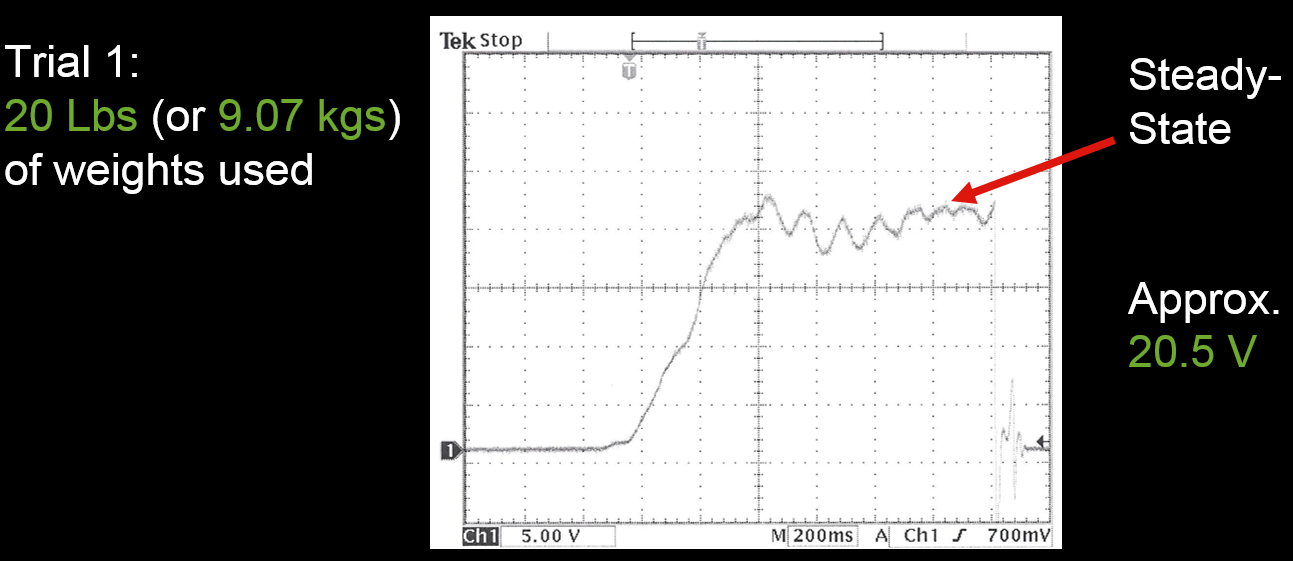
**Design Equations and Calculations**



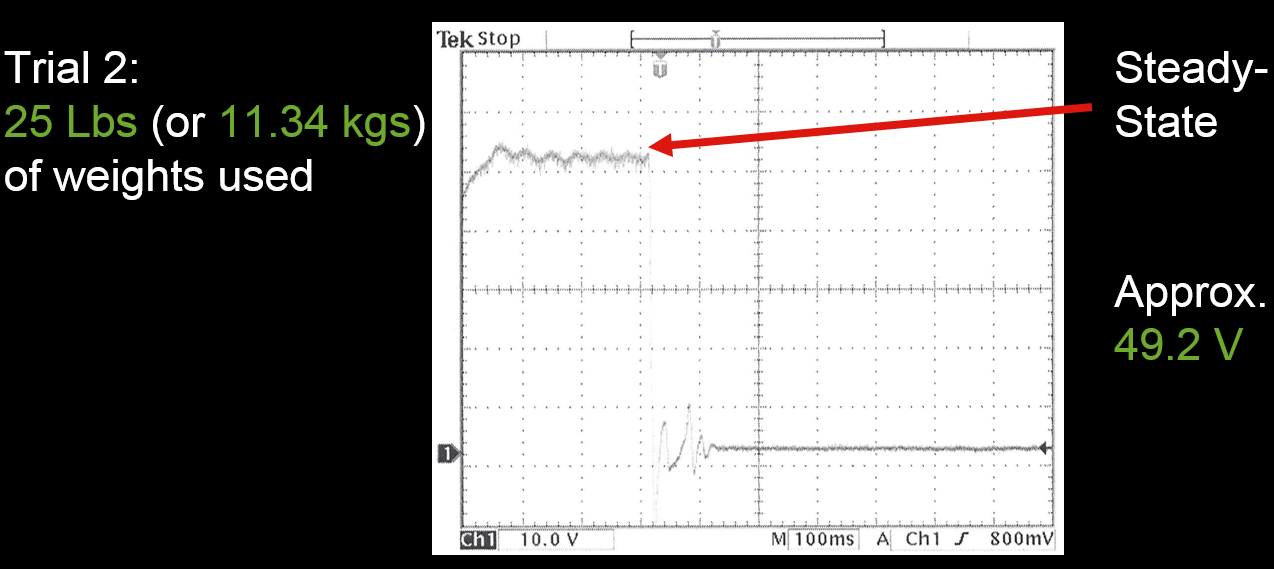
**EQUATION 1: Initial Force to Torque Relationship**

****

**EQUATION 2: Derivation of Force to Torque relationship to determine Tc and b**



**FIGURE 5: Trial 1 to Determine Steady State with 20lbs Load**



**FIGURE 6: Trial 2 to Determine Steady State with 25lbs Load**

Based on the previous group’s work, kE=0.382 [V/rad/sec] so,

Trial 1: ω = 20.5 [V] / .382 [V/rad/sec] = 53.67 [rad/sec]

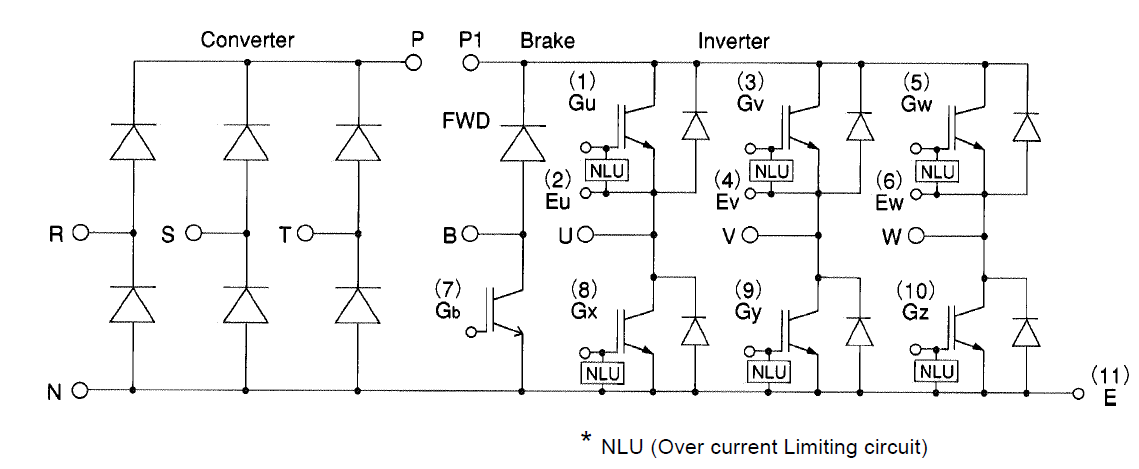
Trial 2: ω = 49.2 [V] / .382 [V/rad/sec] = 128.80 [rad/sec]

Using simultaneous equation solver and Equation 2: **Tc = 0.09304 and b = 7.55 X 10-4**

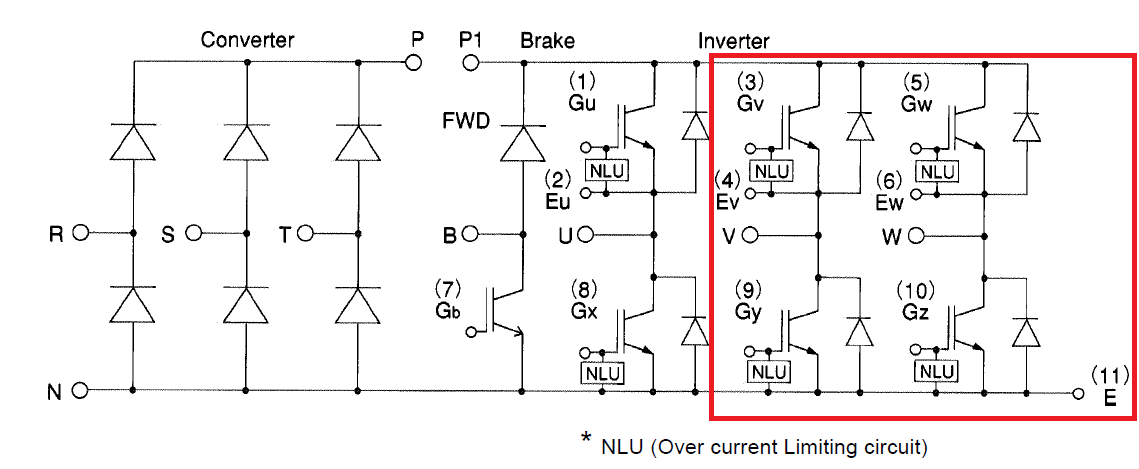
**Related Circuitry**

An H-bridge was designed to control the linear actuator motor. This H-bridge controls whether the linear actuator runs in the forward (up) position, or in the reverse (down) position. For the H-bridge design, we used an IGBT Power Module (Fuji 7MBR30NF060). The complete circuit schematic of this IGBT is show in Figure 7. The part of the IGBT circuit that we used for our H-bridge design is shown in Figure 8. When the "v" transistor and the "z" transistor are ON, the motor will turn in the forward direction. When those two transistors are OFF and the "w" and "y" transistors are turned ON, the motor will turn in the reverse direction.

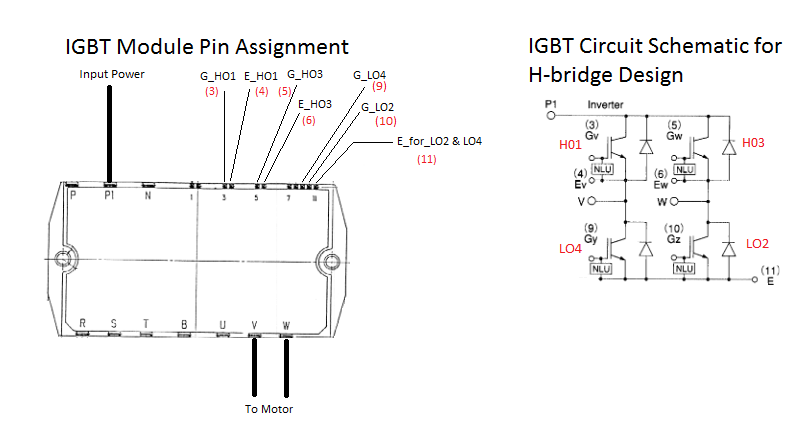
The resulting H-bridge will utilize four (4) HCPL-3120 Gate Drive Octocoupler chips to drive the IGBT Power Module. The HCPL-3120 Gate Drivers have built in optical isolators. This will assist in protecting the logic circuitry from high voltage feedback. This is done by optically passing desired signals, while maintaining electrical isolation between the two systems5.



**FIGURE 7: Complete IGBT Power Module Circuit Schematic**

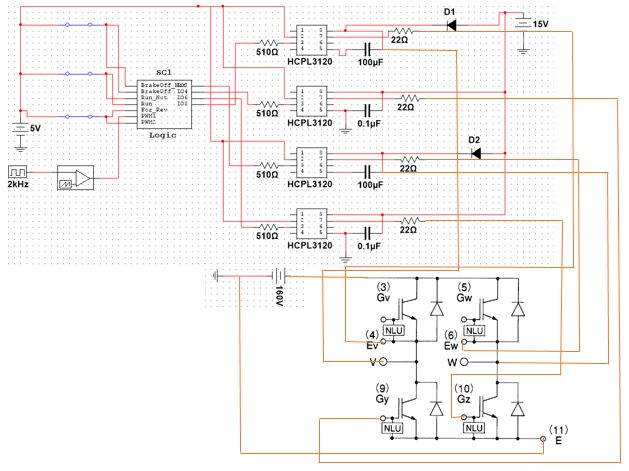


**FIGURE8: Part of the Circuit Used for H-bridge Design (Shown in Red Highlighted Box)**

****

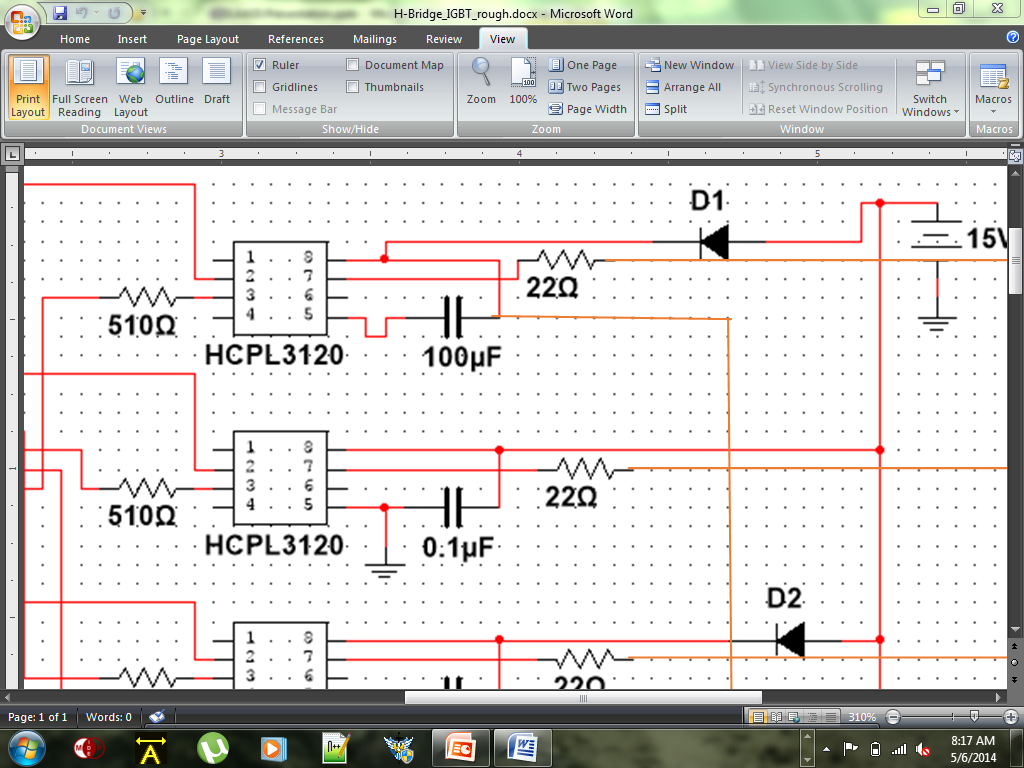
**FIGURE 9: Pin Assignment for the IGBT Power Module**

The system will require isolation circuitry to protect the controller hardware from the relatively high voltages present in the suspension system. A schematic diagram of this circuitry is shown in Figure 10. In addition to the protection circuitry, the H-Bridge will be constructed from one IGBT Full-Bridge Power Module.

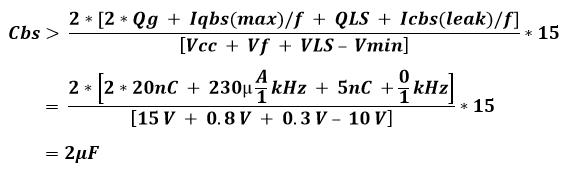


**FIGURE 10: Complete H-bridge Design Circuit Schematic**

The Optocoupler driver circuit requires the bootstrap arrangement consisting of the diode and capacitor as shown in Figure 11. Furthermore, there is additional snubber circuitry on both the low sides of the H-Bridge in order to reduce the effects of voltage spikes and allowing the current to flow through it.

****

**FIGURE 11: Bootstrap Circuit Configuration**



**EQUATION 3: Bootstrap Circuit Configuration**

From the calculations in Equation 3, we see that we needed at least a 2 micro-Farad capacitor. We ended up utilizing a 100 micro-Farad capacitor for our H-bridge circuit. The snubber circuit values are calculated below in Equation 4. We referenced the previous groups work for the equation.

Csopt= (Ico\*TF ) / (sqrt(12)\*VOFF)

Csopt= (25A \* 1.5u) / (sqrt(12)\*161)

Csopt= 6.7238x10-8 ~ 67nF

Rs< (Dmin\* Ts) / (3 \* Csopt)

Rs < (10% \* (1 / 2000)) / (3 \* 67nF)

Rs < 245.7562Ω

Or

Rs<Vbr/ Ico

Rs < 800V / 25A = 32Ω

**EQUATION 4: Snubber Circuit Calculation**

The capacitor we chose was 100nF and the resistor we chose was 22Ω due to availability.

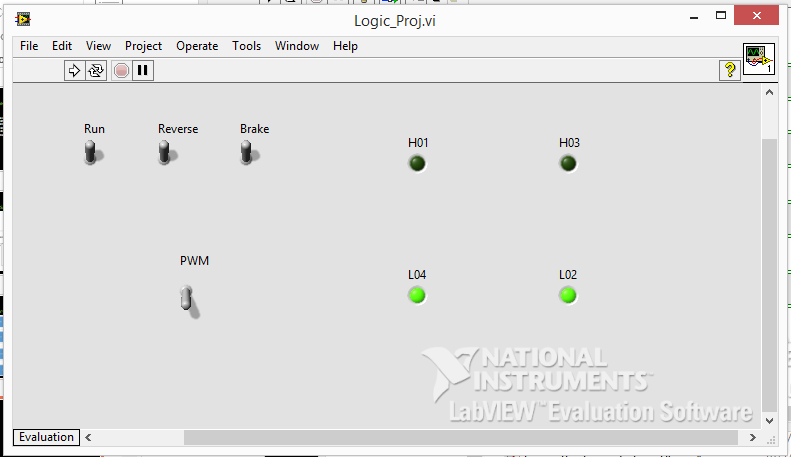
**Equipment List**

* Tektronix Agilent oscilloscope
* DC power supply
* NI cDAQ-9174 NI CompactDAQ 4-slot USB 2.0 Chassis, 9 V - 30 V Input Voltage Range
* NI 9401 8-channel ultra-high speeddigital Input/Output Module
* NI 9221 8-Channel Voltage A/D Module (12-bit resolution), 100 kS/s/ch sample rate, ± 60 V
* One (1) IGBT Full-Bridge Power Module (Fuji 7MBR30NF060)
* Four(4) HCPL3120 Gate Driver Optocouplers

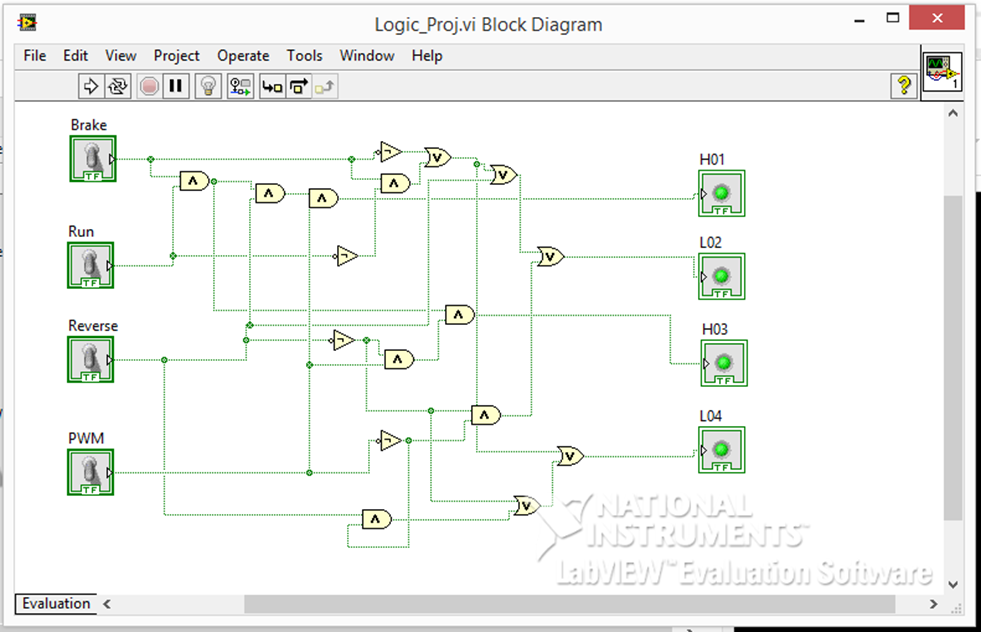
# 

# **Computer Simulations & Software**

We started our use of LabVIEW with a simple block diagram of the logic (Figure 13) used for our H-bridge. Figure 12 shows the front panel in LabVIEW which is used to control manual inputs while the program is running. The inputs for this design include the run, reverse, brake and PWM switches. The front panel is also used for displaying any indicators used in the program. In the logic program, we wanted to observe the signal received by the two high side transistors and the two low side transistors. In order to observe these signals we placed four virtual LEDs on the front panel and connected them to the appropriate outputs of our logic in the block diagram. Once the program was created, we could click on the switches to turn them on or off and note which LEDs illuminated based on the logic.

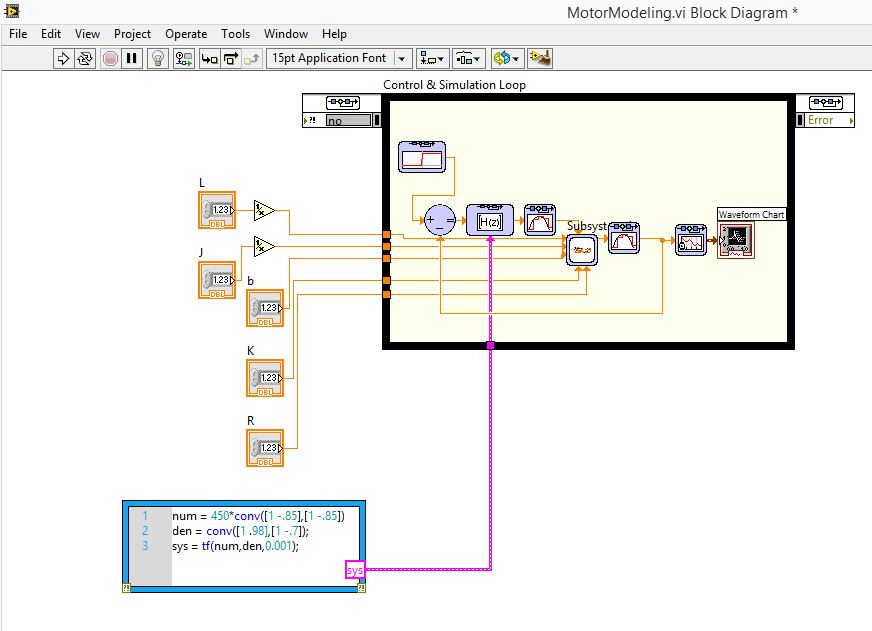


**FIGURE 12: Template for Logic Testing Using Switches**

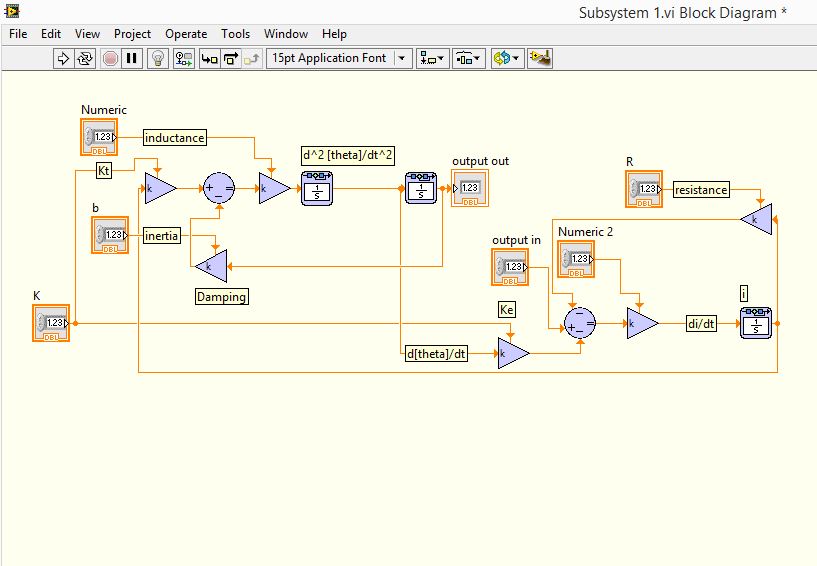


**FIGURE 13: LabVIEWBlock Diagram for Logic**

The next step of the project involved modeling the system including the parameters we found during the trials shown in Figure 5 and Figure 6. The block diagram for the system model can be seen in Figure 14. This diagram includes a subsystem of which an expanded block diagram is shown in Figure 15. After constructing the system model, we ran simulations in LabVIEW to observe the behavior of the system. The results can be seen in Figure 16.

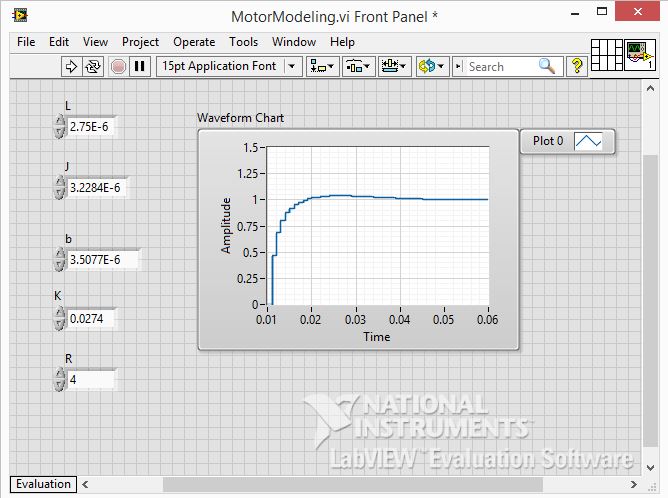


**FIGURE 14: LabVIEWMotor Modeling Block Diagram**



**FIGURE 15: LabVIEW Subsystem Block Diagram**

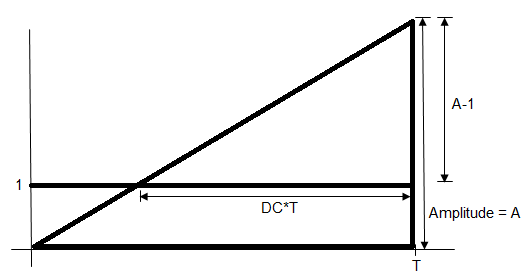
The previous group’s work included a system model constructed in Simulink. This model was incomplete as it did not include the TC (columbic friction) and b parameters. Comparing their results in Simulink with the LabVIEW model confirmed the correct construction of the model in LabVIEW and showed a slight effect contributed by the additional parameters. The results of the LabVIEW simulation can be seen in Figure 16. The front panel shown also includes fields for inputting values for the system model. These variables could have been added as constants, but were kept as numerical inputs for flexibility.



**FIGURE 16: LabVIEW Motor Modeling Front Panel**

To drive the system modeled above, it would be necessary for LabVIEW to send a PWM of varying duty cycles, as well as three other signals represented at switches in Figure 12, to the H-bridge circuit. To generate the PWM, we used a sawtooth waveform generator block and adapted it to output ramp signals. We changed its phase by 180 degrees and included an offset based on half the amplitude to accomplish the sawtooth to ramp conversion. Creating the data samples for the PWM was done by comparing the ramp signals of varying amplitude to a DC value of one. For every sample of the ramp that was above one, a one was stored. For every sample below one, a zero was stored. The different duty cycles were then created by increasing the amplitude thereby increasing number of samples above the DC comparison.

Calculating the amplitude necessary for each duty cycle was based on the relationship between triangles with equal angles.

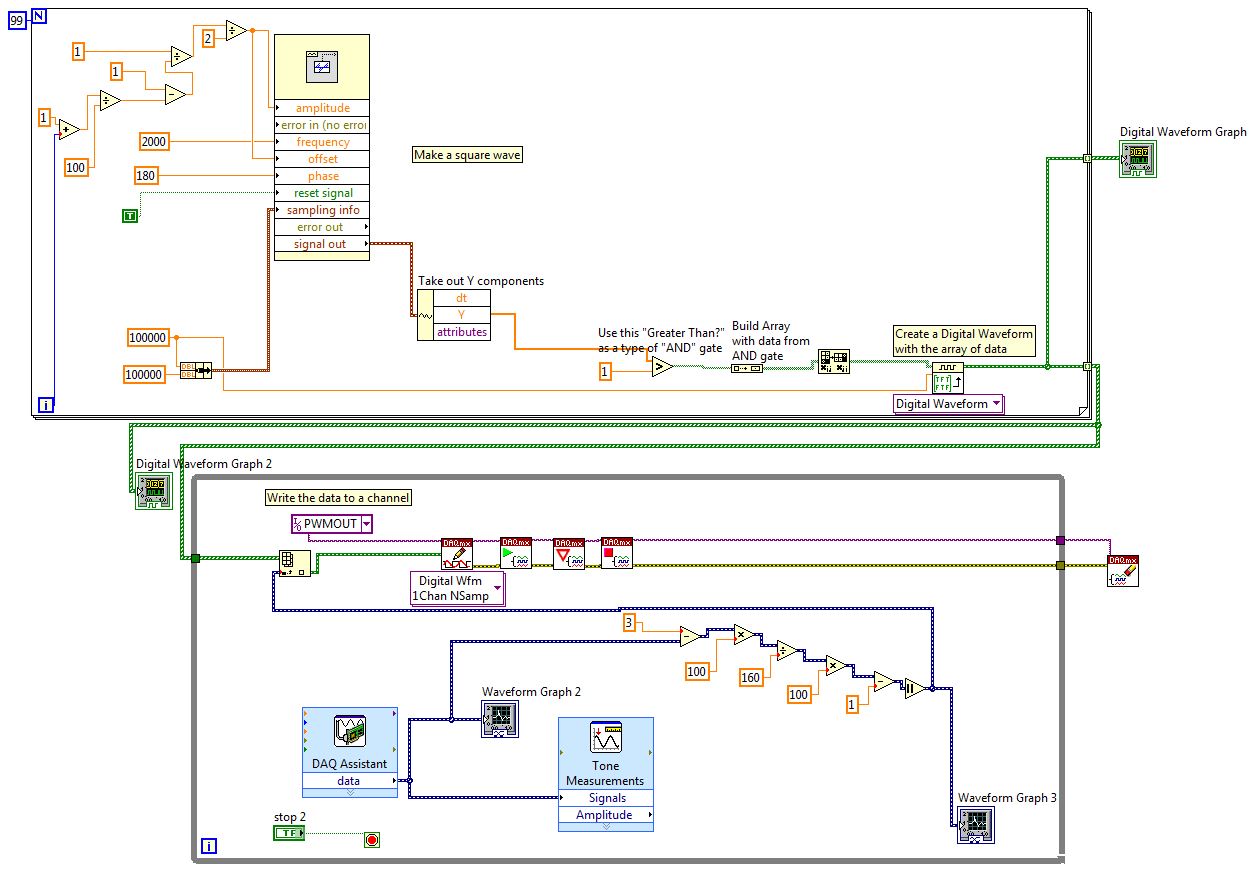


**FIGURE 17: Basis for Amplitude Calculation to Create Duty Cycles**

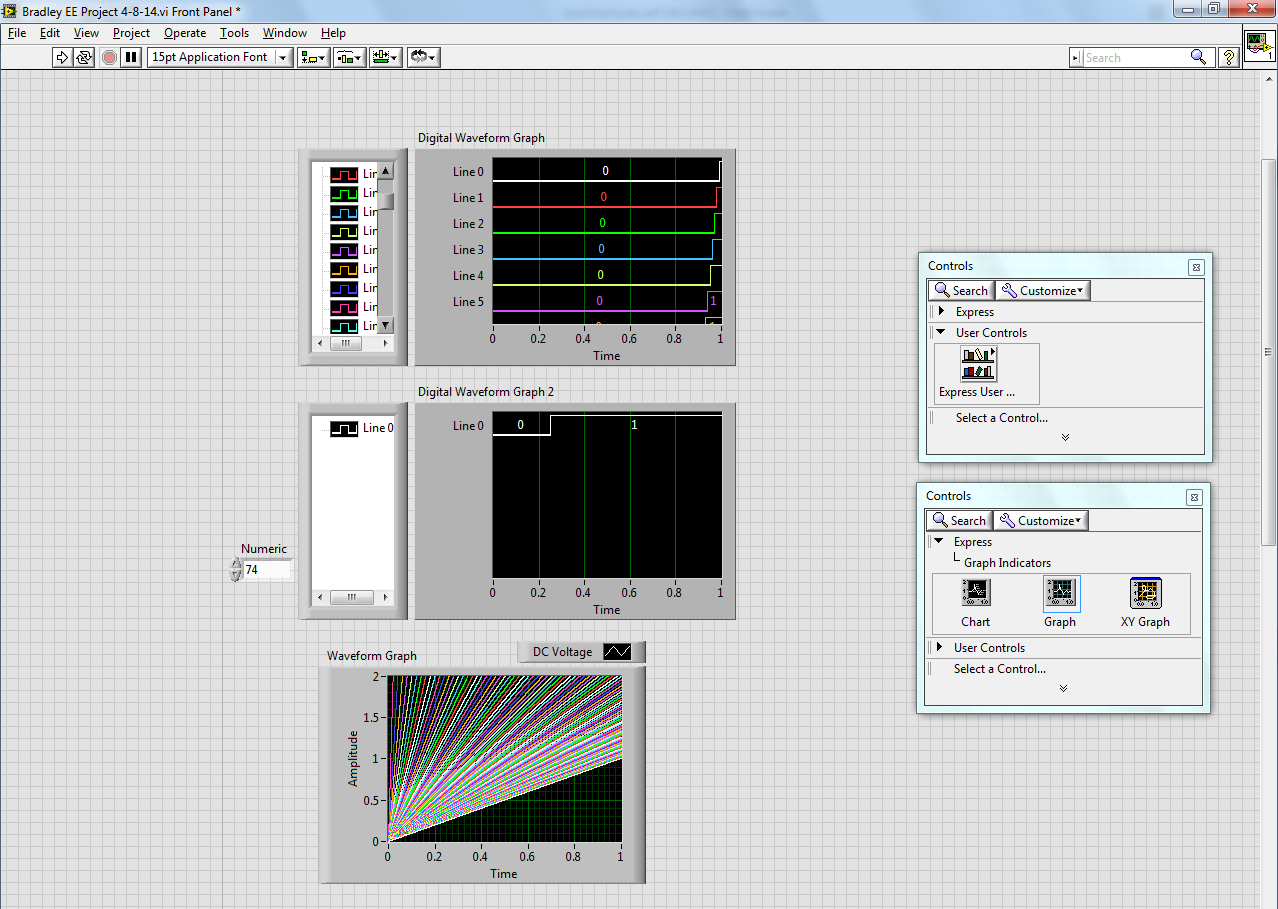
As seen in Figure 17, the right triangles created by the ramp and the DC value have equal angles. Using this information and the variables listed in the figure, we calculated the formula for the amplitudes needed for each duty cycle value from one percent to 99 percent, in integer increments. The calculations can be seen below in Equation 5.

**EQUATION 5: Amplitude and Duty Cycle Relationship**

With the relationship between duty cycle and amplitude calculated, we created a loop to build an array of PWMs with duty cycles from one percent to 99 percent based on the loop iteration. In Figure 19, the array of PWMs is displayed in the top graph. In the same figure, the middle graph shows a 75 percent duty cycle PWM that was selected manually as a test. The bottom graph shows the 99 ramp signals that are compared to a DC value of one in order to create the varying duty cycles.



**FIGURE 18: LabVIEW Waveform Generator and Data Acquisition**



**FIGURE 19: LabVIEWWaveform Output Front Panel**

From the array of PWMs, the controller can select which PWM duty cycle is appropriate to control the speed of the linear actuator. The loop creating the array of PWMs is in the upper part of Figure 18. The bottom loop compares the disturbance signal to a set point and then selects the appropriate duty cycle. This loop also contains the blocks that read the disturbance signal and write the selected PWM to the National Instruments hardware.

# **Analysis of Results**

This project shows that National Instruments software and hardware are viable solutions to controller design and implementation in the design and test phases of controller projects. Since we were unfamiliar with the LabVIEW software and its associated hardware, a lot of time went into researching design methods for the platform. Data acquisition in LabVIEW is fairly easy to set up, but signal generation proved to be a challenge. Specifically, PWM creation can be done one of at least a dozen ways. Despite some of the setbacks we faced, we successfully created a program that can output PWM signals. The LabVIEW program controlled the operation of a Pittman motor via the H-bridge and power electronics we designed and constructed. Based on a disturbance signal, the program selected appropriate duty cycles and output them to the H-bridge circuit.

Due to the time involved in learning LabVIEW, we did not design a controller using control system theory. We also did not integrate the linear actuator into the LabVIEW/H-bridge circuit in order to reduce the wear and tear on the actuator. Once the PWM output is smoothed, the addition of the linear actuator to the system will be a simple fix. We believe that the system is in a state where subsequent groups will be able to pick up where we left off. Some recommendations for future work with the system include expanding the LabVIEW license to include RT and Control System Toolbox modules. Also, a National Instruments CompactRIO instead of the cDAQ would prove beneficial in the controller implementation.

# **References**

[1] Blake Boe and Tyson Richards. “Active Suspension System”, Senior

Project, Electrical and Computer Engineering Department, Bradley University, May

2006, <http://cegt201.bradley.edu/projects/proj2006/actss/>

[2] IDC Motion EC2 Series Linear Actuator Data Sheet

[3] National InstrumentsHardware Data Sheets

[4] Anakwa, Dr. Winfred, "Development and Control of a Prototype Pneumatic Active Suspension System" (with students), IEEE Transactions on Education, Vol. 45, No. 1, February 2002, pp. 43-49.

[5] Jeremy Seah Eng Lee, Alexander Jaus, Patrick Sullivan, Chua Teck Bee. "Building a Safe and Robust Industrial System with Avago Technologies’ Optocouplers," March 2010, http://www.avagotech.com/docs/AV02-0835EN