Dynamic Shock Absorber

A Major Qualifying Project

B-Term Report

By

Joseph Phaneuf

jphaneuf@wpi.edu

Khondkar Faiaz Hasan

fhasan@wpi.edu

Tanvir Anjum

tanjum@wpi.edu

Date of Submission: 5th March, 2010

Advisor: Professor Alexander Emanuel

aemanuel@wpi.edu

Abstract

The goal of this project was to build and test a model of an active suspension system to minimize the vertical oscillations of a vehicle's body for varying road conditions. The road condition was simulated by introducing an oscillation to the system through an electromagnet. An electromagnet based closed loop control system was established to oppose the vehicle's motion. It was concluded that the vertical oscillations experienced by the vehicle were significantly reduced by the implementation of a proportional control system.

Executive Summary

The purpose of this project was to design an advanced suspension system using electromagnetic shock absorbers. The principle behind this innovative idea is to compensate for shocks experienced by a vehicle by producing an equal and opposite force on the vehicle's body to negate the effect of the shock. This would essentially keep the body of the car stationary when subjected to external forces due to factors such as pot holes and speed bumps.

The final design idea was based on many criteria, and the ultimate model was eventually chosen after iterating through various design possibilities. Design decisions were based on economical feasibility of the product, to make sure that the production cost was as low as possible, as the cost of the shock absorbers, in reality, would be the major role player in consumer popularity. The idea was such that, once incorporated in automatic vehicles, the cost of the shock absorbers would be negligible due to the feasibility of mass production.

Modeling the shock absorbers, based on both mechanical and electrical systems, was the first step towards the design model. The mechanical representation of the model reflects the suspension systems currently present in the market after thorough analysis of shock absorbers currently in production. This representation involved prototyping the shock absorbers mechanically, with the electromagnets in place.

The mechanical system, once established, was equated to its equivalent electrical system. This electrical system was researched, and several design ideas were tried out. This trial and error method first involved modeling and simulating the system. The simulation results allowed us to study the system at various frequencies with and without the control system.

The next step towards the model dealt with designing an oscillator circuitry module that would essentially provide the prototype with varying force oscillations, which in reality would resemble the forces experienced by a vehicle due to changing road conditions. This force, once exerted, would cause the body to oscillate vertically. This oscillation can be sensed by an accelerometer. The sensor then sends out a signal to the control module, where the microcontroller drives an actuator to produce an opposing force eventually keeping the body stationery. Ideally the shock absorber would be designed to act towards any changes in road conditions, and maintain constant stability for the vehicle.

Table of Contents

List of Figures

List of Tables

Introduction

Safety and driving comfort for a car's driver are both dependent on the vehicle's suspension system. Safety refers to the vehicle's handling and braking capabilities. The comfort of the occupants of a car correlates to tiredness and ability to travel long distance with minimal annoyance.

Shock absorbers are a critical part of a suspension system, connecting the vehicle to its wheels. Essentially shock absorbers are devices that smooth out an impulse experienced by a vehicle, and appropriately dissipate or absorb the kinetic energy [1]. Almost all suspension systems consist of springs and dampers, which tend to limit the performance of a system due to their physical constraints. Suspension systems, comprising of springs and dampers, are usually designed for passengers' safety, and do little to improve passenger comfort.

One solution to this could be the use of active suspension systems, where road conditions are detected using sensors, and the system instantaneously adapts to the setting. Based on the sensor feedback, the adaptation of the vehicle's system to various road conditions are implemented by a control signal.

The aim of this project is to design a dynamic shock absorber that uses electromagnetic circuits and sensors to detect the movement of the wheels to help the vehicle adapt to varying road conditions. The ideal goal is to keep the body of the vehicle stationery, regardless of road conditions.

1 Modeling of the Suspension System

1.1 Introduction

Before reaching the appropriate design of the dynamic shock absorber, one has to understand the basic functionalities of a shock absorber. A shock absorber is a device which is designed to smooth out sudden impulse responses, and dissipate kinetic energy. Any moving object possesses kinetic energy, and if the object changes direction or is brought to rest, it may dissipate kinetic energy in the form of destructive forces within the object [1]. The purpose of a shock absorber, within any moving object, is to dissolve the kinetic energy evenly while eliminating any decelerating force that may be destructive to the object.

A typical shock absorber may simply comprise of a compression spring that is capable of absorbing energy. Commonly shock absorbers are known as dashpots, which is simply a fluid filled cylinder with an aperture through which fluid could escape under controlled conditions [1]. The dashpot is the building block for pneumatic and hydraulic shock absorbers [1]. These shock absorbers essentially consist of a cylinder, filled with air or fluid, with a sliding piston that moves to dissipate or absorb energy, and in these cases the energy is usually dissipated as heat [1].

Figure 1-1 shows the internal components of a typical hydraulic shock absorber.

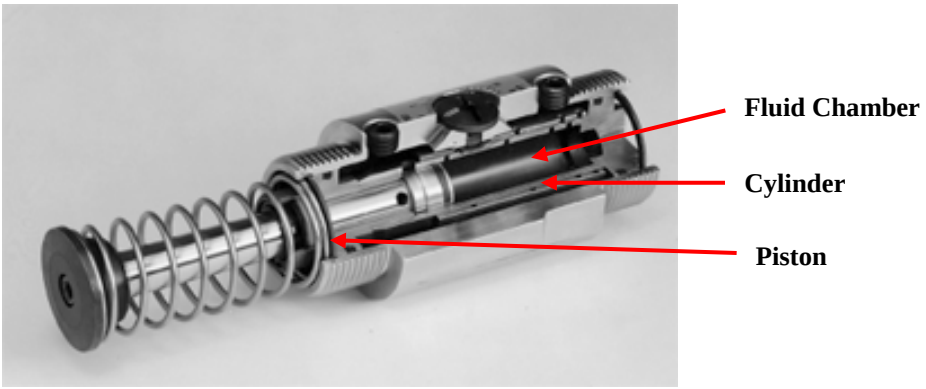

Figure 1-1: Cutaway View of Hydraulic Shock Absorber [1]

The main design constraint associated with shock absorbers is a vehicle's resonant frequency, the tendency of a system or object to oscillate at larger amplitudes for certain frequencies.

After analyzing the shock absorber mechanism in detail, the ideal model for the dynamic shock absorber was designed. Modeling the mechanical prototype of the shock absorber unfolded intricacies involved with suspension systems, and is the primary step to understand how changing variables associated with shock absorbers could affect the functionality of the suspension system.

1.2 Modeling of the Suspension System

1.2.1 Prototyping

The real life prototype of the system is shown in Figure 1-2.

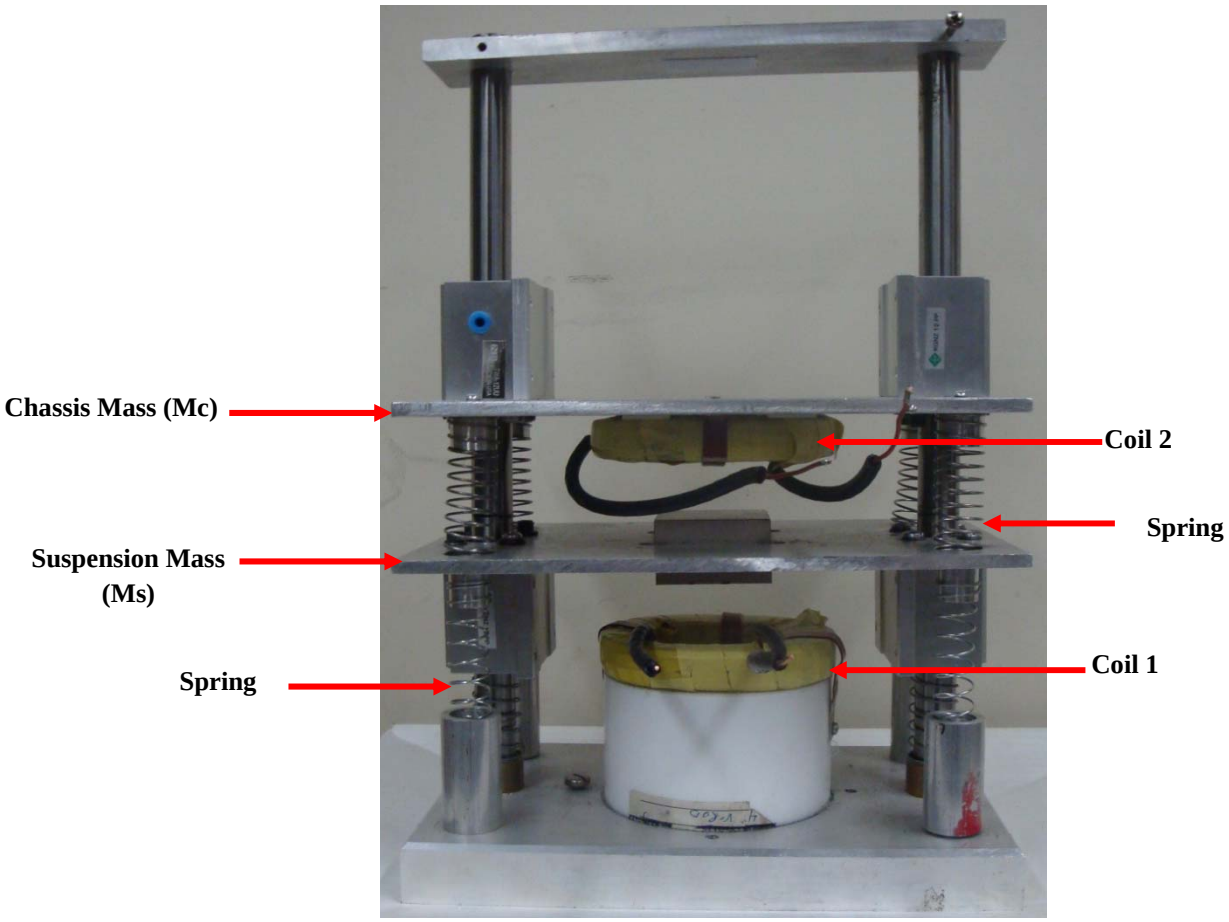

Figure 1-2: Dynamic Shock Absorber Prototype

The modeling of the suspension system is based on this prototype. It is comprised of two masses that represent the Chassis Mass (Mc) and the Suspension Mass (Ms). K1 and K2 represent the spring constants, and D1 and D2 represent the damping coefficients for Ms and Mc. Coil 1 will be used to apply electromagnetic force on the mass Ms to simulate shock or vibration that the system might experience. The main idea is to use Coil 2 to exert a force on the Chassis Mass so that it remains stationary.

1.2.2 Physical Model of Suspension System

The physical model of the suspension system is portrayed in Figure 1-3.

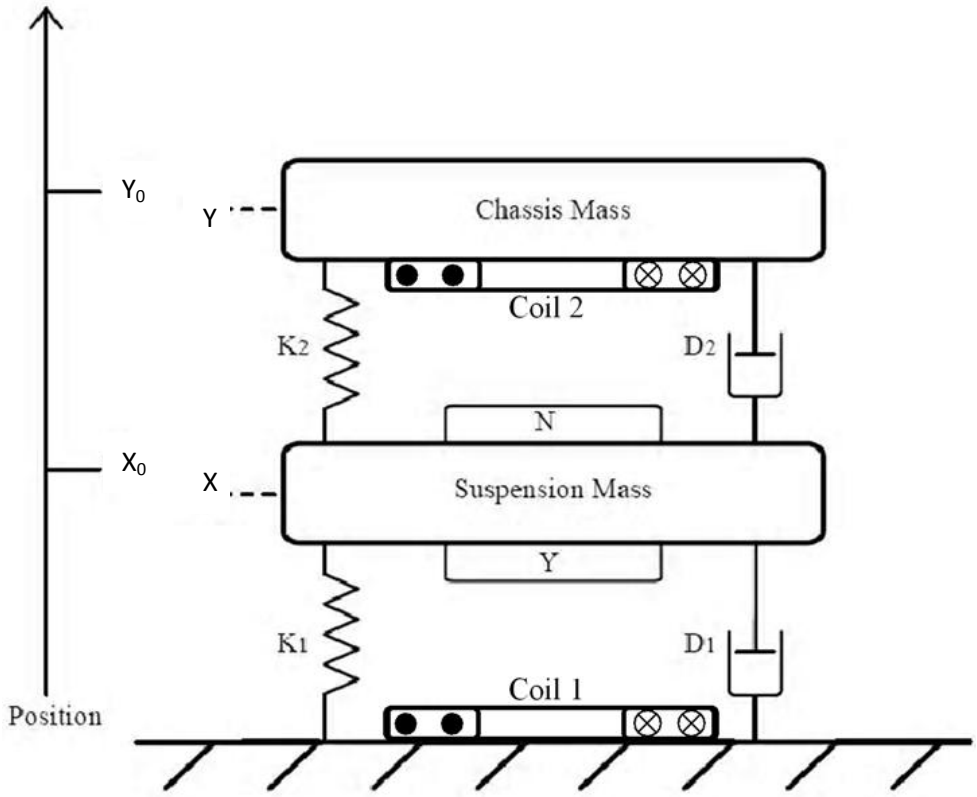

Figure 1-3: Cutaway View of Dynamic Shock Absorber

The following notations are used for the eight forces working on the two masses.

Fs1: Force due to compression of spring K1

Fs2: Force due to compression of spring K2

Fd1: Damping force due to movement of Suspension Mass

Fd2: Damping force due to movement of Chassis Mass

Fas: Force due to acceleration of Suspension Mass

Fac: Force due to acceleration of Chassis Mass

F(t): Oscillating force created by Coil 1 to simulate adverse road conditions

fc: Compensation force created by Coil 2 to counteract the effects of F(t)

Under the effect of F(t), Ms moves X and Mc moves Y.

1.2.2.1 Analysis of Forces on Chassis Mass

The free body force diagram in Figure 1-4 shows the forces acting on the mass Mc. There are three forces acting on the mass. The Force from the spring is the extension of the spring (Y-X) times the spring constant K2. The damping force is the result of movement of the mass Mc. Its magnitude is the product of velocity and the damping constant D2. The third force acting on the mass arises due to the inertia of the mass. It is the product of the mass of Mc and its acceleration. The goal is to keep the net force acting on the body mass approximately to zero.

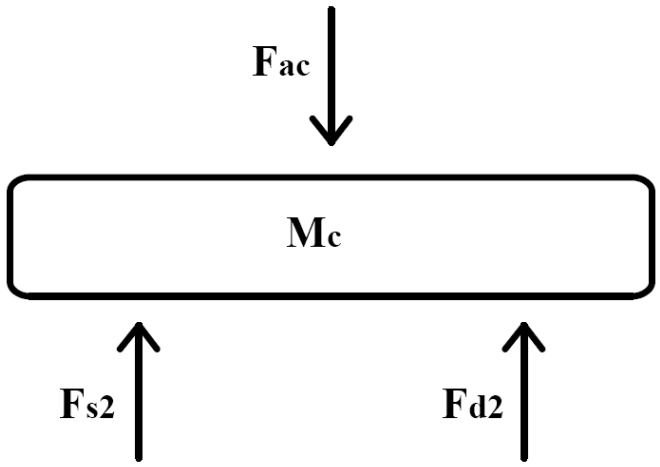

Figure 1-4: Chassis Mass Free Body Diagram

The following equations will show how the forces are acting on Mc. The notations used to describe the velocity and the acceleration are shown in Table 1-1.

$\dot{x}$ *and* $\dot{y}$ *are used to denote velocity ;* $\ddot{x}$ *and* $\ddot{y}$ *are used to denote the acceleration*

$\frac{dx}{dt} = \dot{x} = u_x,$	$\frac{dy}{dt} = \dot{y} = u_y,$	$\frac{d^2x}{dt^2} = \ddot{x},$	$\frac{d^2y}{dt^2} = \ddot{y},$

Table 1-1: Derivative Notations

From figure 1-4, the relationship between the three forces is described by equation 1.2.

$$F_{ac} = F_{S2} + F_{d2} \tag{1.2}$$

This equation can be summarized as:

$$M_C \frac{du_y}{dt} = K_2 \int (u_x - u_y) dt + D_2(u_x - u_y) \tag{1.3}$$

$$M_C \ddot{y} = K_2(x - y) + D_2(\dot{x} - \dot{y}) \tag{1.4}$$

1.2.2.2 Analysis of Forces on Suspension Mass

Figure 1-5 below shows the free body force diagram of the mass Ms. Ms experiences six forces at any given time. F(t) is the force from Coil 1. This force is the simulation of shock the system might experience due to road conditions. Fas is the force exerted on the mass due to its acceleration. The forces applied on the body because of the two springs are denoted as Fs1 and Fs2. Fd1 is the damping force acting on the Suspension Mass due to friction in air, the bearings and within the spring.

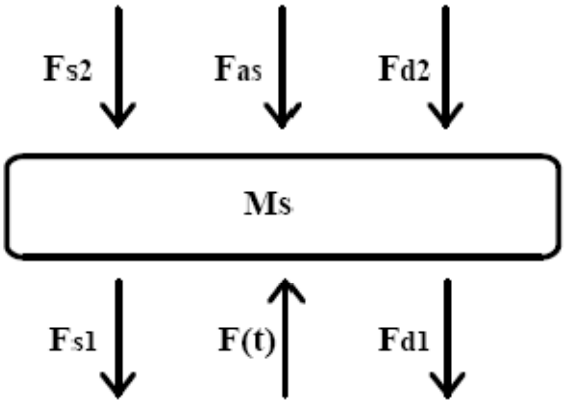

Figure 1-5: Suspension Mass Free Body Diagram

The overall force acting on the Suspension Mass is shown in equation 1.5.

$$F(t) = F_{S1} + F_{d1} + F_{S2} + F_{d2} + F_{as} \tag{1.5}$$

Equation 1.5 can be rewritten in terms of displacement, velocity, acceleration, and the system's physical parameters.

$$F(t) = K_1 \int u_x \, dt + D_1 u_x + K_2 \int (u_x - u_y) \, dt + D_2(u_x - u_y) + M_S \frac{du_x}{dt} \tag{1.6}$$

$$F(t) = K_1 x + D_1 \dot{x} + K_2(x - y) + D_2(\dot{x} - \dot{y}) + M_S \ddot{x} \tag{1.7}$$

1.2.2.3 Analysis of Forces on Both Masses

Figure 1-6 shows masses Mc and Ms, and all of the forces acting on them. This figure includes the compensation force fc. With F(t) oscillating, fc, generated by Coil 2, will act to keep the net force on Mc zero, effectively keeping the Chassis Mass stationary.

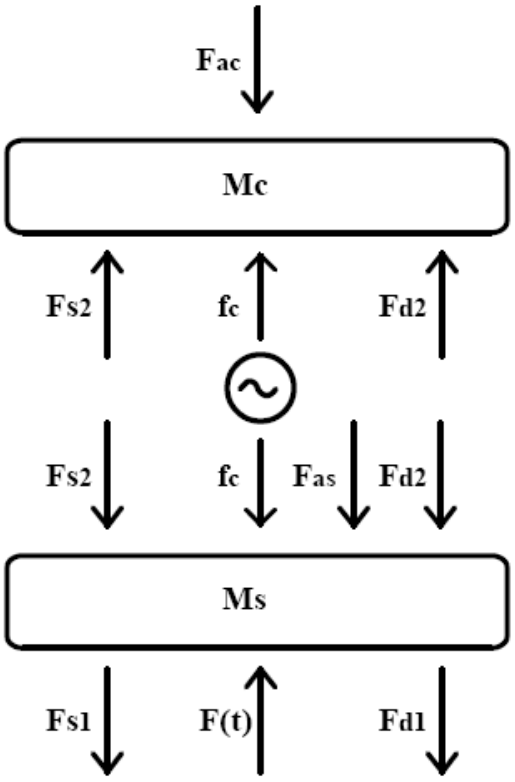

Figure 1-6: Free Body Diagram for Both Masses

The net force acting on the Chassis Mass, M_C, with Coil 2 in place is given by equation 1.8.

$$F_{ac} = f_c + F_{S2} + F_{d2} \tag{1.8}$$

Equation 1.9 is derived from equation 1.8.

$$M_C\ddot{y} = f_c + K_2(x - y) + D_2(\dot{x} - \dot{y}) \tag{1.9}$$

Equation 1.10 describes the net force acting on the Suspension Mass with Coil 2 in place.

$$F(t) = f_c + F_{S1} + F_{d1} + F_{S2} + F_{d2} + F_{as} \tag{1.10}$$

Rewriting equation 1.1 yields equation 1.11, which shows Forces related to the velocity of the masses and the extension of the springs.

$$F(t) = f_c + M_S\ddot{x} + D_1\dot{x} + K_1x + D_2(\dot{x} - \dot{y}) + K_2(x - y) \quad (1.11)$$

1.3 Measurements

1.3.1 Mass

Masses, Ms and Mc, were measured using a force plate in Logger Pro. The entire system was disassembled, and each plate was measured separately. A series of measurements were taken, and then averaged, yielding the following:

$$Ms = 1.641\ Kg \quad (1.12)$$

$$Mc = 1.609\ Kg \quad (1.13)$$

1.3.2 Spring

The spring constant K1 was measured by removing, the top plate, and adding various weights to the bottom plate, while measuring the corresponding height of the plate. The spring constant K2 was measured similarly, with the bottom plate held fixed. The resulting measurements are shown in Table 1-2. Plots of length vs. weight for the suspension spring and chassis spring are shown in Figures 1-7 and 1-8 respectively.

Suspension K1			Chassis K2	
L (cm)	Weight(kg)		L (cm)	Weight(kg)
14.8	0		16.8	0
13.6	2		16.4	0.5
13.25	2.5		16	1
13	3		15.7	1.5
12.65	3.5		15.35	2
12.35	4		15	2.5

Table 1-2: Spring Constant Data

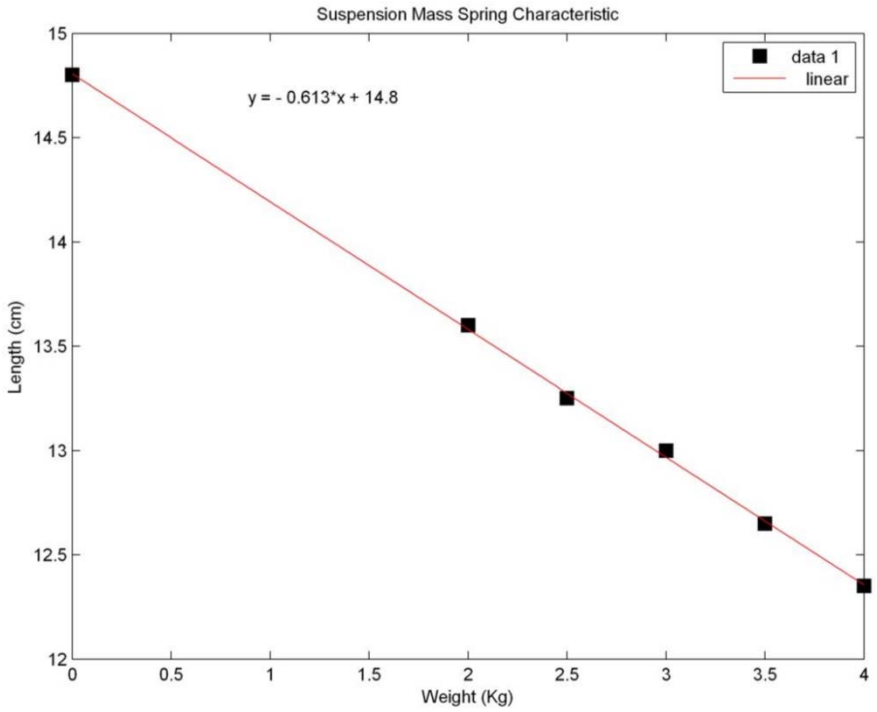

Figure 1-7: K1 Spring Constant

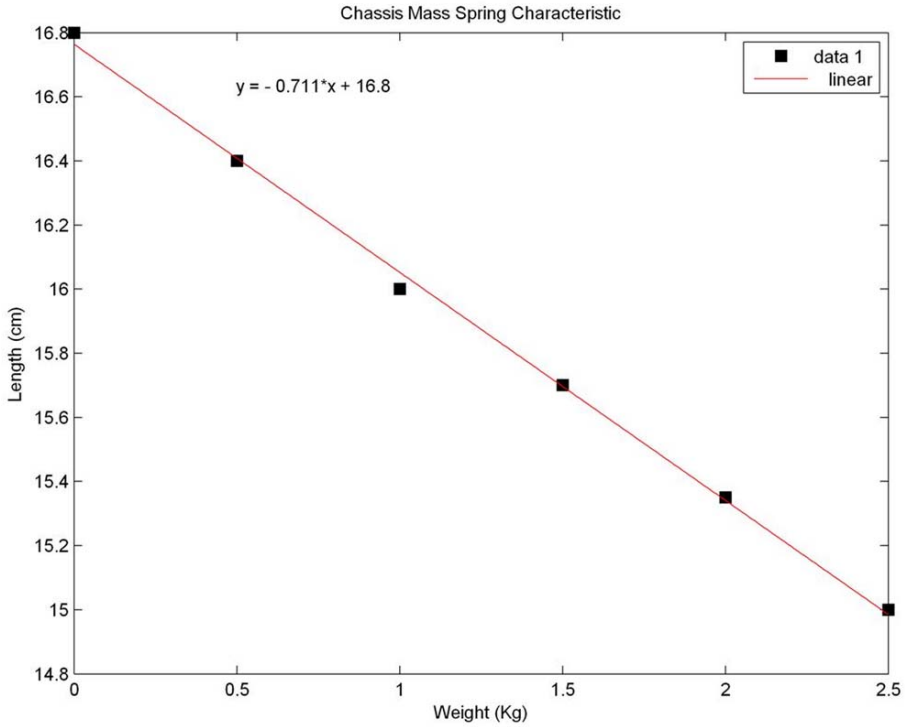

Figure 1-8: K2 Spring Constant

The two spring constants K1 and K2 can be calculated using equations 1.14 and equation 1.15 respectively (recall that acceleration due to gravity, g, is 9.8 m/s^2).

$$K1 = \frac{1Kg * (g)}{(0.711cm)} * \frac{10^2 cm}{1m} = \frac{1{,}378\ N}{m} \tag{1.14}$$

$$K2 = \frac{1Kg * (g)}{(0.613cm)} * \frac{10^2 cm}{1m} = \frac{1{,}598\ N}{m} \tag{1.15}$$

1.3.3 Damping Coefficients

If the system is excited, it will oscillate sinusoidally with exponentially decaying amplitude. The time constant of the exponentially decaying amplitude is related to the mass and damping coefficient by equation 1.16.

$$\tau = \frac{\mathrm{d}}{2\mathrm{m}} \tag{1.16}$$

Each damping constant, D1 and D2, was measured independently by forcing the respective mass to oscillate and observing the resulting waveform from the coils on an oscilloscope. D1 and D2 represent the damping coefficients acting on the Suspension Mass and the Chassis Mass respectively. The oscilloscope data was imported into Matlab for analysis. The formula for the exponential envelope is given by the equation 1.17.

$$y(t) = A * e^{-\tau * t} \tag{1.17}$$

Taking the natural log of both sides yields:

$$\ln\big(y(t)\big) = \ln(A) - \tau * t \tag{1.18}$$

This shows that if the natural log of y(t) is plotted, it will appear as a straight line with slope -τ. y(t) for the Chassis and Suspension Masses is shown in Figures 1-9 and 1-11 respectively, while ln(y(t)) with added best fit line is shown in Figures 1-10 and 1-12.

The natural log plots yield:

$$\tau_s = 3.389 \tag{1.19}$$

$$\tau_c = 5.528 \tag{1.20}$$

$$D1 = 2 * m_s * \tau_s = 11.122 \ \frac{Nm}{s} \tag{1.21}$$

$$D2 = 2 * m_c * \tau_c = 17.79 \ \frac{Nm}{s} \tag{1.22}$$

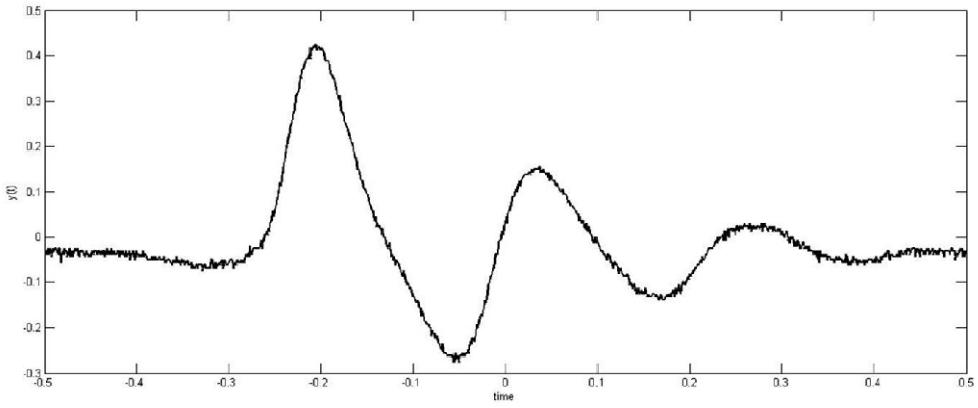

Figure 1-9: Decaying Sinusoid (y(t)) for Chassis Mass

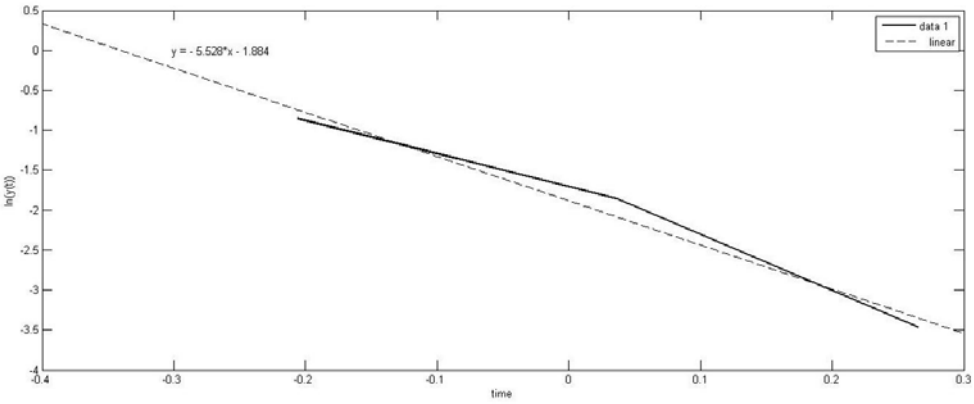

Figure 1-10: ln(y(t)) for Chassis Mass

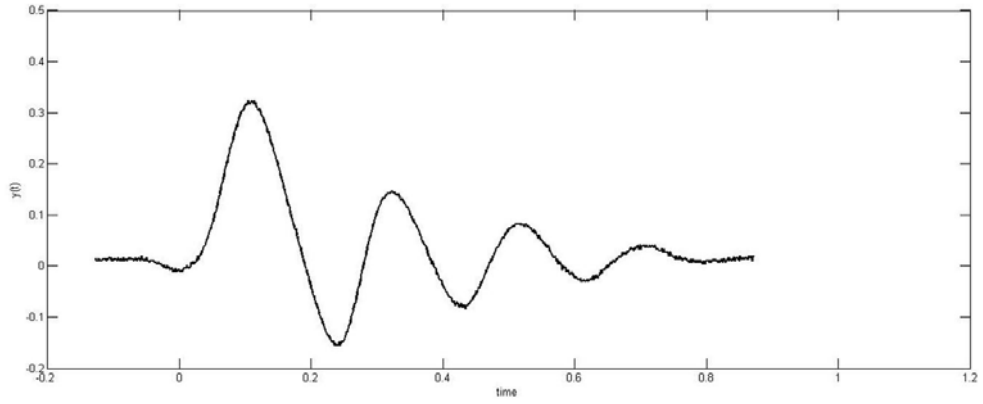

Figure 1-11: Decaying Sinusoid (y(t)) for Suspension Mass

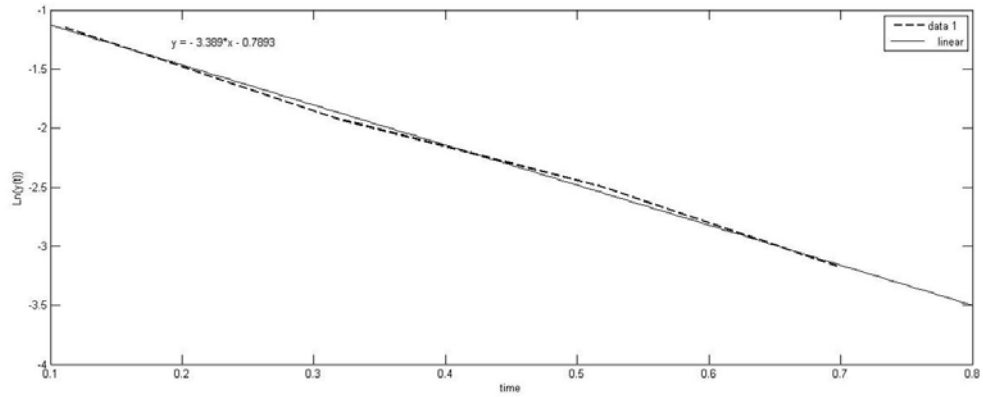

Figure 1-12: ln(y(t)) for Suspension Mass

2 Modeling of the Suspension System

2.1 Introduction

Recall the differential equations describing the mechanical system.

$$F(t) = f_c + M_S\ddot{x} + D_1\dot{x} + K_1x + D_2(\dot{x} - \dot{y}) + K_2(x - y) \tag{2.1}$$

$$F(t) = M_S\ddot{x} + D_1\dot{x} + K_1x + M_c\ddot{y} \tag{2.2}$$

This equation can be viewed as a differential equation describing an electrical circuit, using the "force-voltage" mechanical-electrical analogy. In this comparison, voltage represents force, current represents velocity, damping represents resistance, mass represents inductance, and spring constants represent capacitance. The schematic representing the suspension system is shown in Figure 2-1.

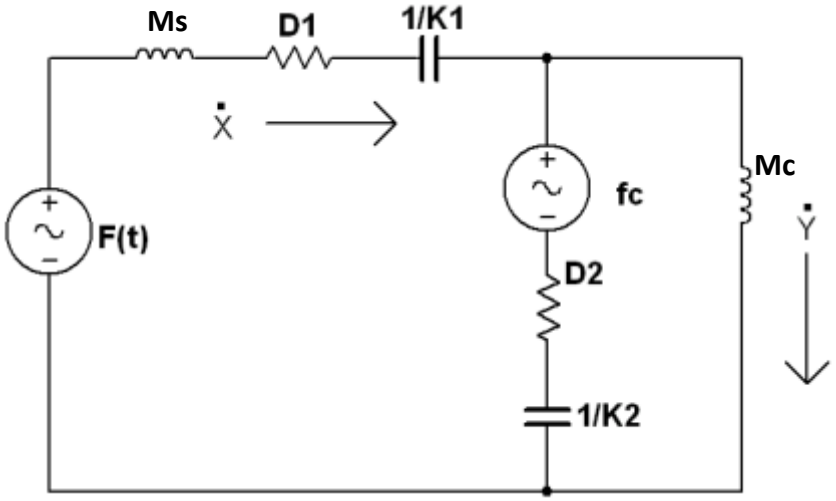

Figure 2-1: Equivalent Electric Circuit

2.2 Frequency Response

A critical factor in the operation of the suspension system is the system's response to different frequencies of oscillation. The effects of the various masses, spring constants, and damping cause certain resonant frequencies where the amplitude of oscillation is dramatically larger than for other frequencies. The electrical equivalent circuit for the mechanical system shown in Figure 2-1 can be simulated in PSPICE to predict the frequency response of this system. Figure 2-2 shows the result of this simulation, which shows the velocity of the Chassis Mass at various frequencies. According to this simulation, the resonant frequency of the system occurs at 2.92 Hz, while the response is significantly less dramatic for the rest of the frequency range. The code for this simulation can be found in Appendix B.

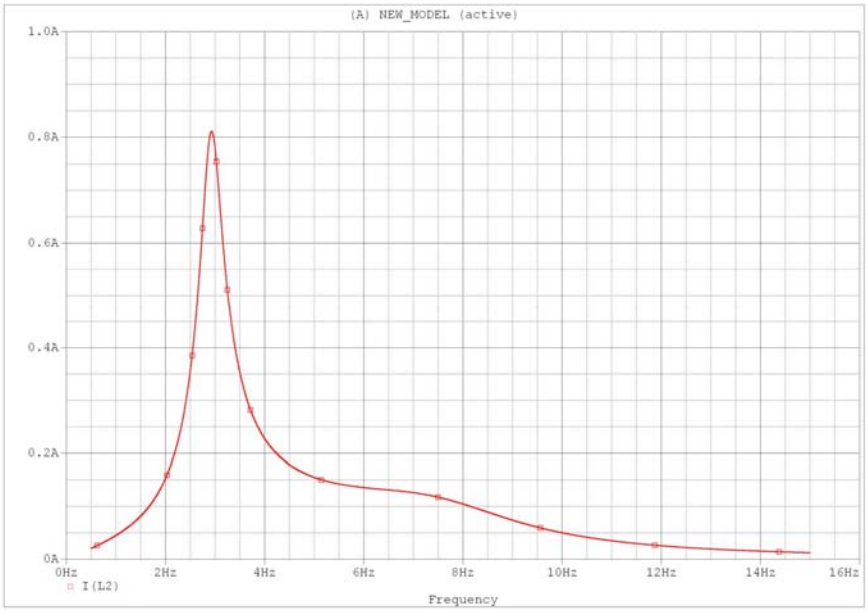

Figure 2-2: Simulated Frequency Response of Chassis Mass

After the actual system was made to oscillate, it was possible to obtain a crude plot of frequency response using Coil 2 on the Chassis Mass to detect the movement of the mass. The frequency of excitation was increased in steps of 1Hz up to 10Hz, while the amplitude of the voltage on Coil 2 was measured. This method is not entirely reliable, and a better response curve will be attained using either an accelerometer on the Chassis Mass or an optical sensor, however this provides a simple and quick means of comparing the real response to the predicted response.

Recall that for a time varying magnetic field in a coil, the voltage is given by equation 2.3 [2].

$$V = N\frac{d\Phi}{dt} \tag{2.3}$$

Where N is the number of turns in the coil, and Φ is the flux within the coil. As the system is excited sinusoidally, it's assumed that Φ is also sinusoidal, giving:

$$V = N * A_{\Phi} * 2\pi\mathrm{f} * \sin(2\pi f t) \tag{2.4}$$

As it's assumed that the speed of the mass is sinusoidal, the magnitude of the speed is proportional to the voltage induced in the coil:

$$U_{chassis} \propto N * A_{\Phi} * 2\pi f \tag{2.5}$$

Therefore, recording the amplitude of the voltage on Coil 2 at various frequencies of excitation will yield an approximate frequency response curve. This measured data is shown in Figure 2-3 below.

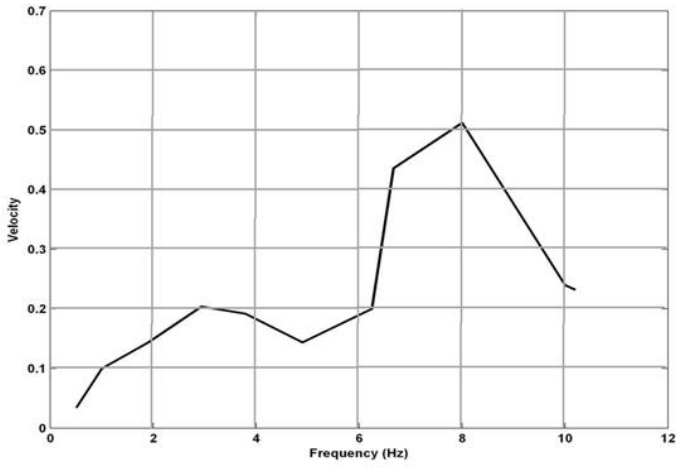

Figure 2-3: Measured Chassis Mass Velocity Frequency Response (arbitrary units)

It is immediately noticeable that the simulation and real response are quite different. The measured response shows that the system has two resonant frequencies, while the simulation only indicates one. The first peak in the measured response occurs at 2.9 Hz, which agrees with the simulation. Upon close visual inspection of the system at the second resonant frequency, it appears that the Chassis Mass does not actually oscillate significantly. At approximately 8Hz, the Suspension Mass oscillates violently, while the Chassis Mass is relatively steady. Because the voltage in the coil is dependent on both the movement of Coil 2 and the movement of the

magnet on the Suspension Mass, the oscillations of the Suspension Mass are included in the measured frequency response.

Essentially, the Chassis Mass and the Suspension Mass have separate resonant frequencies, and when the Suspension Mass hits its resonant frequency, the velocity measurements of the Chassis are offset dramatically. After observing this behavior, it can be concluded that the previous conclusion regarding $U_{chassis}$ is false. In reality,

$$U_{chassis} - U_{suspension} \propto N * A_{\Phi} * 2\pi f \tag{2.6}$$

Hence, the simulation is redone to show the difference in $U_{chassis}$ and $U_{suspension}$, rather than just $U_{chassis}$. The resulting response curve is shown in 2.4 below.

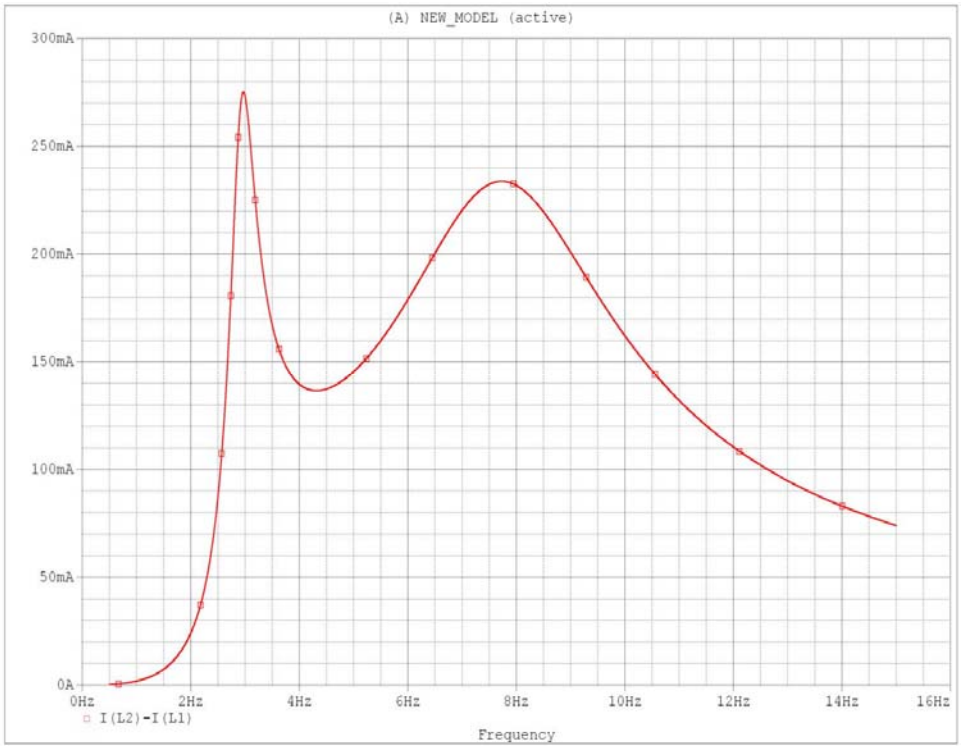

Figure 2-4: Simulated Frequency Response for Difference of Mass Speeds (m/s)

This curve is now quite similar to the measured response curve. While the amplitude spikes do not appear entirely proportionate, the resonant frequencies are nearly identical. While it will still be beneficial to create a measured response curve for the Chassis Mass alone, this combined response indicates that the physical measurements of the system were quite accurate.

2.3 Modeling Coil 1 as an Inductor

In this part of the project, Coil 1 is modeled as an inductor to find the inductance of the coil. The internal resistance of the coil, measured using a multi-meter, is 300mΩ. The circuit in Figure 2-5 was used to calculate the inductance of the coil.

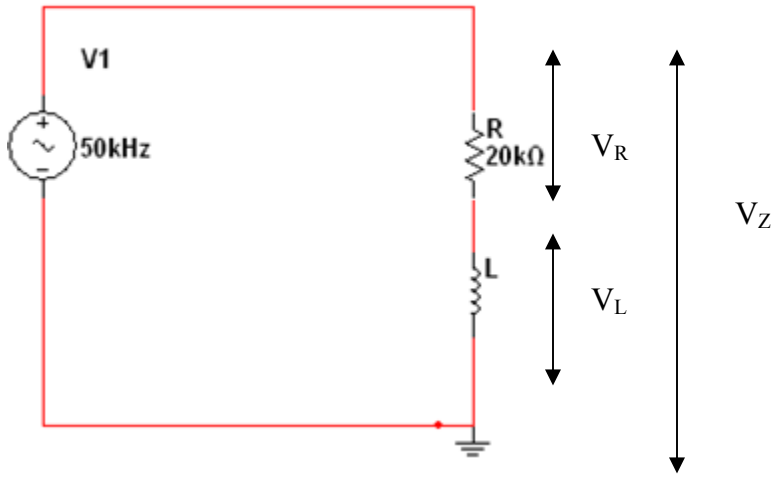

Figure 2-5: Circuit for Modeling Coil as Inductor

Coil 1, modeled as an inductor with an inductance L, is connected in series with a resistor, R, where R is chosen to be 20kΩ. The resistor is connected in series with a function generator outputting a sine wave of frequency 50 kHz.

$$V_R = Voltage\ across\ resistor \tag{2.7}$$

$$V_L = voltage\ across\ the\ coil\ in\ V \tag{2.8}$$

$$V_Z = Total\ voltage\ across\ resistor\ and\ coil \tag{2.9}$$

$$V_Z = V_R + V_L; \tag{2.10}$$

The characteristic formula for the impedance of an inductor is given in equation 2.11.

$$Z_L = wL \tag{2.11}$$

$$w = angular\ frequency\ in \frac{rad}{s}; L = inductance\ in\ H \tag{2.12}$$

$$w = 2\pi f = 2 * \pi * 50 * 10^3 = 3.14 * 10^5 rad/s \tag{2.13}$$

The total impedance of the circuit due to the resistor and the inductor is:

$$Z_T = \sqrt{(wL)^2 + (R)^2} \tag{2.14}$$

The value of the resistor was chosen such that:

$$R \gg wL \; and \; thus \; Z_L \; is \; negligible \tag{2.15}$$

This approximation, it is concluded that:

$$Z_T = \sqrt{(R)^2} \approx R \tag{2.16}$$

From the circuit in Figure 2-5:

$$V_Z = V_R + V_L; \; and, \frac{V_L}{V_Z} = \frac{wL}{Z_T} \tag{2.17}$$

Therefore the equation for calculating the value of the inductance narrows down to:

$$L = \frac{V_L R}{V_Z w} \tag{2.18}$$

The inductance is based on the voltage across the inductor and the voltage across both the resistor and the inductor.

An oscilloscope is used to calculate the peak to peak voltage across the inductor and across both the resistor and the inductor.

The following data was collected while using the oscilloscope.

$$V_Z = 7.04V(pk-pk); \; V_L = 66mV(pk-pk) \tag{2.19}$$

The resistance and the angular frequency are known, and thus the inductance of the coil was calculated using the following equation.

$$L = \frac{V_L R}{V_Z w} = \frac{0.066 * 20 * 10^3}{7.04 * 3.14 * 10^5} = 597 * 10^{-6} H \cong 600\mu H \tag{2.20}$$

The same experiment was repeated for accuracy. The second time the value of the resistor, R, was increased to 25kΩ, and the following data was collected:

$$V_Z = 7.2V(pk-pk);\ V_L = 54mV(pk-pk) \tag{2.21}$$

The value of the inductor was again calculated using the same formula:

$$L = \frac{V_L R}{V_Z w} = \frac{0.054 * 25 * 10^3}{7.2 * 3.14 * 10^5} = 597 * 10^{-6} H \cong 600\mu H \tag{2.22}$$

3 Road Condition Module

3.1 Introduction

The purpose of this part of the project is to simulate road conditions. The Road condition module consists of a function generator circuit that generates sine waves. The frequency and the amplitude are tunable. The signal from this oscillator is amplified using a power transistor to drive Coil 1. The force from Coil1 drives the Suspension Mass. The block diagram for the road condition module is depicted in Figure 3-1.

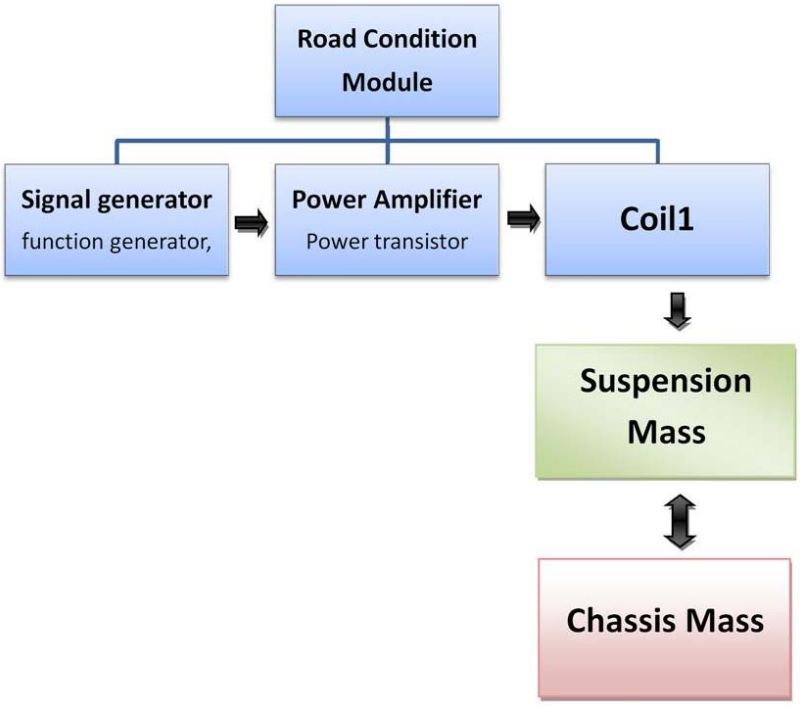

Figure 3-1: Road Condition Block Diagram

3.2 Power Supply

The main purpose of this design is to apply varying voltages to Coil 1, which in turn causes Ms to oscillate. The lab power supply is not equipped to provide enough current required for the system to work. The lab power supply has a maximum current supply of 6A, and, for design purposes, more current is required for the functionality of the system. A commercial power supply with a large amount of maximum supply current is expensive, and thus designing a power supply would make the overall modeling of the suspension system more cost effective.

The supply required to power Coil 1 uses a variable transformer, T1, making it easy to control the supplied voltage. The variable transformer used in this project in this particular module is manufactured by POWERSTAT. It is of 136B series and is designed to deliver continuously adjustable voltage from AC power lines. Figure 3-2 below shows the picture of the transformer used.

Figure 3-2: POWERSTAT Transformer

The power supply is configured to produce an output voltage between 0-30V when drawing power from a 120V wall outlet. The main purpose for the power supply is to supply sufficient current for Coil 1 to force the Suspension Mass to oscillate. The desired output voltage was achieved by connecting the primary and secondary coils of the transformer as depicted in the manufacturer's datasheet.

The transformer output is connected to a rectifier circuit and a large capacitor to provide a steady supply voltage. The rectifier used in the circuit was a bridge rectifier package MP502W-BPMS-

ND mounted on an aluminum heat sink. The rectifier bridge is capable of handling 50A current and a maximum voltage of 200v. The capacitor connected has a capacitance of 27mF and a maximum voltage rating of 75Vdc.

3.3 Oscillator Circuit

3.3.1 Full Schematic

The full schematic for the oscillator circuit built is shown in Figure 3-3. The oscillator module consists of a low power sine wave generator and a power amplifier. The amplified signal from the power amplifier drives Coil1.The module has the capability of varying the output frequency, amplitude, and offset. The output of this circuit is then amplified with a BJT to drive the power transistor.

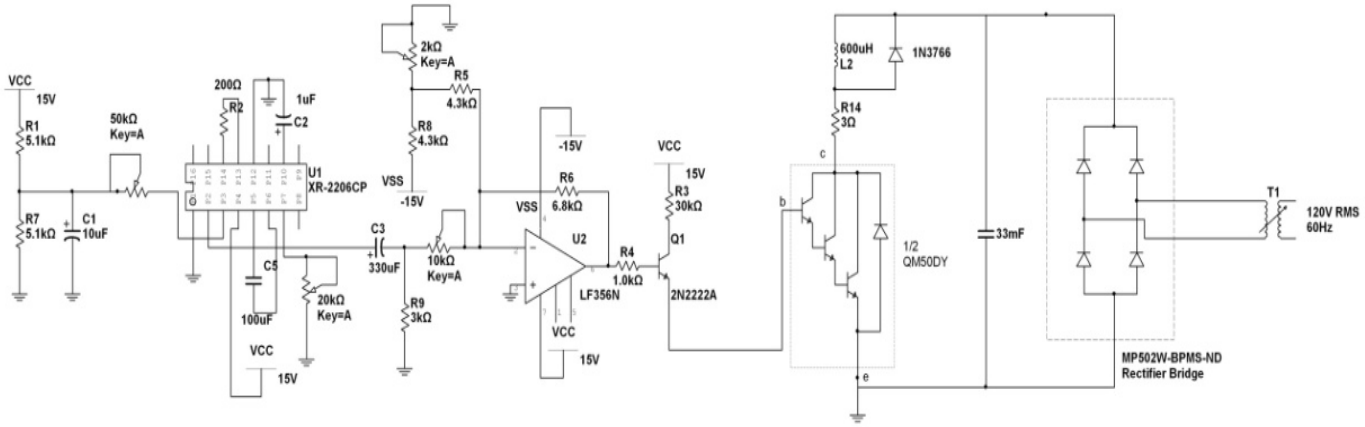

Figure 3-3: Oscillator Circuit

3.3.2 Detailed Analysis of Oscillator Circuit

3.3.2.1 Function Generator

At the heart of the oscillator circuit is the XR-2206CP integrated circuit. The chip is a monolithic function generator which can produce high quality sine, square, triangle, ramp and pulse waveforms which can be both frequency and amplitude modulated. The frequency can be varied from .01 Hz to 1Mhz. While the amplitude of the function generator output signal is adjustable, it rides on a DC offset which cannot be adjusted. To control the power transistor properly, both the signal amplitude and offset at the base of the transistor need to be controllable. To accomplish this, the function generator signal is run through a high pass filter to remove the offset, and then passed through an inverting Op-Amp configuration. This Op-Amp stage also

includes a controllable offset. Both the amplitude and offset of the signal can then be adjusted with separate potentiometers.

3.3.2.2 Adder Circuit

The next stage is a simple Op-Amp adder circuit. This allows an offset to be added to the output with the sine wave. The 2KΩ resistor is used to vary the voltage that gets added as an offset to the sine wave output. The gain of the sine wave can also be varied with the 10kΩ variable resistor. Hence this allows control of the offset and the magnitude of the sine wave independently.

3.3.2.3 Power amplifier

The power transistor used in the circuit required a relatively high input current. A BJT was used to control the current flowing into the base of the power transistor. The current through the base of the power transistor QM50DY is set by the 30KΩ resistor connecting VCC to the collector of the BJT. The QM50DY is used to drive Coil 1 through a 3Ω resistor. It has a maximum collector current of 50A and a maximum collector-emitter voltage of 600V. The expected current drain of the load (Coil 1) is 30A.

One of the design constraints typically encountered with transistors is that they get very hot after prolonged use. This problem can be solved by installing the transistor on a heat sink, with thermal paste in between to increase thermal conductivity for better heat dissipation. Figure 3-4 shows the image of the transistor mounted on the heat sink.

Figure 3-4: Power Transistor QM50DY

3.3.3 Test

The outputs from different parts of the oscillator circuit were viewed on an oscilloscope. Figure 3-5 shows the sign wave generated from the function generator. CH1 is connected to pin 2 of the XR-2206CP IC. Figure 3-6 shows the voltage across the load resistor.

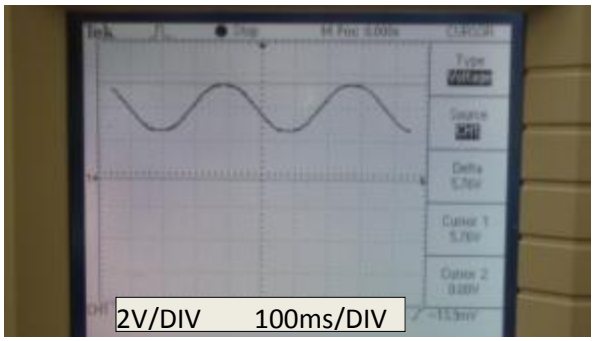

Figure 3-5: Voltage Output of the Function generator chip XR-2206CP

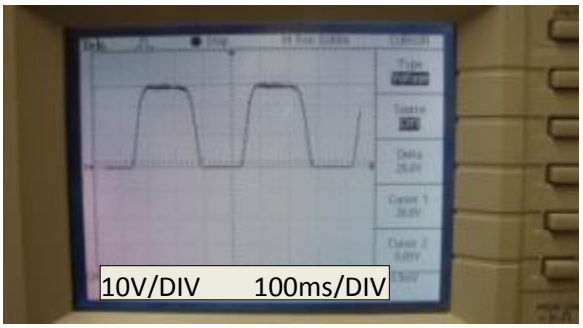

Figure 3-6: Voltage across Coil1 Load Resistor

4 Control System Module

4.1 Introduction

One of the primary objectives of the project is to design the system in such a way, that when force is applied to the body of a car, the body of the car remains stationary. The same phenomenon, when applied to the prototype of the suspension system, would keep the Chassis Mass stationery. This is achieved by implementing a control system that can compensate for disturbing forces. The control system is comprised of a sensor for feedback, a microcontroller to process the feedback signal, and an H-Bridge to deliver power to Coil 2. The detailed block diagram for the entire system, incorporating the control system, is shown in Figure 4-1.

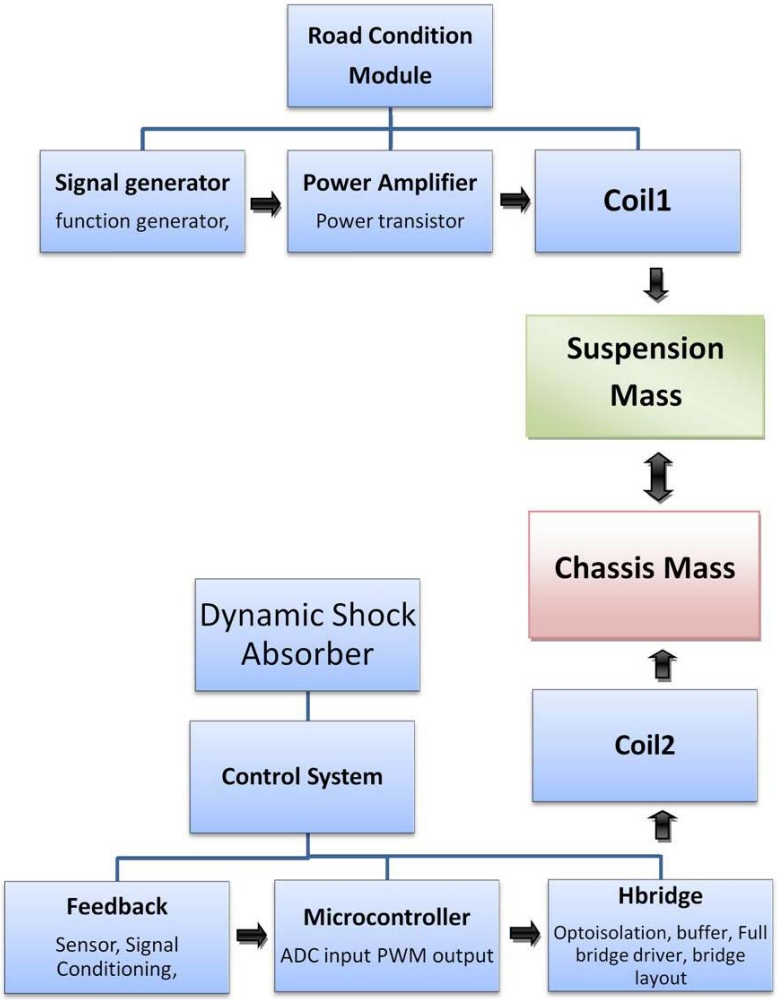

Figure 4-1: Entire System Block Diagram (With Control Module)

4.2 Theoretical Model

4.2.1 Design Simulations

Simulations of the electrical equivalent model of the active suspension system can be used to determine design criteria for the control system.

4.2.1.1 Control Theory

Figure 4-2 shows the electrical equivalent of the suspension system.

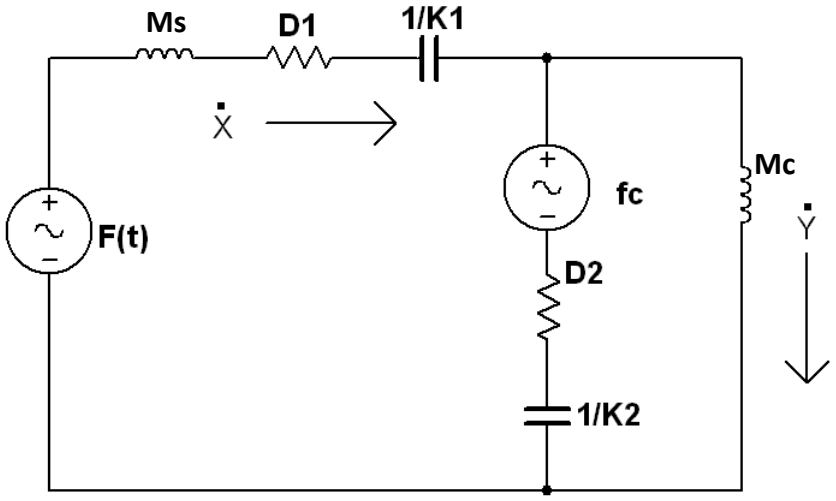

Figure 4-2: Electrical Equivalent of the System

From the schematic in Figure 4-2, it can be seen that when a voltage or a force is applied, a current goes through the Chassis Mass Mc causing it to move. The goal is to make the velocity of Mc zero.

A proportional control system was chosen after analyzing different control systems. Figure 4-3, shows the block diagram for a generic proportional controller.

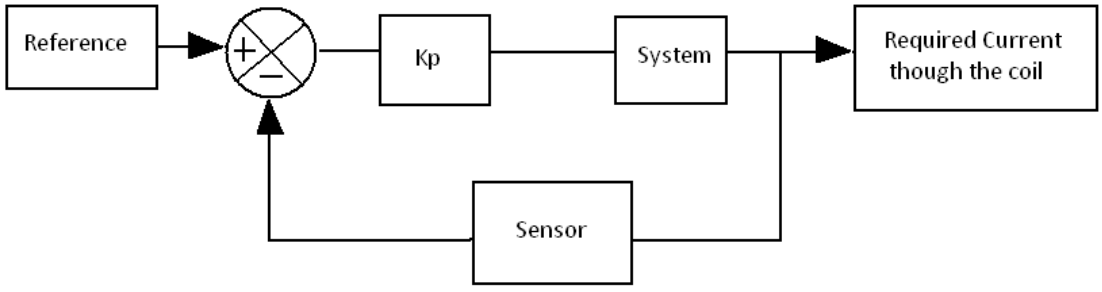

Figure 4-3: Proportional Controller

A proportional controller incorporates a control loop feedback mechanism, which essentially corrects the error between a measured variable and a desired set point. The main idea is to make sure that the current through Coil 2 is zero. This is done by making the voltage across Coil 2 zero.

4.2.1.2 Instability Study

A Bode Plot of the control system's transfer function can indicate if the system will become unstable at certain frequencies. The magnitude and phase of the transfer function are shown in Figure 4-4. The phase segment of the plot shows that the phase never reaches -180 degrees, indicating that the system is stable for all relevant frequencies. The Matlab code for these plots can be found in Appendix F.

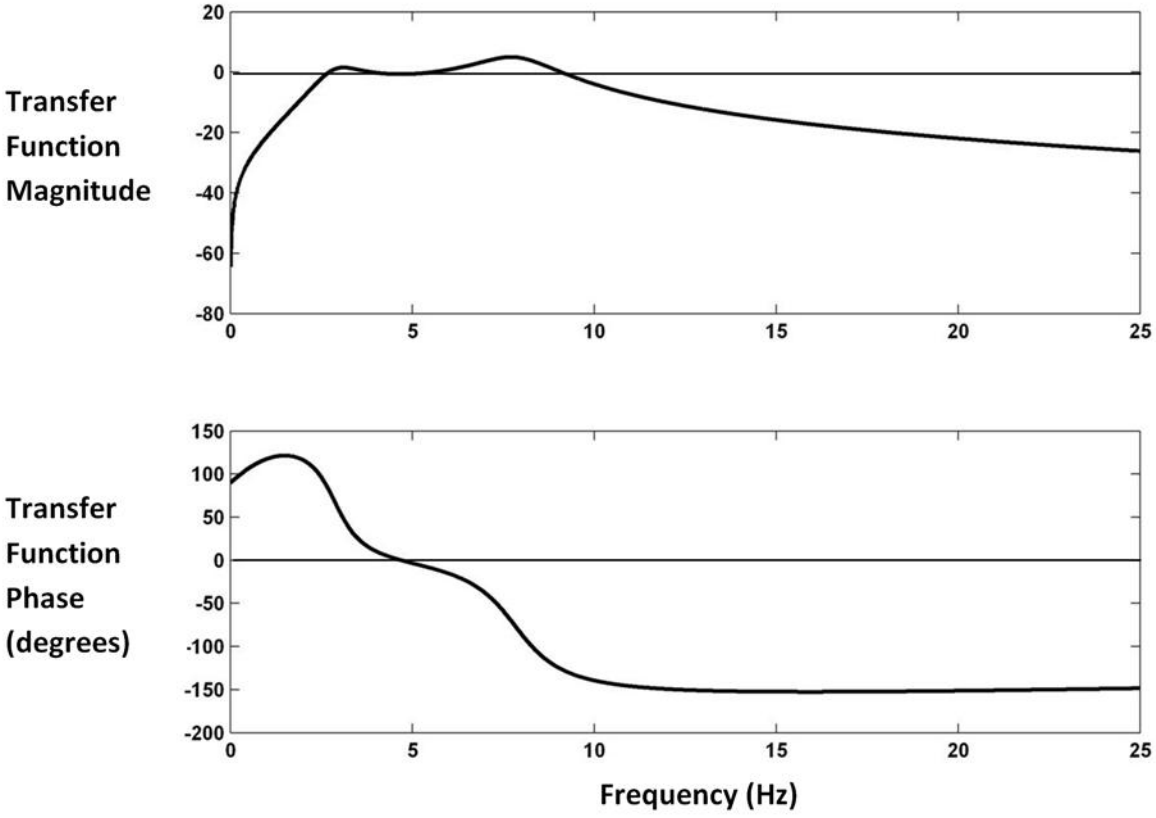

Figure 4-4: Transfer Function Bode Plot

4.2.1.3 PWM based model

A more detailed model is the one shown in Figure 4-5. From Figure 4-5, the control voltage drives Coil 2 which causes the current dependant voltage source F(c) to vary. This model is more accurate as this includes the delay of change in fc due to the inductance in Coil 2. This delay is one contributor to the overall delay in the feedback loop.

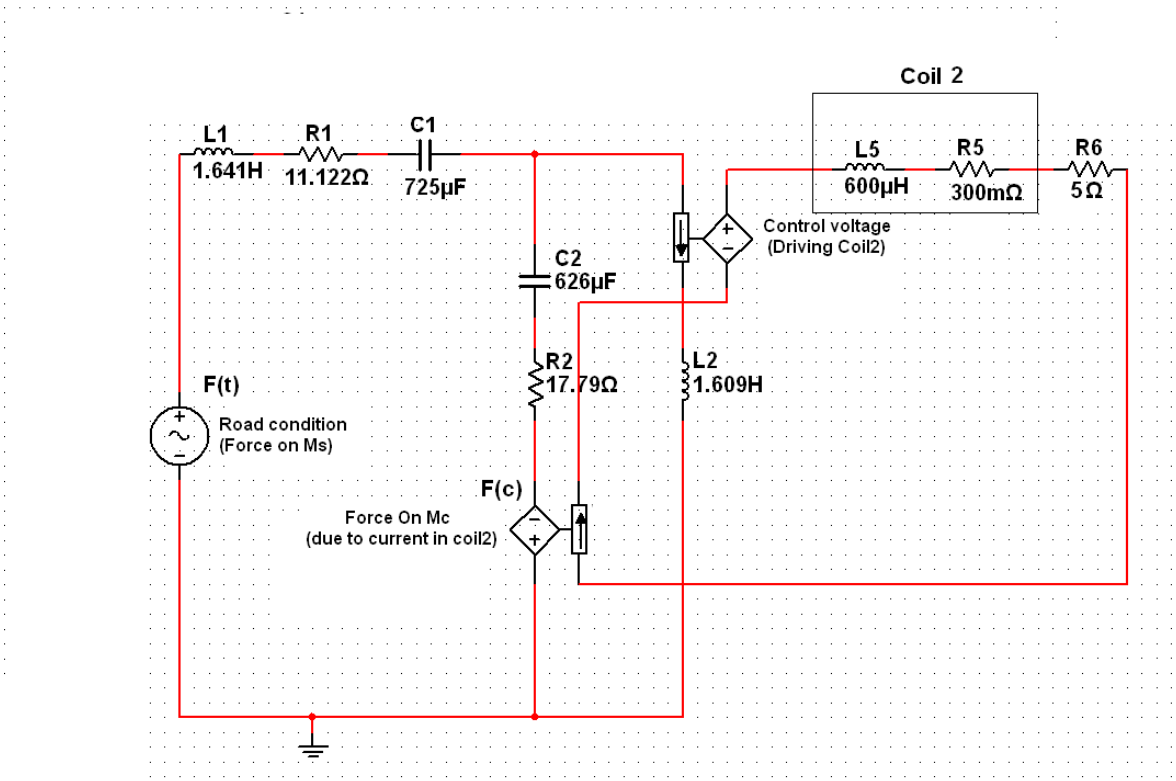

Figure 4-5: Control Circuit including Coil 2

Figure 4-5 shows the general idea of the control system. However an improvement of the model is to include the PWM based control voltage. This new model was made in PSPICE for simulation.

```
.SUBCKT PWMgen in inref out outref
Vtri tri inref PULSE(-1.1 1.1 .01m 1.24m 1.24m .01m 2.5m)
*1.1 keeps duty cycle between 10 and 90 %
Eout out outref TABLE {(V(dsh,inref)-V(tri,inref))}=(-.1u -1) (.1u 1)

Sit in sr tri 0 Switchit
Rsh sr dsh 1m
Csh dsh inref 1u IC=0
.MODEL Switchit VSWITCH(Ron=1m Roff=1G Von=1 Voff=.99)
.ENDS
```

This sub circuit creates a 200Hz triangle waveform. This waveform has two purposes. First, it triggers a switch to operate a sample and hold operation on the input signal (duty cycle), which simulates the functionality of the digital feedback loop. Secondly, it is used to compare against the duty cycle signal to generate the PWM.

```
.PARAM Kp = 1
.STEP PARAM kp list  9  2000
.PARAM Vsup 100
```

These are the supply voltage and gain variables.

```
V 1 0 Sin(0 5 2.9)
L1 1 2 1.641 IC=0
R1 2 3 11.122
C1 3 4 725u IC=0
C2 4 5 626u IC=0
L2 4 7 1.609 IC=0
Vvelocity 7 0 0
Vdcoil2 11 0 0
R2 5 6 17.79
L3 8 9 600u IC=0
Rcoil2 9 10 300m
Rcoil2series 10 11 3
Eforce 6 0 VALUE={.005*I(Vdcoil2)*I(Vdcoil2)*I(Vdcoil2)+.8*I(Vdcoil2)}
Ebridgepwm 8 0 VALUE={-Vsup*v(B)}


Eduty duty 0 TABLE {kp*I(L2)} = (-1,-1) (1,1)
Xpwm duty 0 B 0 PWMgen


Emotionfilter mf 0 VALUE={1*I(L2)}
Rmf mf mfout 1k
Cmf mfout 0 16u

.Probe
.TRAN 1 1 0 1m UIC
.END
```

This section of code simulates the mechanical portions of the system, i.e. masses, spring constants, damping constants and forces.

Eduty is the product of kp and the velocity of the Chassis Mass, but this product is constrained to be between -1 and 1 so that it can be used to produce the PWM

4.2.1.4 Attenuation Limitations

In this system, the force opposing movement of the Chassis Mass is proportional to the velocity of the Chassis Mass. Ideally, the gain of this feedback system would be increased until the attenuation was close to 100%. However, multiple factors impose limits on how high the gain can actually be. If the gain is made too high, the system becomes unstable, and the control system amplifies the oscillations of the Chassis Mass.

One of the most significant factors that limit the gain of the system stems from the method in which the microcontroller creates the PWM signal. At the beginning of a cycle, the microcontroller samples a value from the sensor, does the necessary calculations, and produces a pulse of the necessary duration. Any changes in velocity that occur before the next pulse is started will essentially be ignored. Therefore, if the PWM frequency is 200Hz, then the delay between a change in velocity and the corresponding correction from the PWM can be up to 5ms.

Two variables in this system affect the signal gain. The first is the multiplying factor K_p, which is a value within the microcontroller, and the second is the supply voltage. The following

simulation results demonstrate the maximum possible attenuation for different supply voltages, as well as visual evidence of instability due to increased gain.

The simulation was conducted considering a 5 Newton sinusoidal force perturbing the Suspension Mass, at a frequency of 2.9Hz. Each supply voltage was simulated separately with a wide range of K_p values.

Supply Voltage	Maximum Possible Attenuation
20	53%
30	91%
40	89%
50	94%
75	95%
100	91%

Table 4-1: Maximum Possible Attenuations for Various Supply Voltages

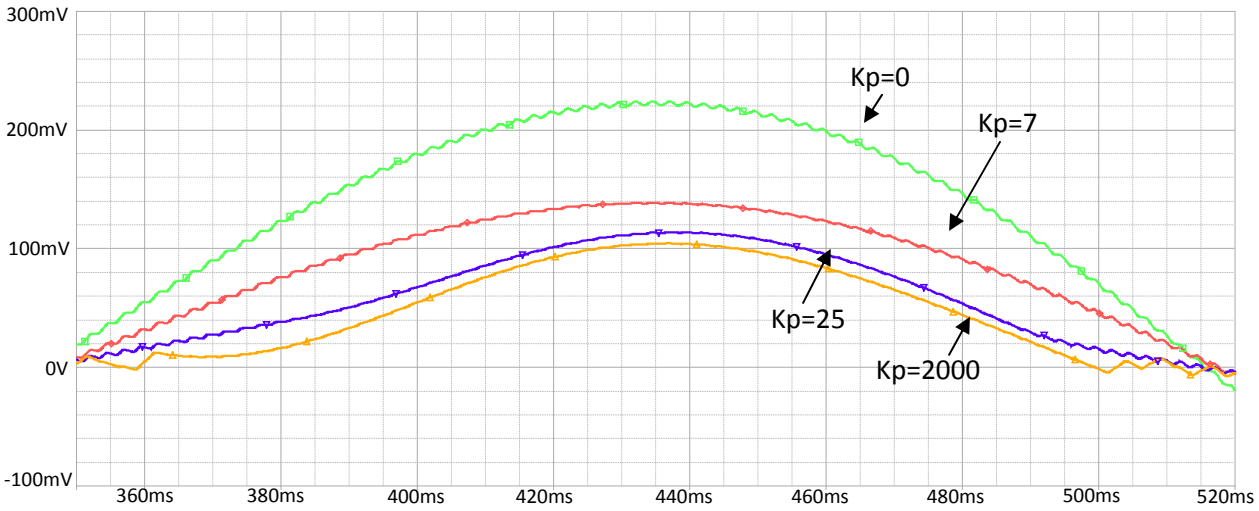

Figure 4-6: Velocity of Chassis Mass (in m/s) at Different Values of Kp; Vs=20V

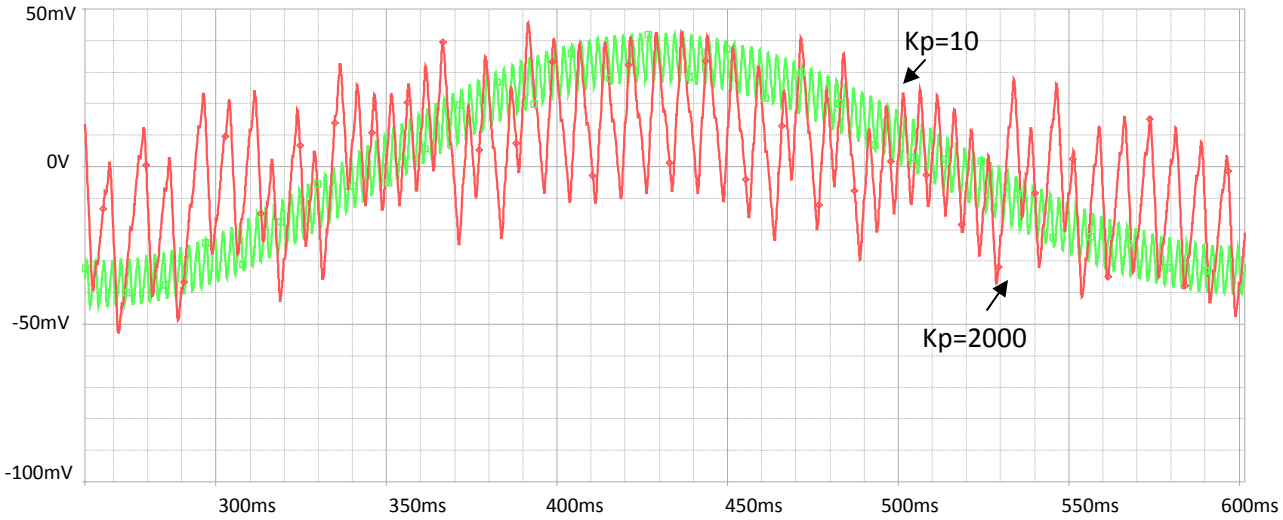

Figure 4-7: Velocity of Chassis Mass (in m/s) at Different Values of Kp; Vs=50V

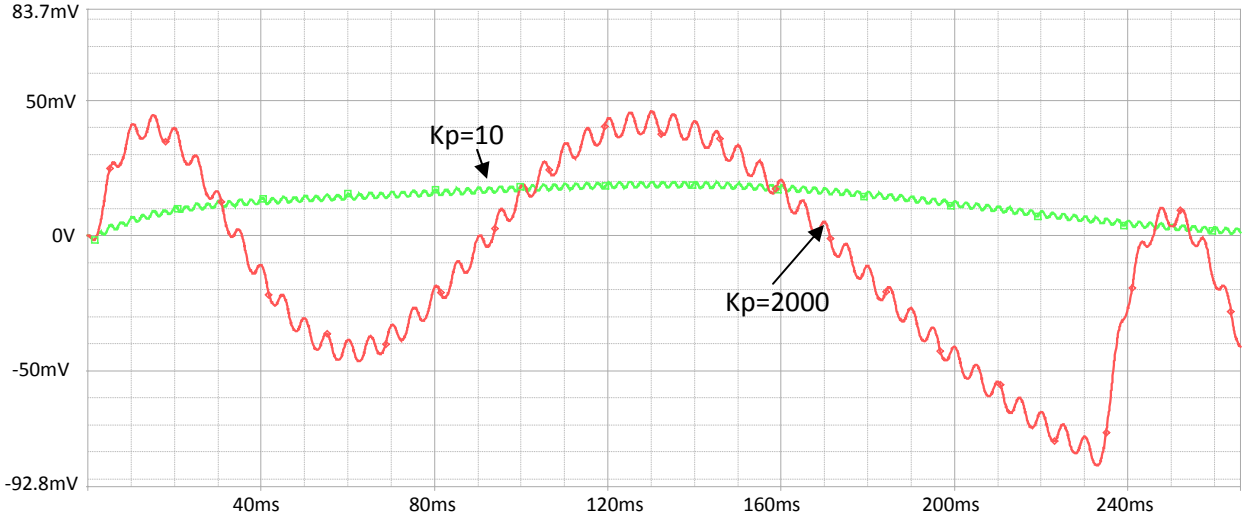

Figure 4-8: Velocity of Chassis Mass (in m/s) at Different Values of Kp; Vs=100V

Table 4-1 shows that the attenuation of this system doesn't exceed 95%. It's important to note that these are theoretical limits. In actuality, noise inherent in the system, as well as additional propagation delays can further lower the allowable gain value.

One important thing to notice is that for a given supply voltage, there is a diminishing return on increased K_p. In Figure 4-6, there is a large difference in the amplitude of Chassis Mass velocity between $K_p = 0$ (no control) and $K_p = 7$. Between $K_p = 25$ and $K_p = 2000$, there is very little difference.

In Figure 4-7, the supply voltage is increased to 50V. Here, $K_p = 10$ provides a good attenuation, but the larger $K_p = 2000$ results in heavy oscillation at the PWM frequency. This is the first sign of instability.

Figure 4-8 shows the results for a supply voltage of 100V. $K_p = 10$ provides excellent attenuation, but the larger K_p makes the system less stable.

4.3 Feedback system

4.3.1 Introduction

In this section, sensors are analyzed thoroughly to choose a suitable sensor for detecting changes in the motion of Mc.

4.3.2 Sensors

The microcontroller, a MSP430, generates a PWM control signal to counter the force expereinced by Mc. To do this the Microcontroller requires an input signal proportional to Mc's velocity. There is a wide range of sensors that could be used for this purpose. The two sensors considered for this task were the Sharp GP2D12 distance sensor, and the MMA7260Q 3 axis accelerometer chip. This decision was based on the reaction time of the sensors and the range of signal inputs.

4.3.2.1 Distance Sensor

Figure 4-9 shows the distance sensor and its characteristic output pattern [3].

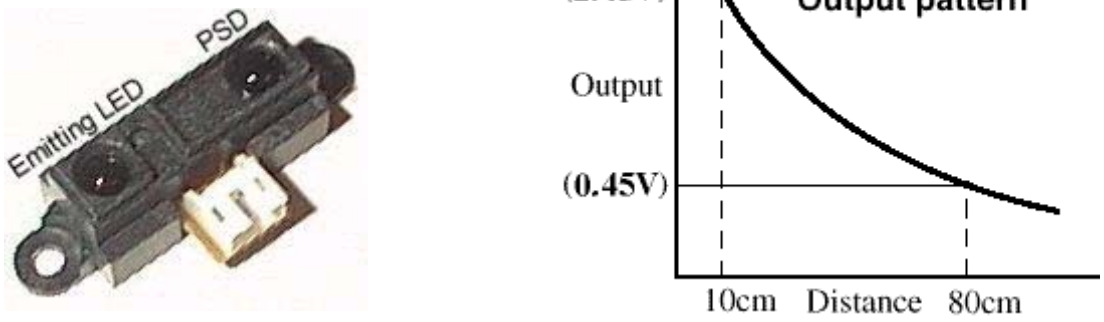

Figure 4-9: Distance Sensor with Characteristic Output Pattern [3]

4.3.2.1.1 Theory of operation

The GP2D12 is a distance sensor with non linear analog output. The sensor uses triangulation and a small linear charged coupled device (CCD) array to compute the distance of objects in the field of view [3]. The sensor contains IR LEDs that are pulsed with an oscillator. This light travels out in the field of view and either hits an object or just keeps going. In the case of no object, the light is never reflected and the reading shows no object. If the light reflects off an

object, it returns to the detector and creates a triangle between the points of reflection and the emitter, and the position sensing photodiode (PSD). The angles in this triangle vary based on the distance to the object. The PSD of the sensor is actually a precision lens that transmits the reflected light onto various portions of the enclosed linear CCD array based on the angle of the triangle described above [3]. The CCD array can then determine at what angle the reflected light came back. The signal processing circuit then filters out and calculates the distance of the object and produces an analog output. This method of ranging is almost immune to interference from ambient light and offers amazing indifference to the color of object being detected. However there is significant noise on the output due to the current it draws during firing the LEDs. This can be reduced by adding a large capacitor (at least 100uF) between Vcc and Ground as close to the sensor as possible. The sensor drives IR LEDs at a rate of 1kHz. This pulse causes a spike in the Vcc as well as the Output. The noise is shown in Figure 4-10.

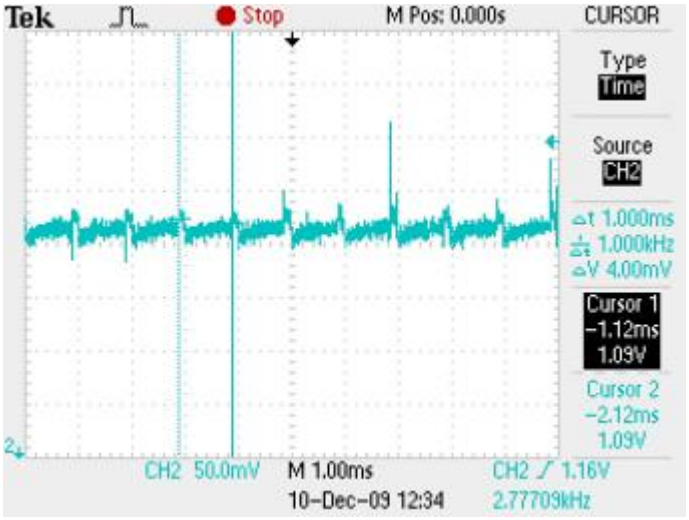

Figure 4-10: Noise at Distance Sensor Output

Even though the spikes have a very short duration, these can be lethal to the ADC. The ADC can be damaged if the input voltage is more than 3v. The observed spikes ranged from a few hundred mV to over 800mV. The manufacturer datasheet recommends a supply of 5v for this device, for which the Vout maximum is measured as 2.5v. However with these voltage spikes observed, the Vout could reach a value greater than the 3v which is the limit for ADC input and cause permanent damage to the ADC.

4.3.2.1.2 Supply Bypass Filter

Since the source of the output noise is the rapid firing of the LED, the best solution was to add a capacitor between Vcc and Ground. This connection was made on the sensor PCB itself with capacitor leads as short as possible. The idea was to connect it as close to the sensor Vcc as possible.

The picture in Figure 4-11 shows the distance sensor with the capacitor connected. An experiment was carried out to test this theory. Three sensors were powered with 5V supplies and the output was observed on the oscilloscope. Between the Vcc and Ground on two of the sensors were connected 100uf and 440uf capacitors, respectively. For the third sensor there was no capacitor connected between Vcc and ground.

Figure 4-11: Supply Bypass Filter

In the oscillogram in Figure 4-12, Ch1(yellow) shows the sensor output with 440uF capacitor, Ch2(pink) shows the output with 100uF and Ch3(blue) shows the sensor with no capacitor. It is clear from the picture that the noise reduces significantly with the increase of capacitance.

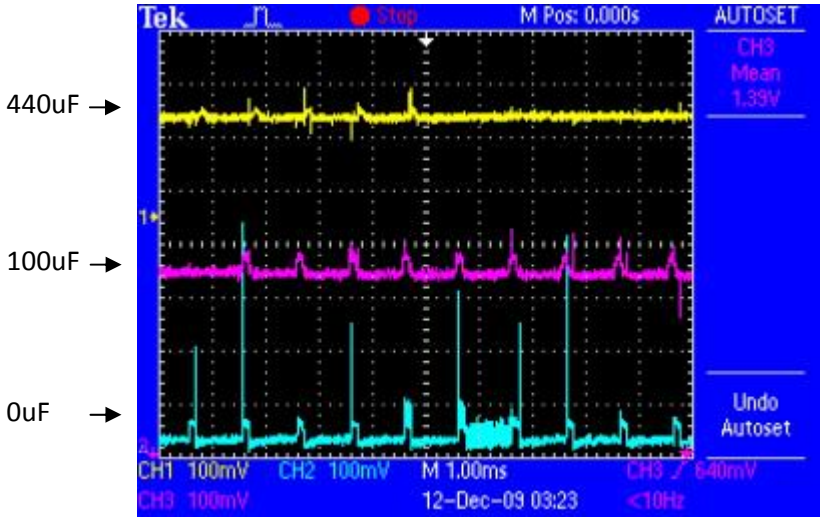

Figure 4-12: Sensor Output noises for different supply bypass capacitors

Using the cursors on the oscilloscope, the noise reductions can be compared numerically. With the 100uF capacitor the ripple is 52mv and the spikes are approximately 100mv. With a 440uF capacitor the ripples is 34mV and the spike is up to 72mV. For the purpose of interfacing with the MSP430, both of the sensors can be considered safe to connect with the ADC.

4.3.2.2 Accelerometer:

Figure 4-13 shows the Sparkfun Triple Axis Accelerometer Breakout - MMA7260Q board, which is based on the Freescale Semiconductor MMA7260Q three-axis accelerometer chip.

Figure 4-13: Triple Axis Accelerometer MMA7260Q Breakout board

The breakout board contains all the necessary external components needed for operation recommended by the chip manufacturer. Figure 4-14 shows the full schematic of the breakout board. The accelerator has an adjustable sensitivity selection option and a maximum sensitivity of 800mv/g. Since the suspension system deals with the detection of slower motion, the high sensitivity option at a low (3.3V) makes it a suitable sensor compared to other low cost accelerometers.

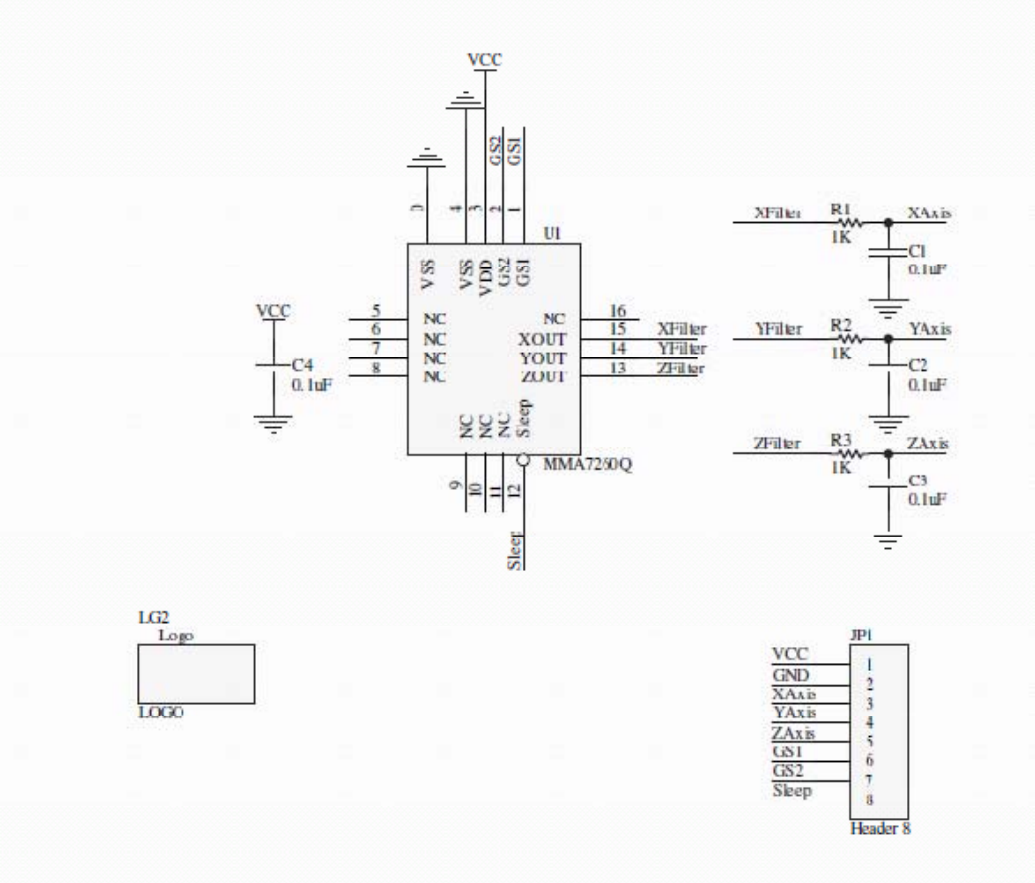

Figure 4-14: Full Schematic of the Accelerometer Breakout Board

The schematic in Figure 4-14 shows that the output for each axis has a 10kHz low pass filter. This is required as the chip has an internal oscillator operating at 10kHz. Capacitor C4 is a power supply bypass capacitor. Since power consumption is not an issue for this application, the sleep pin is connected to Vcc. The GS1 and GS2 can be connected to microcontroller I/O if software controlled sensitivity is required. Table 4-2 shows the selection options for sensitivity and acceleration measuring range.

g-Select2	g-Select1	g-Range	Sensitivity
0	0	1.5g	800mV/g
0	1	2g	600mV/g
1	0	4g	300mV/g
1	1	6g	200mV/g

Table 4-2: Table for Selecting Accelerometer Sensitivity

4.3.2.2.1 Theory of Operation:

The Freescale accelerometer is a surface micro machined integrated-circuit accelerometer. The device consists of two surface micro machined capacitive sensing cells (g-cell) and a signal conditioning ASIC contained in a single integrated circuit package. The g-cell can be modeled as a set of beams attached to a movable central mass that moves between fixed beams. The movable beams can be deflected from their rest position by subjecting the system to acceleration (Figure 4-15). As the beams attached to the central mass move, the distance from them to the fixed beams on one side will increase by the same amount that the distance to the fixed beams on the other side decreases. The change in distance is a measure of acceleration. The g-cell beams form two back-to-back capacitors. As the center beam moves with acceleration, the distance between the beams changes and each capacitor's value will change. Equation 4.1 describes the capacitance.

$$C = A\varepsilon/D, \qquad (4.1)$$

Where A is the area of the beam, ε is the dielectric constant, and D is the distance between the beams. The ASIC uses switched capacitor techniques to measure the g-cell capacitors and extract the acceleration data from the difference between the two capacitors. The ASIC also conditions and filters the signal, providing a high level output voltage that is ratio metric and proportional to acceleration.

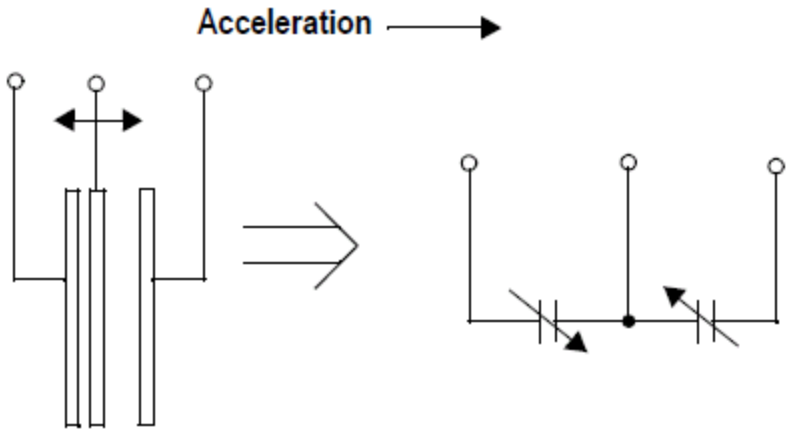

Figure 4-15: Simplified Transducer Module

4.3.2.3 Sensor Selection

The choice of the sensor was solely based on testing results. Figure 4-16 below shows the output of the distance sensor, on Channel 1, and the accelerometer, on Channel 2, at 2Hz. The distance sensor has a built-in signal processing circuit that averages 39 values, and sends an output every 39 milliseconds. This gives rise to the step like graph.

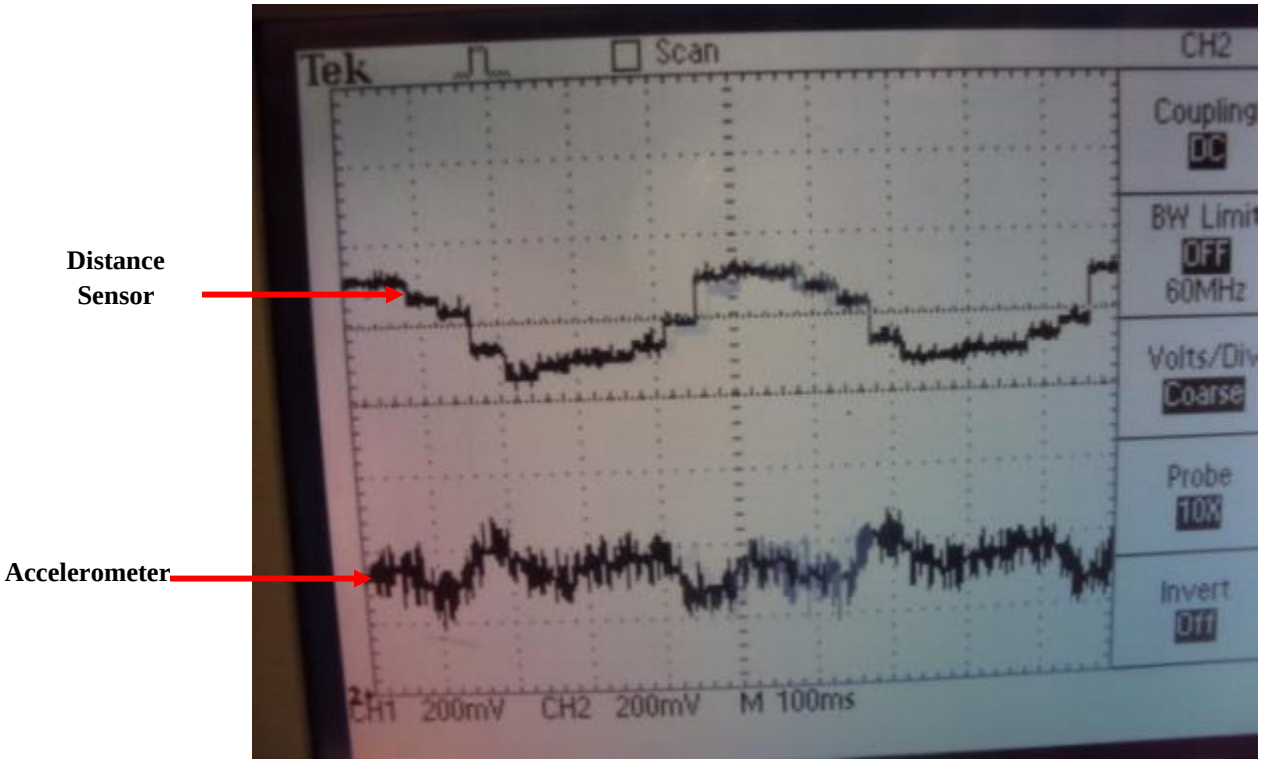

Figure 4-16: Output Signal from the Distance Sensor and Accelerometer

The same procedure was carried out, but at this time the mass, Mc, was made to oscillate at a higher frequency (7.1 Hz). These results are shown in Figure 4-17.

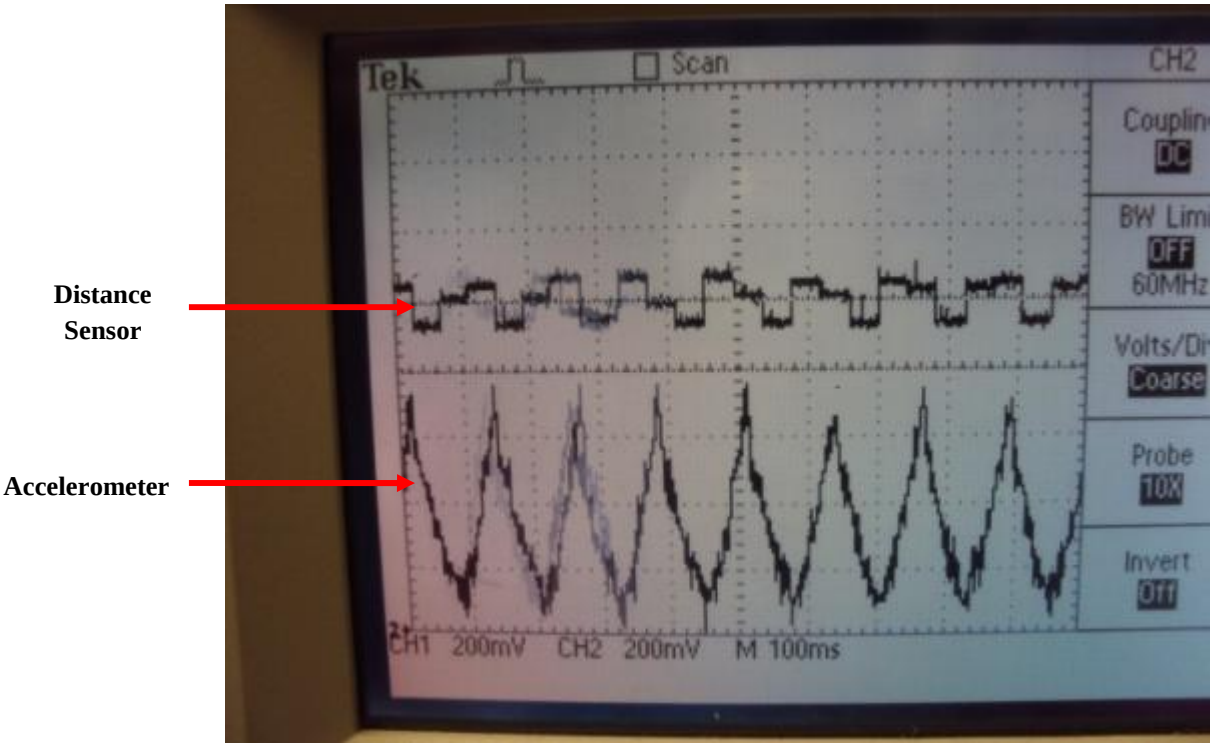

Figure 4-17: Output Signal from the Distance Sensor and Accelerometer

4.3.2.3.1 Discussion

From the two figures, Figure 4-16 and Figure 4-17, it is clearly visible that the Distance sensor works relatively well for lower frequencies. However for higher frequencies, the distance sensor output is not as linear as that of the accelerometer. Hence it could be concluded that the distance sensor is more efficient when it works for a slower motion, obstacle detection, static proximity sensing.

For the detection of a rapid motion, the accelerometer performs better. Even though using an algorithm to use a combination of both sensors could improve performance, for this project the accelerometer was chosen to be the primary sensor.

4.3.3 Signal Conditioning

The unfiltered output signal from the accelerometer has noise ranging from 50mv to 70mv. This noise becomes an issue when calculations are done by the microcontroller. Additionally, the signal amplitude varies greatly over the range of accelerations experienced by the system. Slight movement of the plate causes the accelerometer to generate a small signal, which may be overpowered by the noise. At the same time, the extreme motion of the plate at resonance forces the signal to be clipped.

The accelerometer has two logic inputs that change its sensitivity. Taking advantage of this feature, an active circuit was designed to amplify and filter the accelerometer output signal. Small signals, generated due to slight movements of the plate, are amplified by a factor of 4, thus making it easier for the microcontroller to process the signal [6]. At the same time, if the signal begins to clip, the accelerometer's sensitivity can be adjusted using the microcontroller.

The schematic for the active filter is shown in Figure 4-18.

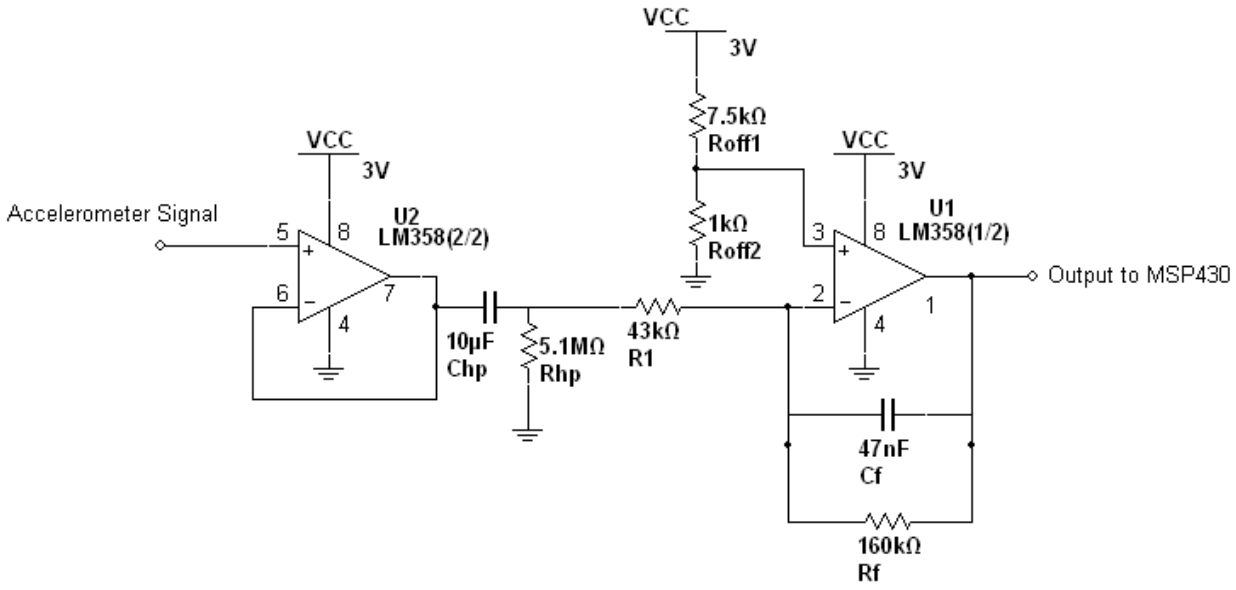

Figure 4-18: Accelerometer Signal Amplifier and Filter

4.3.3.1 Signal Amplifier

One half of a LM358 (U_2) is used to buffer the accelerometer signal. The signal is then run through a high pass filter (R_{hp} and C_{hp}) to remove the signal's offset [6]. The gain of the amplifier and the cutoff frequency of the filter are:

$$Gain = \frac{R_f}{R_1} = 3.72 \approx 4 \tag{4.2}$$

$$f_{3db} = \frac{1}{2\pi R_f C_f} = 21.2Hz \tag{4.3}$$

The gain is high enough to improve readings for low amplitude signals, and low enough to not get clipped at resonance, when the accelerometer is at its lowest sensitivity. The cutoff frequency of the filter is low enough to effectively remove high frequency noise, without infringing on the relevant frequency range of 0 to 10Hz.

4.3.3.2 Filter Simulation

This signal amplifier/filter circuit was simulated in PSPICE to test its functionality, with a sample input signal of a 400mV$_{pk\text{-}pk}$, 10Hz sine wave with a 2v offset. A 50mV, 10 kHz sine wave is superimposed on the signal to simulate noise. The simulation results in Figure 4-19 show the output being approximately 4 times greater in amplitude than the input, with almost no trace of the noise seen at the input.

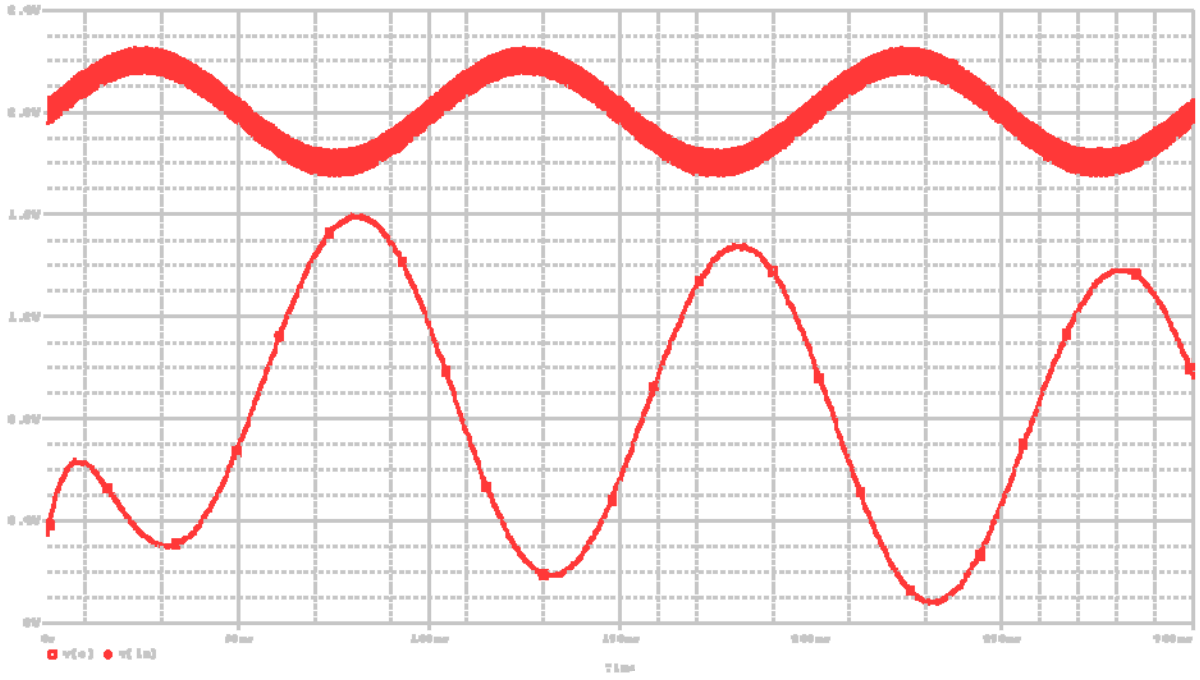

Figure 4-19: Raw Accelerometer Signal (top), Filtered and Amplified Signal (bottom)

4.3.3.3 Filtering the Accelerometer Output

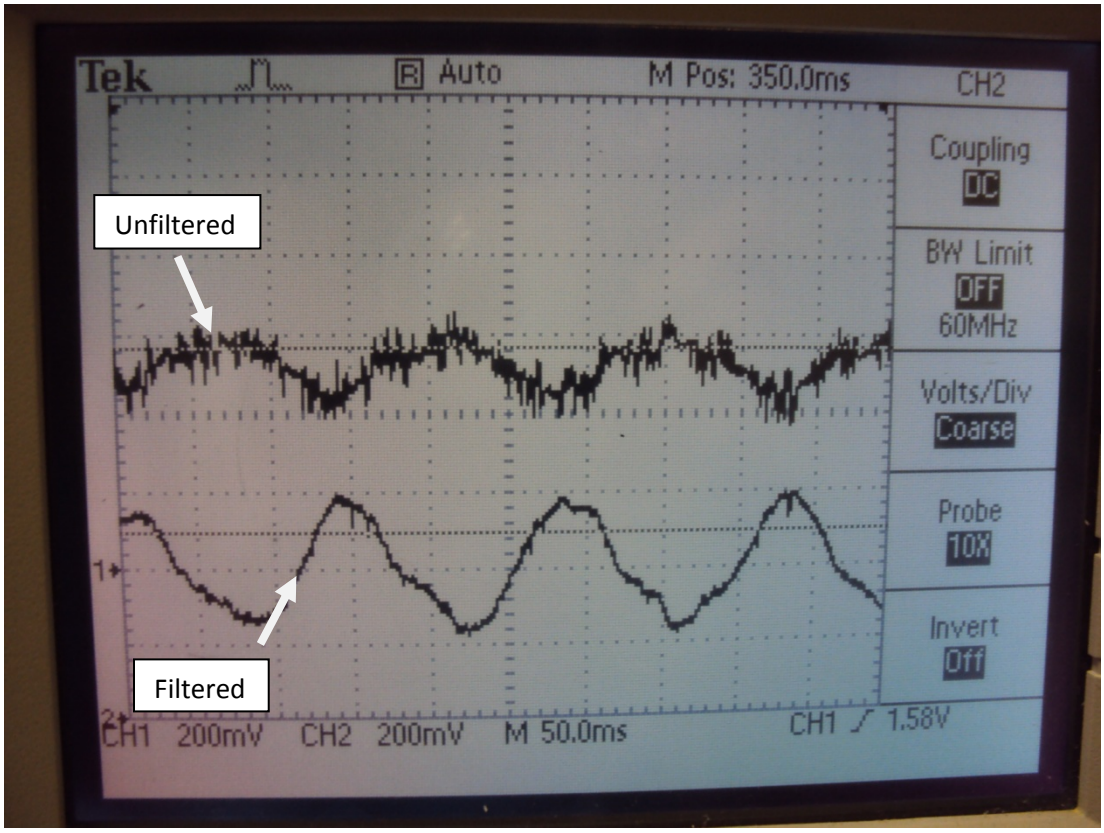

Figure 4-20: Comparison between filtered and unfiltered Accelerometer output

The oscillogram in Figure 4-20 shows the accelerometer output with and without filtering. The signal is amplified as expected, and the noise is significantly reduced.

4.4 Microprocessor

4.4.1 Introduction

The microcontroller used in the Control System is MSP430F449, manufactured by Texas Instruments. An advantage of using this microcontroller is that it only needs an analog-to-digital converter (ADC) with PWM capabilities as peripherals. All the controls can be tuned by modifying software.

The electronic controller required for the Dynamic Shock Absorber has to be a very fast computing machine that reads from sensors, calculates the velocity of the mass Mc, implements control equations and produces an output to control an H-Bridge module to drive Coil 2 and oppose the motion of Mc.

4.4.2 Overview of the MSP430 Code

4.4.2.1 The "main()" Function:

The Control system was designed with a single microprocessor chip. The goal was simple: calculate velocity, implement control equations, generate required PWM and set or reset the output of a port depending on the direction of Mc's motion. The Microprocessor was programmed in the C language. The code comprised of 7 functions:

```
void Time_Count();                  // generate interrupt
void Init_PWM();                    //setup PWM
void Setup_Sensors();               //setup ADC
void Compute_Motion_data();         // converts sensor voltages to velocity, acceleration etc.
void Controls();                    // implement controls equation to find the required force;
void Set_PWM();                     // Sets the timer B registers to output the desired PWM signal
```

In the main loop the two functions: Void_Init_PWM and void Setup_Sensors are called at first.

The Void_Init_PWM function configures the timerB registers.

void Setup_Sensors sets up the ADC12 registers to sample voltages from 2 inputs. The SHTx bit is used for extended sampling time. To account for any stray capacitance in the wires and on the PCB, the sampling time is set to a high value.

The main part of the program is the while loop within the main loop. In the while loop are the functions Compute_motion_data(), Controls(), and Set_PWM().

Every time the while loop is executed The "**ADC12CTL0 |= ADC12SC;"** instruction starts the ADC conversion and all the functions are called, and finally the Set_PWM() function configures the TimerB CCR1 register to acquire the desired PWM duty cycle. The function Time_Count

generates an interrupt which determines how fast the program loops. The loop time can be modified by changing the TACCR0 register value.

4.4.2.2 Using ADC12 to Measure Sensor Values:

Configuring the ADC12:

The MSP430 provides 16 ADC inputs that can be sampled and stored in the ADC memory registers. Once the conversion is done the ADC12 IFGx bits are set, an automatic interrupt is generated and within the ISR a code is written so that the corresponding inputs are stored in variables A0results and A1results. Once the data is moved from the ADC12memory registers, the IFG x bits are automatically cleared.

The MSP430 ADC supports 4 operation modes. The single channel, single conversion was selected. The reference voltage is set to +AVCC which is +3V. Conversion and IFG bits are enabled. The ISR copies the values form ADC12MEM registers to the variables A0results and clears the IFG bits.

4.4.2.3 Computing Motion Data:

The values obtained from ADCMEMx registers are stored in A0results and A1results. These values are used to obtain the input voltages from the sensor and then the actual motion data**.**

$$\text{A0 results} = 4096 * \text{sensor voltage/full scale voltage} \quad (4.4)$$

Hence the sensor input voltage in milliVolts is

$$(\text{A0results}*3.3\text{v}/4096) \quad (4.5)$$

This equation gave the raw input voltage. To calculate the actual motion data i.e. acceleration, any offset voltage is subtracted from the result and then divided by the rate from the spec sheets. The accelerometer had two control inputs for 4 different sensitivity settings. For this configuration, the accelerometer had its output set to .8v per 9.8m/s^{-2} of acceleration.

Hence the acceleration from the accelerometer was calculated by the equation:

$$\textbf{Acceleration} = (\text{vy})*1.5*(9.8/.8); \quad (4.6)$$

The velocity was found by integrating (adding) the current acceleration with its difference from the previous value. This method of adding gets rid of the offset voltage from the sensor.

4.4.2.4 Using TimerB to Generate PWM:

The TimerB of the MSP430 has 7 capture compare modes. The code used the TimerB in compare mode to produce PWM. This is done by writing 110 in the OUTMODx registers to select reset-set as the output. The internal clock frequency is set to 1048576 Hz by the TBSSEL_2 command. The CCR0 value sets the frequency for the PWM signal.

$$\text{CCR0} = \text{clock frequency} / \text{Required frequency} = 1048576\text{Hz}/2\text{kHz} = 5242 \quad (4.7)$$

When the timer counts up to CCR0 value the output is set and remains high until the timer reaches the CCR1 value. Hence the CCR1 value can be varied from 0 to 5240 to generate a PWM signal from 0 to 99.6% duty cycle.

The Set_PWM() function simply sets the TBCCRx registers to the required PWM values calculated by the controls() function.

As the CCR1 value is written to the timer while the timer is running, it is important that CPU clock and timer clock are synchronized. Hence the MCLK or SMCLK must be chosen as a clock source.

4.4.2.5 The Time_count() Function:

The Time_count() function is used to configure TimerA to run in up mode. Using the TimerA compare mode, each time the timer counts up to the TBCCR0 value, an interrupt is generated. The Interrupt service routine sets the value “go” to 1. Within the main function a while loop awaits the the variable “go” to be set. Each time this value is set the while loop executes all the required functions and resets the value of go to zero and awaits the next interrupt. The timer was configured to generate an interrupt every 1ms. This value determines how frequently the system acquires data and produces a corrective output.

$$\text{TACCR0 value} = \text{SMCLK frequency}/1000\text{Hz} \quad (4.8)$$

$$\text{TACCR0} = 1048576\text{Hz} /1000\text{Hz} = 1049 \quad (4.9)$$

4.5 H-Bridge Module

4.5.1 Introduction

The main purpose of this module is to provide Coil 2 with a current, after getting an input signal from the microcontroller, to keep the motion of the body stable over time. After denoting changes in the motion of the Suspension Mass, the accelerometer sends out a signal to the microcontroller, which then feeds a pulse width modulated signal (PWM) into the current driver. The current driver will then provide a PWM signal to the H-Bridge module, providing a controlled current to Coil 2, thus exerting an opposing force on the Chassis Mass to keep it stationery

4.5.2 H-Bridge Supply Voltage Requirements:

Considering the Chassis Mass at a fixed distance from the Suspension Mass, the force acting on the Chassis Mass is proportional to the current flowing through Coil 2. This leads to the conclusion that the attenuation of chassis movement is improved with increased coil current. Clearly, a smaller resistance in series with Coil 2 will allow for greater current, however, the ratio of resistance to inductance affects the time constant of this circuit. Reducing the resistance increases the time required for the current to reach its maximum value. This creates a potential problem, as in actuality, the control system controls the voltage across the coil, but the current is what provides the force. If the delay between attempted control and actual force is too great, it's possible that the control system would increase the oscillation of the Chassis Mass, rather than decrease it. The expression for the current through the inductor in this system, as a function of time, is:

$$I = I_{max}(1 - e^{\frac{-tR}{L}}) \tag{4.10}$$

To find the time at which the current is at 99% of its maximum value, the following expression applies:

$$T = -\frac{L}{R}\ln.99 \tag{4.11}$$

The inductance of Coil 2 is approximately 600uH. Sweeping the resistance from .1 to 8Ω provides the plot in Figure 4-21.

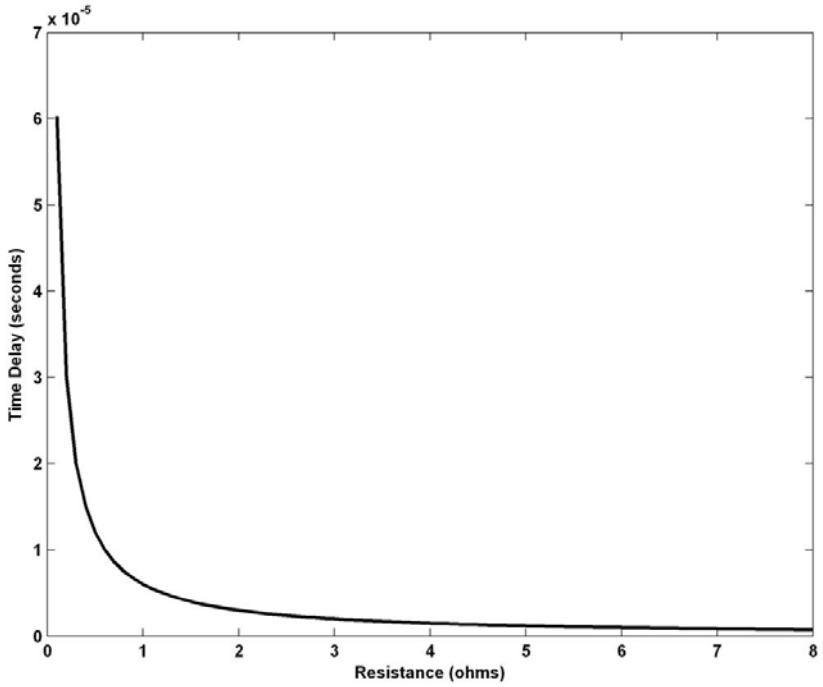

Figure 4-21: Time Delay vs Resistance

The plot shows that the time delay only starts to increase dramatically for resistances below 1Ω. High power resistor banks are readily available that can be configured to be as low as 3Ω, so this value will be used for the following design calculations.

4.5.3 Power Supply for H-Bridge Module

The power supply required to drive Coil 2, as described above, is a large one. Purchasing such a power supply, similar to the power supply required for the oscillator module, would be a large investment, as would be the components for making a respectable one. In this case, it is advantageous to study how low the quality of the power supply can be without affecting performance to a noticeable degree.

The most basic power supply configuration, consisting of a full wave rectifier and a filter capacitor, is the cheapest solution, but the capacitor can only be so large before cost becomes an issue, and a smaller capacitance will exaggerate the ripple of the supply voltage. In PSPICE, a rippling voltage can be superimposed on the supply voltage, and the resulting attenuations can be observed.

The magnitude of the voltage ripple is a function of filter capacitance, and an approximate formula is derived as follows:

$$I = C\frac{dV}{dt} = C\frac{\Delta V}{\Delta t} \tag{4.12}$$

$$\Delta V_{max} = \frac{I_{max} * \Delta t}{2C} = \frac{\Delta I_{max}}{2 * C * f} = \frac{30}{120 * C} \tag{4.13}$$

This formula can be implemented in PSPICE, and the value C can be swept, providing different attenuations of velocity. This particular simulation mimics the effect of pushing down on one of the plates, and letting it go abruptly to create a transient oscillation. The results are shown in Figure 4-22.

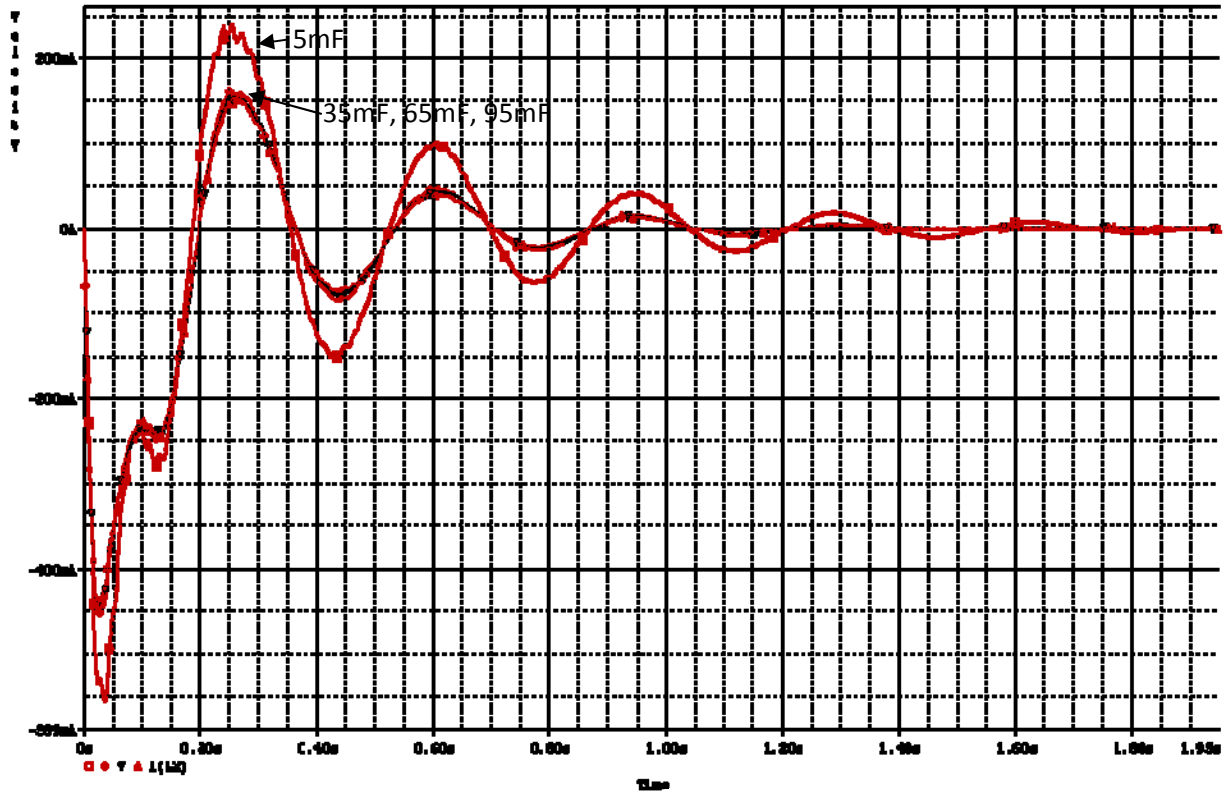

Figure 4-22: Transient Plot of Velocity of Mc, Dependant on Filter Capacitance

Here, four values of filter capacitance are simulated: 5mF, 35mF, 65mF, and 95mF. The graphs show that the ripple resulting from the small 5mF capacitor has a noticeable degrading effect on the attenuation of motion, but there is a diminishing return on increasing the capacitance. For this application, a 75mF Capacitor, rated at 65V, was found to be an excellent compromise between quality and cost.

4.6 H-Bridge Module

4.6.1.1 Introduction

The schematic in Figure 4-23 details the various components of the H-Bridge module.

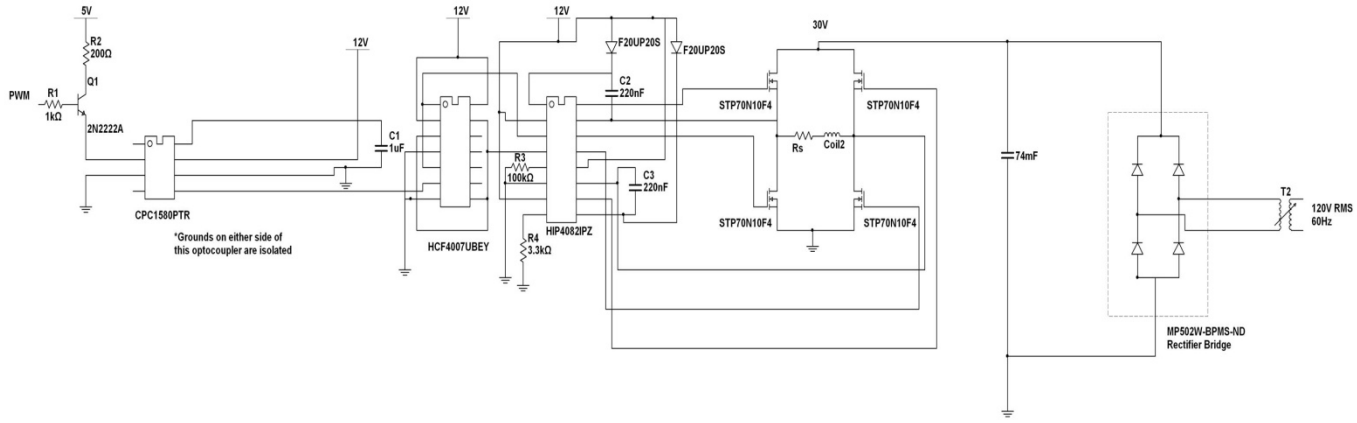

Figure 4-23: H-Bridge Module Schematic

4.6.1.2 H-Bridge Configuration of MOSFETs

The H-Bridge circuit, used to apply voltage across the coil in either direction, is comprised of four power MOSFETs. The schematic for the H-Bridge is shown in Figure 4-24. Using square waves of varying duty cycle, the average current flowing though the coil can be changed. Current paths in blue and red lines show that when T1 and T3 turn on the coil has current flowing through it in one direction. When T4 and T2 are turned on the flow of current is in the opposite direction. By switching between the pairs of transistors the direction of current in the coil is switching the direction in which force is exerted on Mc.

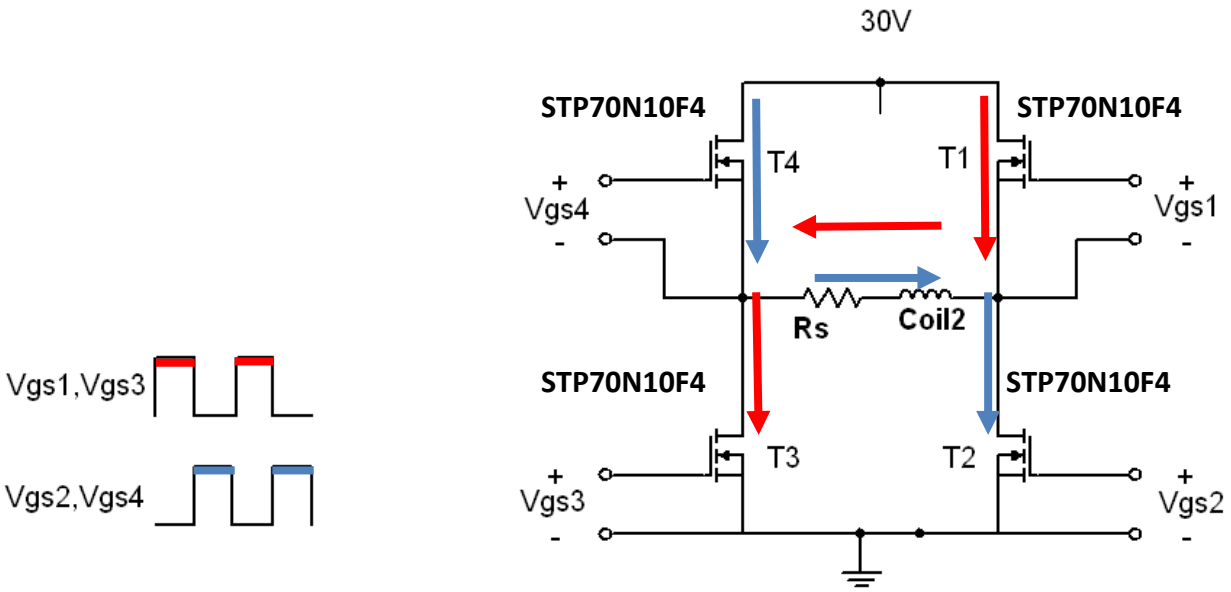

Figure 4-24: N channel MOSFET H-Bridge

The high power MOSFET, STP70N10F4, can withstand a maximum current of 60A and a maximum voltage of 100V. It was chosen for its extremely fast switching capability, along with its high power rating. Moreover, it has exceptional dv/dt capability and extremely low on-resistance $R_{DS(on)}$, providing outstanding switching performance.

The bridge is supplied with a voltage of 30V, and Coil 2 is connected in series to a resistance of 5Ω. This provides the bridge with up to 6A of current.

4.6.1.3 Opto-Coupler

The first stage in the H-Bridge module is a Clare CPC1580PTR opto-coupler. In the case of excessive voltage or current surges in the H-Bridge, the opto-coupler protects the expensive MSP430 development board. This chip can isolate the input from output side voltages up to 3750Vrms. A 2n2222 transistor allows the MSP output to drive the opto-coupler internal led. The input and output side of the opto-coupler operate off of de-coupled power supplies.

4.6.1.4 Inverter

Right after the opto-isolator is an HCF4007UBEY inverter IC. The bridge driver requires a PWM signal, as well as the inverted form of that signal, and this chip provides the inversion. Initially, it was intended for the opto-coupler output to go directly to one of the bridge driver's inputs, while the signal from the inverter would go to the other input. However, it was discovered experimentally that the output from the opto-coupler was slightly ambiguous, in that a logic low was approximately 2v, which is potentially high enough to be considered logic high by the bridge driver. To eliminate this ambiguity, the two inverters on the inverter chip are used

one after the other, providing both the inverted signal, and a cleaner version of the original signal.

4.6.1.5 Current Driver

4.6.1.5.1 Introduction

The power MOSFET used in the H-Bridge requires at least 4V greater than the source terminal to turn on. Power MOSFETs have a large stray capacitance, C_{iss}, between the gate and other terminals [5]. When a pulse arrives at the gate of a power MOSFET, it must charge up this capacitance before the gate voltage can reach the 4V required for the MOSFET to turn on. For this reason, the circuit driving the gate of the MOSFETs should be able to supply a reasonable amount of current to charge up this stray capacitance as quickly as possible. Dedicated MOSFET current drivers are the best solution for this, making sure that the stray capacitance is charged up before the MOSFET can turn on.

For the purpose of controlling the current flowing into the H-Bridge module, a HIP4082IPZ current driver made by Intersil was used. The HIP4082 is an 80V, 1.25A peak current, N-Channel MOSFET Driver IC which drives all of the four MOSFETs in a complete H-Bridge configuration. It has the capability of driving every single MOSFET (switch) combination, other than the shoot-through condition. The driver embodies various functionalities that make it an ideal candidate for driving the MOSFETs.

4.6.1.5.2 Dead time:

Dead time is the time spent when one of the upper MOSFET gate is turned off, and one of the lower MOSFETs is turned on, and vice-versa. HIP4082 makes it possible to use just one resistor to adjust the dead-time, and helps in choosing the most appropriate dead-time. This prevents the conduction of current by one upper/lower MOSFET and one lower/upper MOSFET simultaneously, when they are switching to change on/off properties

The dead-time setting ensures that a high side and low side transistor on the same side of the bridge are never on at the same time. Such a shoot-through condition is hazardous for the bridge, bridge controller, and power supply.

4.6.1.5.3 Bootstrap Supply Design:

The bootstrap supply technique, using two diodes and two capacitors for a full bridge driver, provides bias supplies for the floating logic parts of the driver. The bootstrap capacitor gets charged by the low voltage supply (V_{CC}) [5]. Between the bootstrap capacitor and V_{CC}, a quick recovery diode is connected [5]. The other side of the capacitor is connected to COM or V_{SS} through a lower MOSFET, as long as the lower MOSFET or its body diode is conductive [6]. As an upper MOSFET turns off, the remaining load current causes the body diode to conduct, and so it is generally advisable to produce a short time period between the PWM cycle as it causes the bootstrap capacitor to charge [6]. This time period causes the bootstrap capacitor to get rid of all the charge that has been present in the circuitry from the previous charging session.

The upper logic section of the MOSFET requires a bias supply when it is gated on or off, and it is provided by the bootstrap capacitor. Moreover, the capacitor provides the charge required to pump the equivalent MOSFET input capacitor to the V_{CC} level. This charge is described by equations 4.14 and 4.15 [5].

$$\text{Charge at Gate} = \text{Gate Capacitance} \times \text{Gate Voltage} \tag{4.14}$$

$$Q_{GATE} = C_{GATE} \times V_{CC} \tag{4.15}$$

The upper bias supply operating current could be varied by changing the PWM duty-cycle. To measure current to calculate the bootstrap capacitance, the exact-level of current should be calculated at the desired duty-cycle.

Choosing the right bootstrap capacitor is extremely important as it plays a major role in making the H-Bridge work properly. As the HIP4082 IC first powers up, there is an extremely short time window during which the capacitor has to charge up. Approximately a 500ns pulse is applied to the lower two MOSFETs that charges the bootstrap capacitors for the first time. At the same time, the upper MOSFETs are kept low. These start-up pulses are applied regardless of any input logic signals, except for DIS [5]. If the capacitor value is too high, then it might not get completely charged up during that initial time window, which is approximately 500ns [6].

Using data from the datasheet of the STP70N10F4 N-Channel MOSFETs and the diode, F20UP20S, the required bootstrap capacitance was calculated.

$$\text{Maximum Gate Charge: } 65\text{nC} \tag{4.16}$$

$$\text{VDS: } 12\text{V} \tag{4.17}$$

$$\text{Recovery Time of the diode: } 35\text{ns} \tag{4.18}$$

$$\text{Recovery time duration} = 17.5\text{nC} \quad \text{(assuming half the peak reverse current magnitude which is at 1A peak)} \tag{4.19}$$

$$\text{CBS} = (175\text{nC} - 17.5\text{nC})/(12-11) = 0.1575\text{uF, will result in less than 1 V drop} \tag{4.20}$$

Therefore a bootstrap capacitance of 0.1575uF will result in less than 1V drop in the voltage across the bootstrap capacitor when either one of the upper MOSFETs are turned on. Any capacitance above this value should tend to the situation, but if the capacitor value is too high it will not get charged up properly during the first power up situation.

4.6.1.6 H-Bridge Physical Construction

The first iteration of H-Bridge construction, shown in Figure 4-25, was centered around giving the MOSFETs enough space to be equipped with heat sinks. Since copper clad perf-board was used for the construction, 12 gauge wire was run along the top of the board to make the connections between the MOSFET drains and sources, power supply, and load. While testing this board and the bridge controller, the transistors would occasionally burn out, and replacing them within this board was extremely problematic.

The second iteration of the H-Bridge was almost identical to the first, but had the addition of screw terminals for the MOSFETS, so that the MOSFETS could be replaced easily. This reduced the time to fix the bridge drastically.

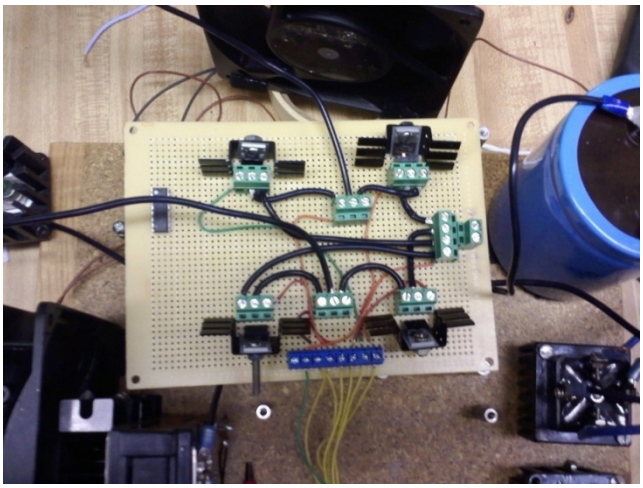

Figure 4-25: H-Bridge layout

After operating the second generation bridge for some time, it was found that the MOSFETS could operate without heat sinks, as long as there was good air flow, and the layout of the system was already designed so that the H-Bridge would be in front of a fan. This led to the third iteration of H-Bridge construction, shown in Figure 4-26. In this version, the transistors were placed as close together as possible on a copper clad perf-board. Essentially, the MOSFET terminals are soldered together directly, with the board simply acting to hold everything in place. This setup decreased the H-Bridge's stray inductance significantly.

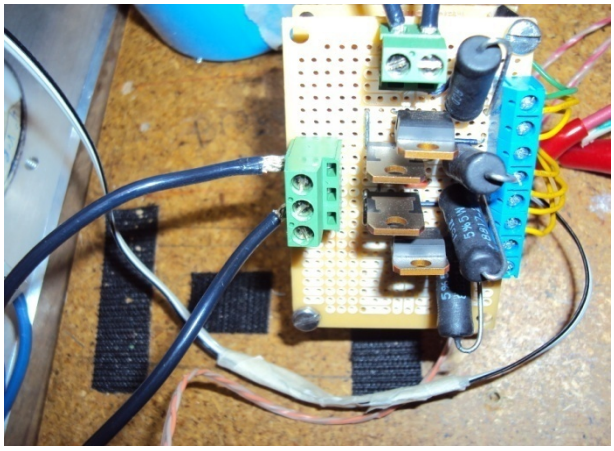

Figure 4-26: Compact H-Bridge layout

4.6.2 Force/Current Relationship

To simulate how the current through the second coil affects the actual motion of the masses, it is necessary to know how much force is produced on the Chassis Mass for a given amount of current. This data was measured by fixing the bottom mass motionless, passing known levels of current in increments through the coil, and measuring the corresponding displacement of the top mass. Knowing the total spring constant pertaining to the springs between the top and bottom masses, force can be calculated from the distance data. A best fit curve was applied in Matlab, yielding the following equation:

$$F = (.005)I^3 + .8 * I \tag{4.21}$$

This curve is plotted against the measured data in Figure 4-27:

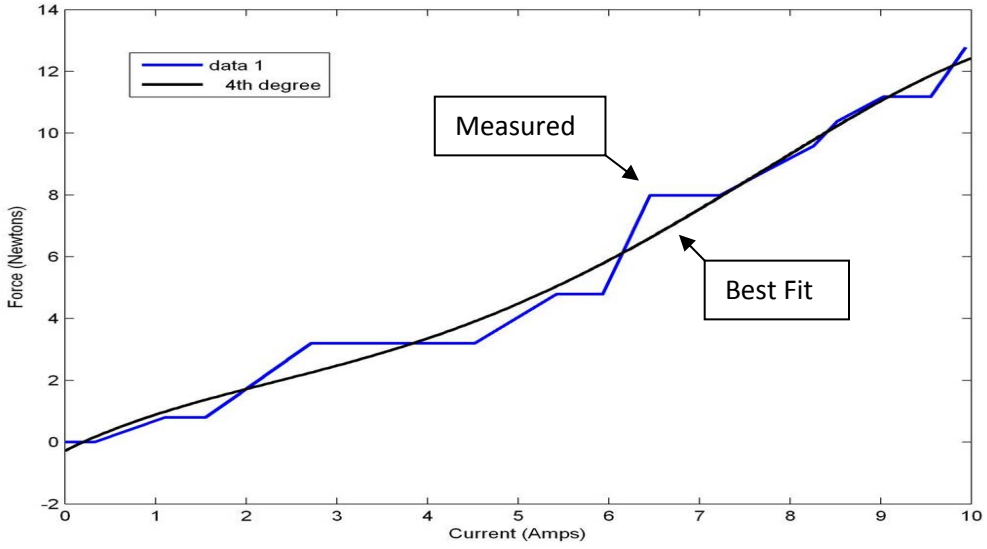

Figure 4-27: Force VS Current Plot (Coil 2)

5 Performance of the Dynamic Shock Absorber

5.1 Measured Data

To test the Dynamic Shock absorber, the Oscillator circuit drove the bottom coil (Coil1) to simulate road conditions, and the top coil was driven by the H-Bridge receiving a control signal from the microprocessor. The PWM signal at the output of the H-Bridge had a peak voltage of 30v and the coil had an external resistor of 5ohms. The Maximum control current though Coil 2 is calculated in equation 5.1:

$$\text{Current} = 30\text{v}/5\text{ohms} = 6\text{A} \qquad (5.1)$$

For an initial test of the control system, the accelerometer was mounted firmly on the Chassis Mass and the system was forced to oscillate at resonance (7.8Hz). Figure 5-2 and Figure 5-3 show oscillograms for the acceleration of the Chassis Mass without controls and with controls respectively. With the controls activated, the oscillation of Mc reduces from 568mv to 256 mV. This is a decrease in acceleration of 55%.

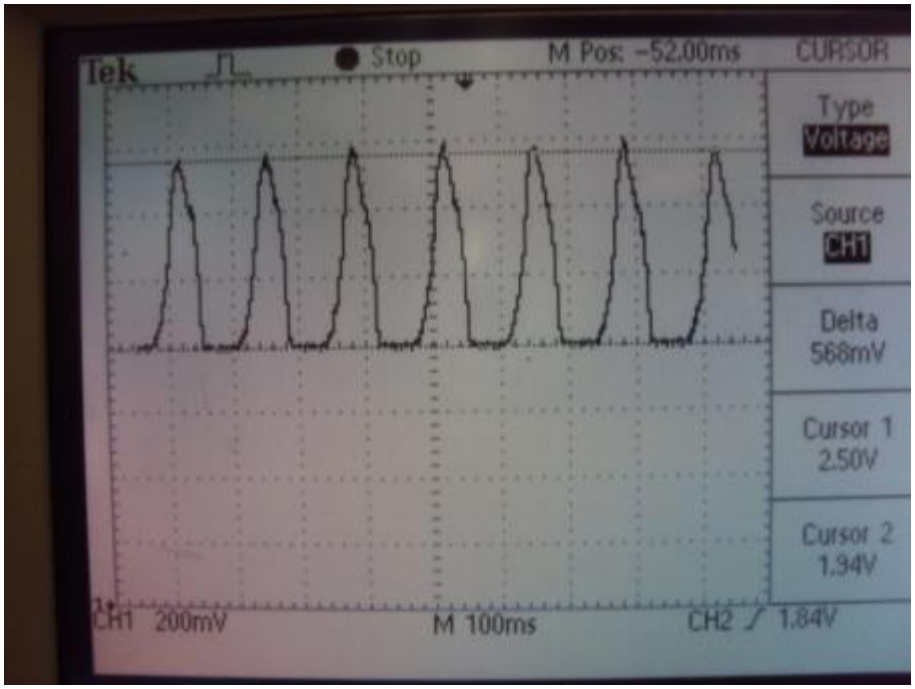

Figure 5-1: Accelerometer Output of Mc oscillations without Controls

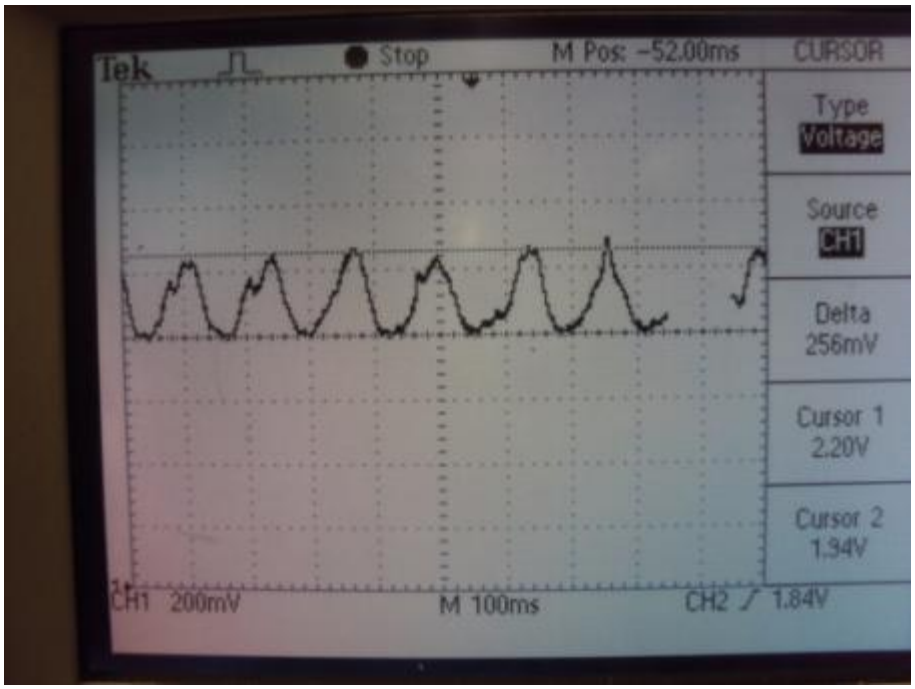

Figure 5-2: Accelerometer Output of Mc Oscillations with Controls

The same experiment was carried out with the system oscillating at 3.52Hz. Figure 5-3 shows oscillation without controls and Figure 5-4 shows oscillation with controls activated. These oscillograms show that the amplitude is reduced from 336mV to 240mV, a reduction of 29%.

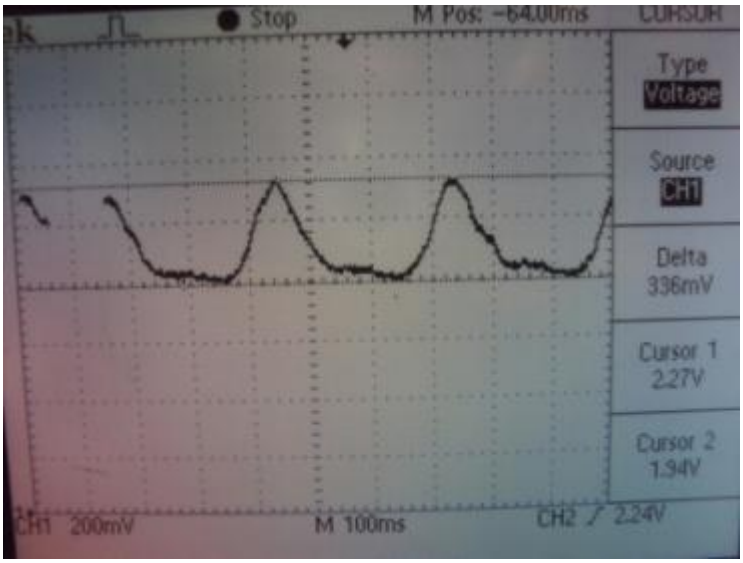

Figure 5-3: Accelerometer Output of Mc Oscillations (3.52Hz) Without Controls

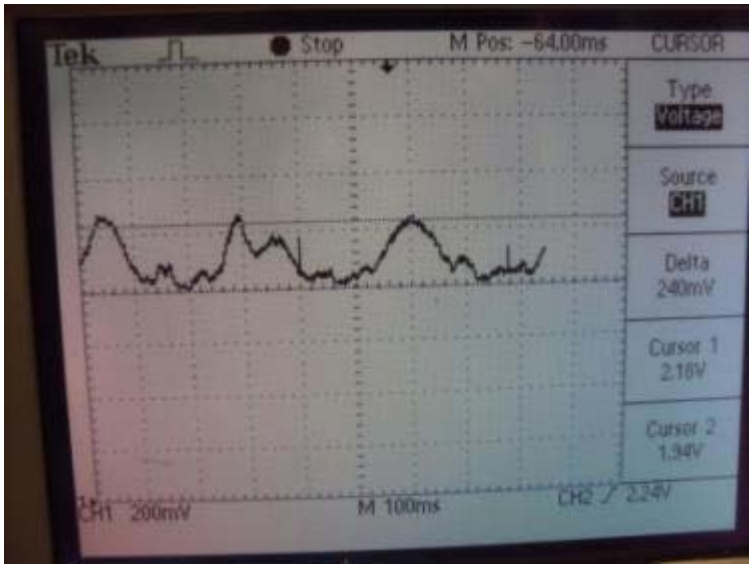

Figure 5-4: Accelerometer Output of Mc Oscillations (3.52Hz) With Controls

To further study the performance of the Dynamic Shock absorber, the acceleration of the Chassis Mass was measured with and without controls for frequencies ranging from 0Hz to 11Hz and the results were tabulated. The graphs shown in Figure 5-5 show the accelerometer output voltage for the controlled and uncontrolled conditions. The red graph shows the magnitude of acceleration with controls and the blue graph shows without controls.

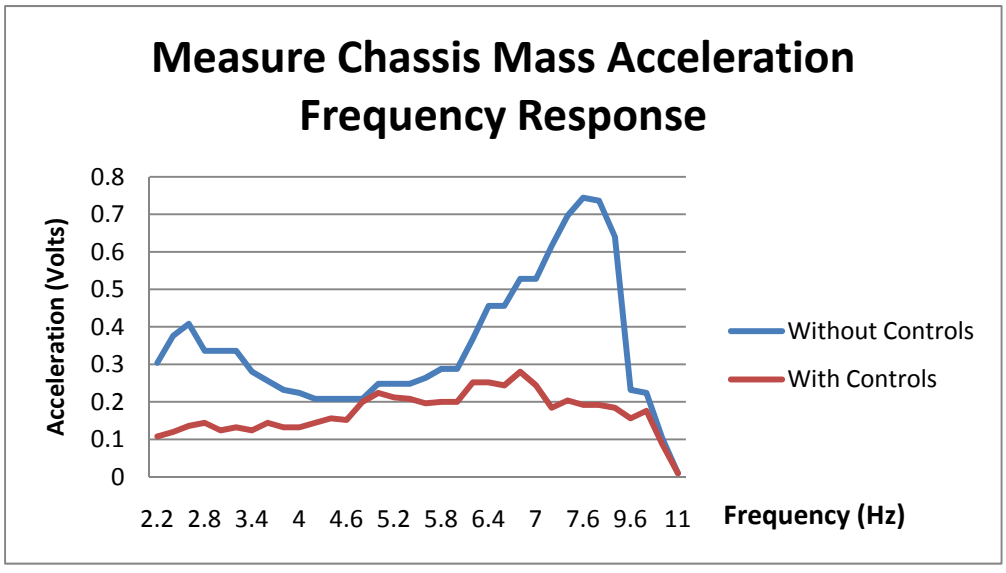

Figure 5-5: Measured Chassis Mass Acceleration Frequency Response

Previously, measurements of the system's oscillations had been done by monitoring the voltage across Coil 2, as explained in section 2.2, effectively providing the difference in velocity

between the two masses. These newer measurements utilized an accelerometer on the Chassis mass alone. To convert from acceleration to velocity, the data points for the acceleration data set were all divided by the frequency of oscillation in radians. The resulting Chassis Mass velocity frequency response is shown in Figure 5.6. The results turned out to be similar to the original findings, in that there are two resonant frequencies. The second resonant frequency, however, is not seen in the Chassis mass's velocity simulation (refer to Figure 2-2). This result necessitates an adjustment in the simulation model.

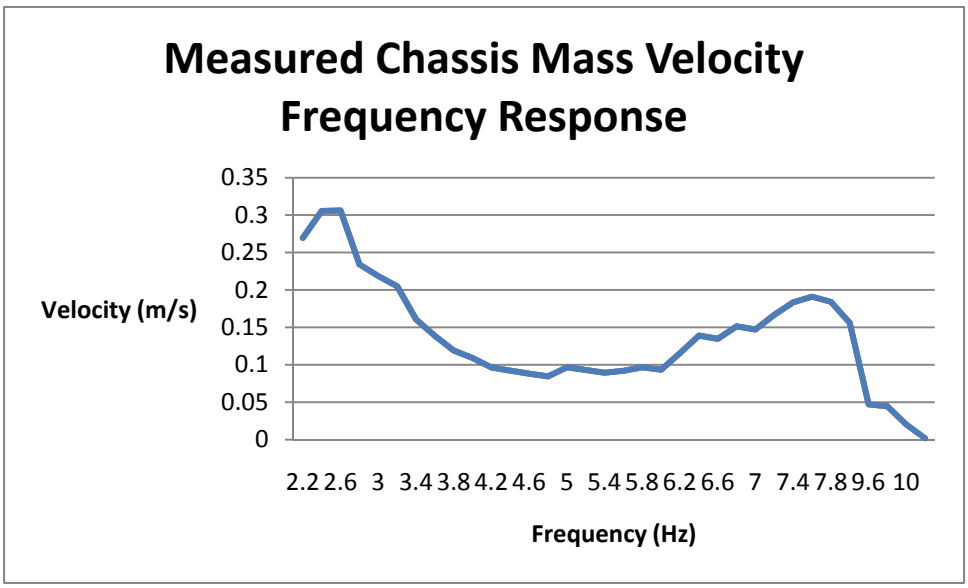

Figure 5-6: Chassis Mass Velocity Measured.

5.2 New Simulation Model

A suspected cause for the discrepancy between the measured and simulated data is the effect of fluid damping on the masses. Damping due to friction within the springs is proportional to the difference in velocity between the two plates, whereas damping due to air resistance is proportional to the velocity of the top plate alone. This necessitates a revision in the simulation model.

Generally, the force on an object due to air resistance is proportional to the square of the velocity of that object. For objects moving slow enough so that air turbulence is minimal, the force due to air resistance is proportional to velocity [8]. The proportional condition was tested first. This new configuration was achieved by placing a resistance in series with the inductor representing the Chassis Mass. It was found that this model significantly improves the matching between measured and simulated results. The PSPICE code for this new model can be found in Appendix E. The frequency response curve for the Chassis Mass velocity of this new model is shown in Figure 5.7.

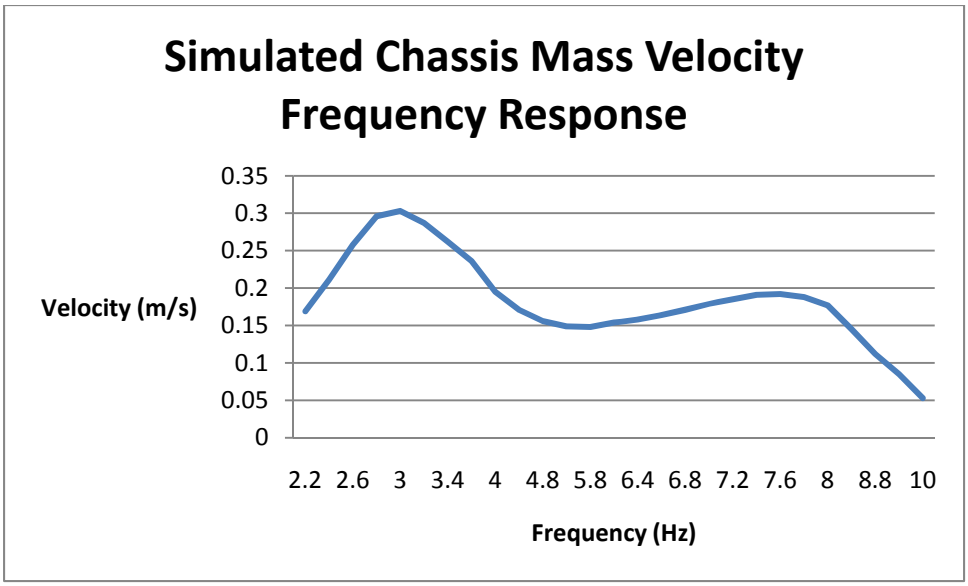

Figure 5-7: Chassis Mass Velocity Simulation

Because this model appears to be quite accurate, the more complex and convoluted model (force proportional to velocity squared) will be forgone. The new electrical equivalent model is shown in Figure 5-8.

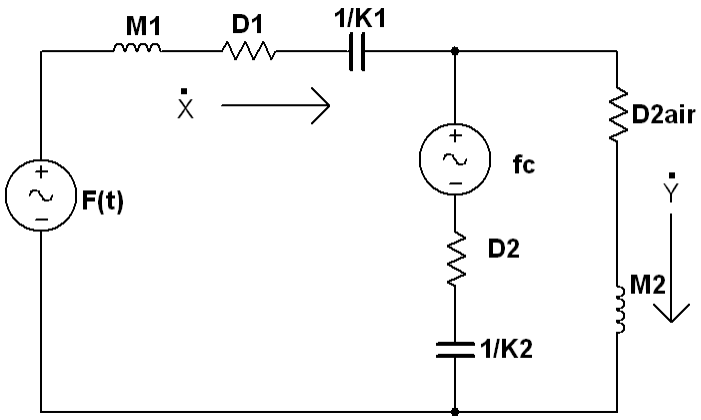

Figure 5-8: Electrical Equivalent for Adjusted Model

In section 1.3.3, the damping constants were derived from oscillograms of the chassis and Suspension Masses moving separately. The model adjustment will therefore not affect the damping coefficient of the Suspension Mass, however the measured value of approximately 18nm/s for the chassis damping coefficient should be split between the damping due to air and the damping due to resistance within the springs. Experimentally, the values best suiting the simulation were:

$$D_2 = 5\frac{n * m}{s} \tag{5.2}$$

$$D_{2air} = 13\frac{n * m}{s} \tag{5.3}$$

5.3 Discussion

Plots of the simulated and measured frequency response curves of the velocity of the Chassis Mass for both the controlled and uncontrolled condition are shown in Figure 5-9 and Figure 5-10, respectively.

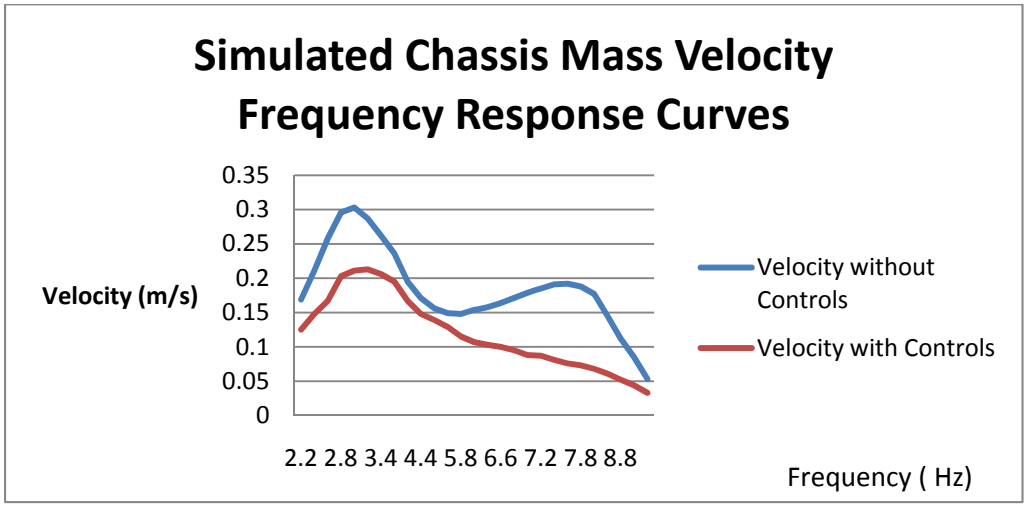

Figure 5-9: Simulated Chassis Mass Velocity Frequency Response with and without Controls

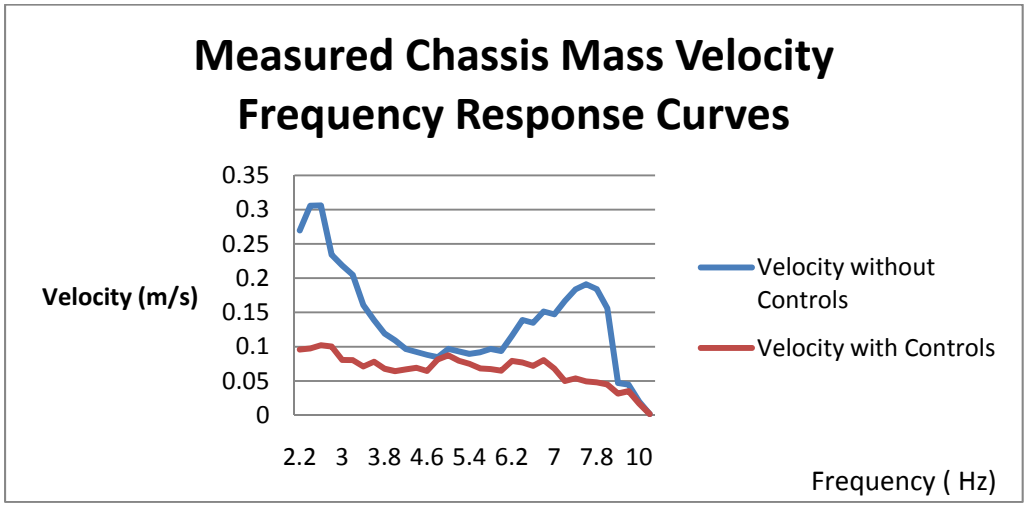

Figure 5-10: Measured Chassis Mass Velocity Frequency Response with and without Controls

For a comparison of the measured and simulated data, plots of attenuation as a function of frequency can be made. The simulated and measured attenuations are shown in Figure 5-11 and Figure 5-12, respectively.

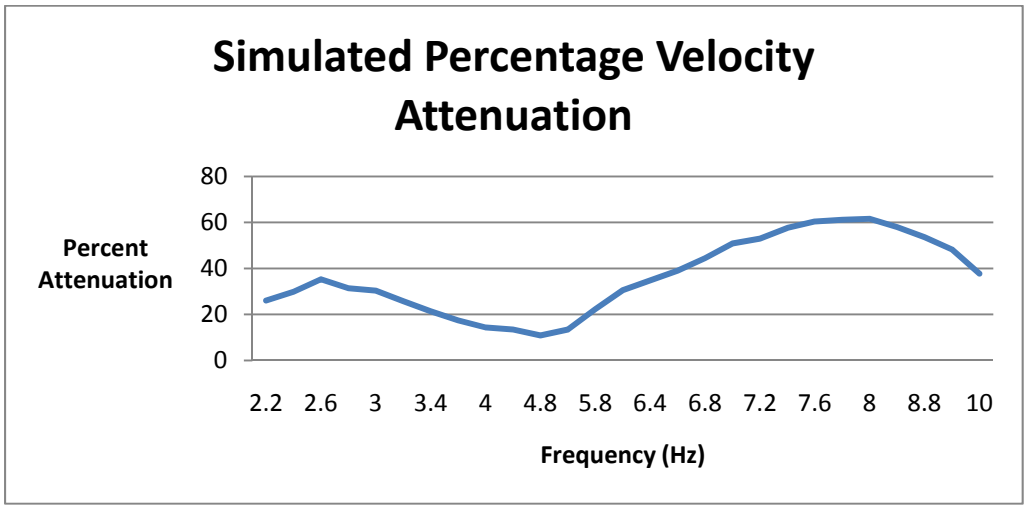

Figure 5-11: Simulated Velocity Attenuation

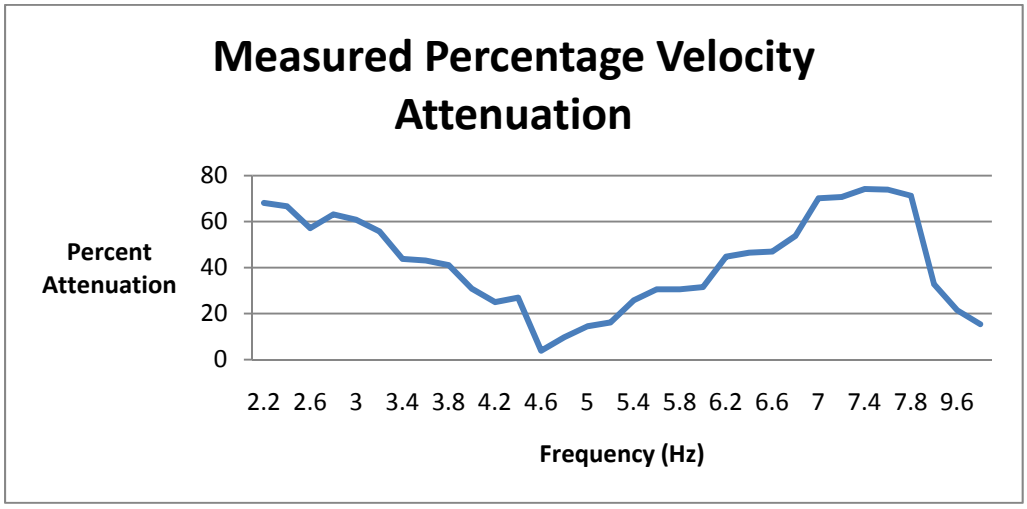

Figure 5-12: Measured Velocity Attenuation

The model introduced in section 5.2 is effective in providing approximate predictions of how the physical model reacts to the control system. However, these simulations could be improved upon with a more complex model for air damping.

On its own, the measured data demonstrates the success of the control system. At the resonant frequencies, the velocity of the Chassis Mass is reduced by over 70%.

6 Conclusion

The objective of the project was to design a suspension system which was more efficient than the commercial shock absorbers in production today. Ideal goal was to use electromagnets to design an active suspension system that would keep the body of a vehicle constant regardless of different road conditions.

Electromagnetic suspension systems have been researched, designed and developed from the early 90s, and ideally they are chosen over any other sort of suspension systems, like hydraulic and pneumatic, due to their fast response and regeneration of power through motors, as the energy dissipated during shock impulses are usually absorbed by the electromagnets [7]. But the disadvantages of using electromagnetically active suspension systems are high cost, design intricacies and frequent repairs and maintenance, requiring highly trained personnel.

The initial approach involved extracting the electrical equivalent circuit from the mechanical circuit, which eventually lead to the design and implementation of the model. Breaking the system in block diagrams helped in analyzing the system in parts. The main modules of the project consisted of a road condition module and a control module.

The road condition module consisted of designing an oscillator circuit to mimic various road conditions, and produce oscillations of variable frequency to Coil 1 in order to make the Suspension Mass oscillate.

The control module was the bigger section of the two modules, as it consisted of various segments to produce a controlled current to Coil 2, which in turn would produce a force to keep the Chassis Mass constant. The control module consisted of a sensor, a microprocessor, and an H-Bridge circuit.

An accelerometer was used as a sensor that detected changes in acceleration of Mc. The accelerometer output signal was amplified and filtered, for proper signal conditioning, and then sent to the MSP430 microcontroller. The amplification and filtration of the output signal changed the results significantly, once oscillations from both the filtered and unfiltered signals were compared.

The MSP430, depending on the signal from the accelerometer, would send out a PWM signal to the H-Bridge, making Mc oscillate. Control equations were implemented into the MSP430 code to generate the desired current required to drive Coil 2. The control equations were initially simulated to make sure that they complied with the actual design ideas.

The H-Bridge, however, required a lot of troubleshooting and debugging for desired end results. Initial physical design constraints led to poor performance, and at times ended in circuit failure. Due to cost effectiveness, the H-Bridge circuit was built from scratch using Power MOSFETs.

This personal design constrained the usage of 50V, due to safety, and was decided to operate under 30V. This was sufficient enough to produce a significant amount of current for proper operation of the shock absorber. Even though the data collected sufficed this device, a current of larger magnitude would yield better results, as observed from the force current plot.

The overall results of the suspension system enabled the oscillations experienced by Mc to be attenuated by approximately 70%. Even though this did not resemble the desired output of zero motion experienced by the mass, it was enough to minimize disruptive impulses, experienced at resonance, significantly.

Reference

[1]. Shock Absorbers. *Hydraulics and Pneumatics.* [Online] 2005. [Cited: February 27, 2010.] http://www.hydraulicspneumatics.com/200/TechZone/FluidPowerAcces/Article/True/6442/TechZone-FluidPowerAcces.

[2]. **Ulaby, Fawwaz Tayssir.** *Fundamentals of Applied Electromagnetics.* s.l. : Pearson Prentice Hall, 2007. 0132413264.

[3]. **Acroname Robotics.** Demystifying the Sharp IR Rangers. *Acroname Robotics Website.* [Online] January 10, 2008. [Cited: November 29, 2009.] http://www.acroname.com/robotics/info/articles/sharp/sharp.pdf.

[4]. **George E. Danz.** A DC-AC Isolated Battery Inverter Using the HIP4082. *Intersil.* [Online] February 2003. [Cited: January 30, 2010.] http://www.intersil.com/data/an/an9611.pdf. AN9611.

[5]. **Michael O'Loughlin.** A better bootstrap/bias supply circuit. *Texas Instruments Incorporated.* [Online] October 2005. http://focus.ti.com/lit/an/slyt077/slyt077.pdf. SLYT077.

[6]. **Kugelstadt, Thomas.** Chapter 16: Active Filter Design Techniques. *Texas Instruments Incorporated.* [Online] 2008. http://focus.ti.com/lit/ml/sloa088/sloa088.pdf. SLOA088.

[7]. **Milton J. Bermejo Calle, Besnik Pashaj, Patrick Sebuwufu.** WPI Electronic Projects Collections. *Dynamic Shock Absorber.* [Online] March 17, 2009. http://www.wpi.edu/Pubs/E-project/Available/E-project-032709-133346/unrestricted/Dynamic_Shock_Absorber.pdf. E-project-032709-133346.

[8]. **Air Friction**. *Georgia State University.* [Online] Georgia State University, June 16, 2006. http://hyperphysics.phy-astr.gsu.edu/Hbase/airfri.html.

7 Appendix A: Schematics

Full Schematic of the Dynamic Shock Absorber

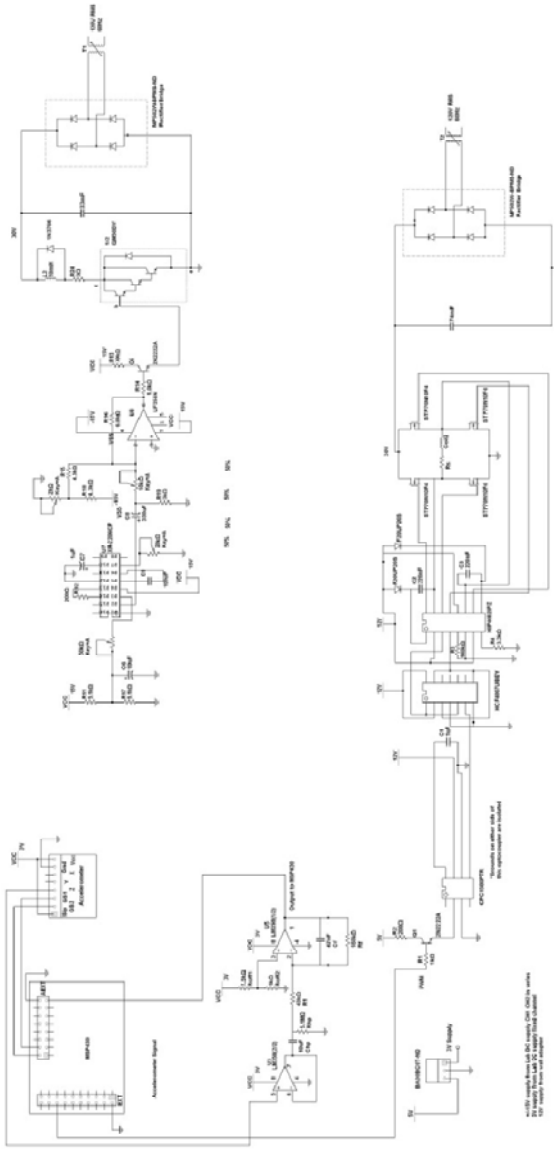

Figure 7-1: Full Schematic

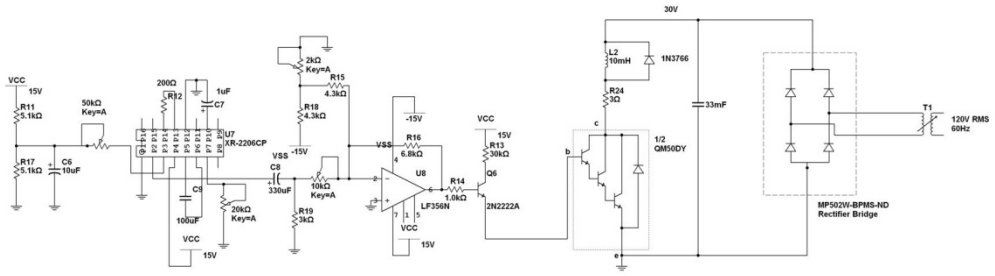

Figure 7-2: Oscillator Circuit

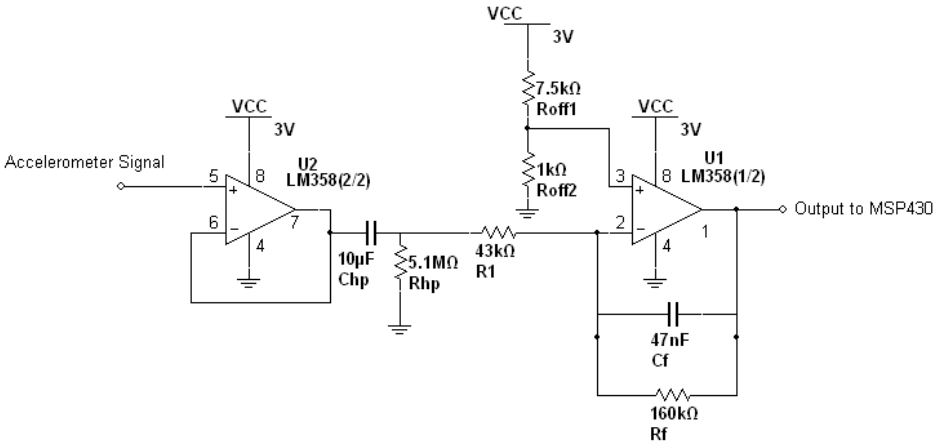

Figure 7-3: Accelerometer Signal Amplifier and Filter

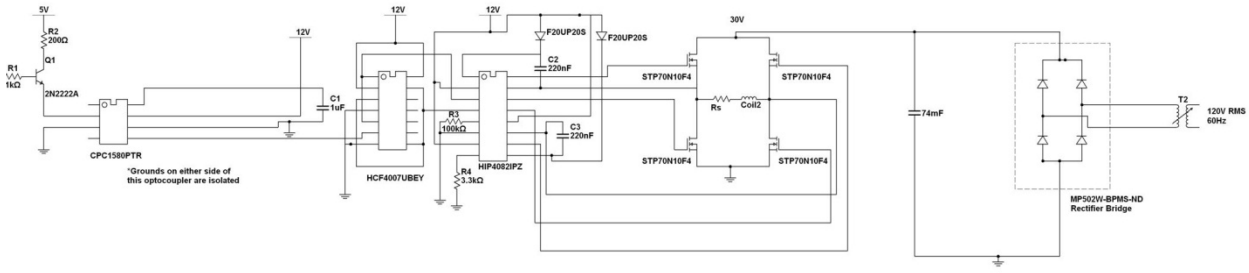

Figure 7-4: H-Bridge Module Schematic

8 Appendix B: PSPICE Code

```
.SUBCKT PWMgen in inref out outref
Vtri tri inref PULSE(-1.1 1.1 .01m 1.24m 1.24m .01m 2.5m)
*1.1 keeps duty cycle between 10 and 90 %
*Rin in inref 1G
Eout out outref TABLE {(V(dsh,inref)-V(tri,inref))}=(-.1u -1) (.1u 1)

Sit in sr tri 0 Switchit
Rsh sr dsh 1m
Csh dsh inref 1u IC=0
.MODEL Switchit VSWITCH(Ron=1m Roff=1G Von=1 Voff=.99)
.ENDS

.PARAM Kp = 1
.STEP PARAM kp list  9  2000
.PARAM Vsup 100

V 1 0 Sin(0 5 2.9)
L1 1 2 1.641 IC=0
R1 2 3 11.122
C1 3 4 725u IC=0
C2 4 5 626u IC=0
L2 4 7 1.609 IC=0
Vvelocity 7 0 0
Vdcoil2 11 0 0
R2 5 6 17.79
L3 8 9 600u IC=0
Rcoil2 9 10 300m
Rcoil2series 10 11 3
Eforce 6 0 VALUE={.005*I(Vdcoil2)*I(Vdcoil2)*I(Vdcoil2)+.8*I(Vdcoil2)}

Eduty duty 0 TABLE {kp*I(L2)} = (-1,-1) (1,1)
Xpwm duty 0 B 0 PWMgen

Ebridgepwm 8 0 VALUE={-Vsup*v(B)}

Emotionfilter mf 0 VALUE={1*I(L2)}
Rmf mf mfout 1k
Cmf mfout 0 16u

.Probe
.TRAN 1 1 0 1m UIC
.END
```

9 Appendix C: MSP430 Code

```
// The code gets inputs from ADC, calculates acceleration and integrates to get velocity
//and produces a PWM output that drives the H-Bridge module.




#include  <msp430x44x.h>
#include <math.h>

void Time_Count();              // generate interrupt
void Init_PWM();              //setup PWM
void Setup_Sensors();       //setup ADC
void Compute_Motion_data();  //converts sensor voltages to velocity,  acceleration  etc.
void controls();               // Calculates the required PWM output to drive the H-Bridge
void Set_PWM();                          // Sets the timer B registers to output the desired PWM signal

int go;
float velocity=0;
float y=0;
float A0results;
float DESTR1;                  // Destination register for PWM duty cycle
float vy=0;                     //  vy: voltage at A0
float vyprev=0;




void main(void)

{

 WDTCTL = WDTPW +WDTHOLD; // Stop WDT
_BIS_SR(GIE);   // Enable interrupt



Init_PWM();                          //Configure timer B for PWM output
```

```
Setup_Sensors();    //Setup registers for A/D conversion
Time_Count();       //Setup TimerA to generate interrupt. Sets the frequency of the while loop

while(1)

{

while(go==0) // wait for timerA interrupt
{;}
go=0;               // reset go

ADC12CTL0 |= ADC12SC;           // start ADC conversion
while (ADC12BUSY&ADC12CTL1==1) // while ADC is busy do nothin
{_NOP();}

Compute_Motion_data();  // computes acceleration and velocity
Controls()          // calculates a value for TimerB register to set duty cycle
Set_PWM();                  // Sts the value to timer B register

}

}




void Init_PWM()

{

  P2DIR |= 0x0c;                    // P2.2 - P2.3 output
  P2SEL |= 0x0c;                   // P2.3 - P2.3 TBx options
  TBCCR0 = 5200;                    // PWM Period

  TBCCTL1 = OUTMOD_7;                                   // CCR1 reset/set
  TBCCR1 = DESTR1;                 // CCR1 PWM duty cycle

  TBCTL = TBSSEL_2 + MC_1;          // Selects SMCLK clock source ,
                                    sets timerB in up mode
}



void Setup_Sensors()

{
```

```
  P6SEL |= 0x01                     // Enable ADC channel A0
  ADC12CTL0 = ADC12ON+SHT0_2;         // Turn on ADC12, set sampling time
  ADC12CTL1 = SHP;                    // Use sampling timer, set mode
  ADC12IE = 0x01;                   // Enable ADC12IFG.0
  ADC12CTL0 |= ENC;                   // Enable conversions
  _EINT();                          // Enable interrupts

}



#pragma vector=ADC12_VECTOR
__interrupt void ADC12ISR (void)
{
  A0results = ADC12MEM0 & 0x0FFF;                   // Move results, IFG is cleared

}



void Compute_Motion_data()
{
//This fucntion converts the value stored in ADC register to raw //acceleration value adn then to velocity
//It stores a value to DESTR1 which is set to the tomerB register to set PWM

vy=(3.3*(A0results)/4095); //full scale voltage * ADC value/4095 which is for 12 bits

y= (vy)*1.5*(9.8/.8);      // subtract offset from vy which is VDD/2  //.8v = 1.5g (from accelerometer datasheet)

if (vyPrev!=0)
{
velocity+= vy-vyPrev;
}
vyPrev=vy;
}

void controls()
{
DESTR1= 2600-6000*(velocity); // Calculate value for timerB register

if(DESTR1  < 520)           // Keep duty cycle less than 91%
{
DESTR1 = 520;
}
if(DESTR1 > 4680)  // keep duty cycle more than 9%
{
DESTR1 = 4680;
}
}
```

```
void Set_PWM()
{
TBCCR1 = DESTR1;
}

void Time_Count()
{
TACCR0 =1048;     // ( t* 1048567 )set value to change sampling interval
TACCTL0=CCIE;
TACTL = TASSEL_1 + MC_1;   // SMCLK, contmode
}

#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A (void)
{go=1;}
```

10 Appendix D: Bill of Materials

DRIVER and BRIDGE	Part Number		Description	QTY
	2n2222A		Transistor	1
	CPC1580PTR		Optocoupler	1
	HCF4007UBE		Inverter IC (12v)	1
	T983-P5P-ND		12V Wall adapter	1
	F20UP20S		Diode	1
	HIP4082IPZ		Bridge Driver	1
	STP70N10F4		MOSFET (100V, 65A)	4
	x	R2	200Ω	
	x	R1	1kΩ	
	x	R4	3.3kΩ	1
	x	R3	100kΩ	1
		C1	1uF	1
	399-3510-ND	C2,C3	.022uF 500V ceramic	1
	BA30BC0T-ND		3V regulator	
	1N3766		Diode	1
Road Condition	MP502W-BPMS-ND		Bridge Rectifier	2
Module	QM50DY		Power BJT	1

	XR-2206CP		Function generator IC	1
ACCELEROMETER and CIRCUITRY				
	SEN-00252		MMA7260Q Accelerometer Breakout Board	1
	LM358		Low Voltage, Single Supply Op-Amp	1
	x	Roff2	1k	
	x	Roff1	7.5k	
	x	R1	43k	
	x	Rf	160k	
	x	Rhp	5.1M	
	x	Chp	10uF	
	x	Cf	.047uF	
Distance Sensor	GP2D12		Distance Sensor	1

Figure 10-1: Ratings Chart for Transformer Input/Output Connections

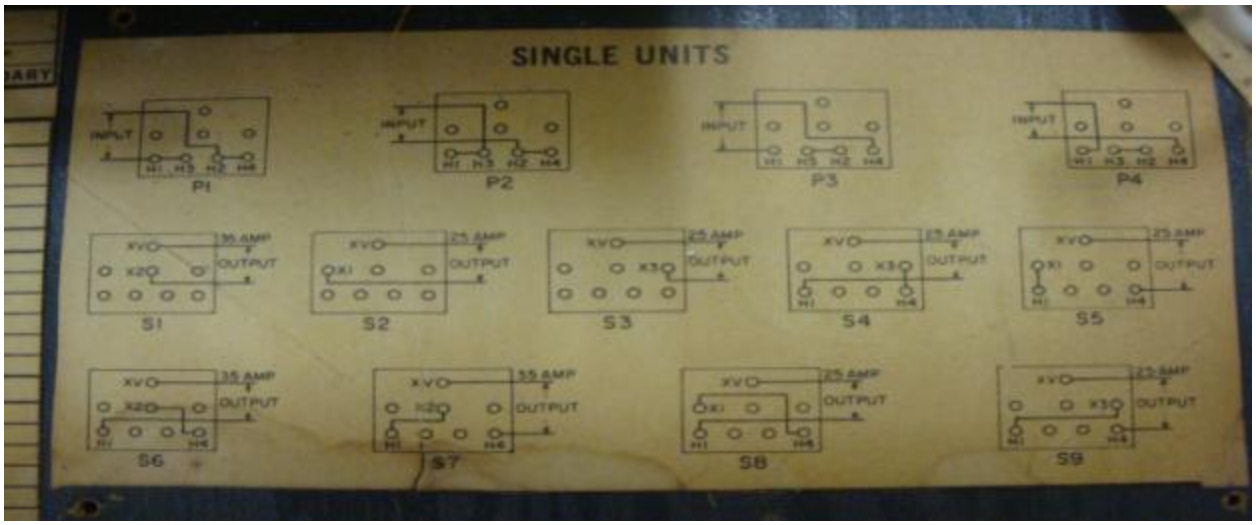

Figure 10-2: Schematic Chart for Primary and Secondary Coil Connections

11 Appendix E: Revised PSPICE Model

```
Control Circuit
.SUBCKT PWMgen in inref out outref
Vtri tri inref PULSE(-1.1 1.1 .01m 1.24m 1.24m .01m 2.5m)
*1.1 keeps duty cycle between 10 and 90 %
Eout out outref TABLE {(V(dsh,inref)-V(tri,inref))}=(-.1u -1) (.1u 1)
Sit in sr tri 0 Switchit
Rsh sr dsh 1m
Csh dsh inref 1u IC=0
.MODEL Switchit VSWITCH(Ron=1m Roff=1G Von=1 Voff=.99)
.ENDS
.PARAM Kp = 10
.PARAM Vsup 30
.PARAM frequency 7
V 1 0 Sin(0 8.5 {frequency})
*V 1 0 AC 8.5 0
L1 1 2 1.641 IC=0
R1 2 3 11.122
C1 3 4 725u IC=0
C2 4 5 626u IC=0
L2 4 7p2 1.609 IC=0
Rair 7p2 7 13
Vvelocity 7 0 0
Vdcoil2 11 0 0
R2 5 6 5
L3 8 9 600u IC=0
Rcoil2 9 10 300m
Rcoil2series 10 11 3
Eforce 6 0 VALUE={.005*I(Vdcoil2)*I(Vdcoil2)*I(Vdcoil2)+.8*I(Vdcoil2)}
Eduty duty 0 TABLE {kp*I(L2)} = (-1,-1) (1,1)
Xpwm duty 0 B 0 PWMgen
Ebridgepwm 8 0 VALUE={-Vsup*v(B)}
Emotionfilter mf 0 VALUE={1*I(L2)}
Rmf mf mfout 1k
Cmf mfout 0 16u
.Probe
.TRAN 2 2 0 1m UIC
.END
```

12 Appendix F: Transfer Function Bode Plot Matlab Code

```
M1 = 1.641;
M2 = 1.609;
D1 = 11.122;
D2 = 5;
D2air = 13;
C1 = 725*10^-6;
C2 = 626*10^-6;
f = .01:.01:25;
w = 2*pi*f;
s = j*w;
Z1 = s*M1;
Z2 = 1./(s*C1);
Z3 = 1./(s*C2);
Z4 = s*M2;

Zin = Z1+D1+Z2;
Branch1 = Z3+D2;
Branch2 = Z4+D2air;

Zout = 1./((1./Branch1)+(1./Branch2));

Transfer_Displacement = Zout./(Zin+Zout);
Transfer_Velocity = Transfer_Displacement.*w;
TDdb = 20*log10(abs(Transfer_Displacement));
TDphase = 360*phase(Transfer_Velocity)/(2*pi);
subplot(2,1,1)
plot(f,TDdb)
subplot(2,1,2)
plot(f,TDphase)
```