MODELING AND CHARACTERIZATION OF A SOLENOID AS A PROPORTIONAL ACTUATOR

by

Cavit Çağrı Özkan

Submitted to Graduate School of Natural and Applied Sciences
in Partial Fulfillment of the Requirements
for the Degree of Master of Science in
Mechanical Engineering

Yeditepe University

2017

MODELING AND CHARACTERIZATION OF A SOLENOID AS A PROPORTIONAL ACTUATOR

APPROVED BY:

Assist. Prof. Dr. Nezih Topaloğlu
(Thesis Supervisor) ....

Assoc. Prof. Dr. Koray Şafak ....

Assoc. Prof. Dr. İpek Başdoğan ....

DATE OF APPROVAL: ..../..../2017

ACKNOWLEDGEMENTS

First of all; I would like to thank to my thesis advisor Assist. Prof. Dr. Nezih Topaloğlu and Assoc. Prof. Koray Şafak for their sensible and valuable advices during my higher education.

I also would like to thank to Tekno Kauçuk and TÜBİTAK since this thesis became possible under their project.

I thank to Can Polat Çıgay for his grateful help.

Also I am thankful to Yeditepe University and Department of Mechanical Engineering for my education and opportunity to complete my master’s degree.

Last but not the least, I would like to thank to my family for their love, support and encourage.

ABSTRACT

MODELLING AND CHARACTERIZATION OF A SOLENOID AS A PROPORTIONAL ACTUATOR

Solenoids are simple electromagnetic actuators, which convert electrical energy to mechanical energy. They can be used in variety of engineering applications such as in valves. They usually operate at on-off mode and it is a nonlinear motion, i.e. the output mechanical motion highly depends on the plunger position. However several attempts have been made recently to operate solenoids as a position-controlled actuator. Therefore accurate modelling of solenoids is essential, in order to use them in position-controlled mode. The objective of this study is to build an accurate solenoid model and to validate the model by experiments.

In this thesis, an off-the-shelf solenoid is modelled. Modelling includes employing governing equations and finite element analysis. An experimental setup for characterization of a solenoids is designed and manufactured. Characterization includes driving the solenoids at different excitation and loading conditions. The displacement of the plunger and the force applied by the plunger to constant and spring type loads are measured. In addition, the frequency response of the solenoid under AC voltage with bias is obtained experimentally. As a result; the built solenoid model is validated with the experimental results, and the characterization of the solenoid as a proportional actuator has been realized.

ÖZET

SOLENOİDİN ORANSAL EYLEYİCİ OLARAK MODELLENMESİ VE KARAKTERİZASTONU

Basit elektromanyetik eyleyici olan solenoidler elektrik enerjisini mekanik enerjiye çevirirler. Valf olarak kullanımı gibi geniş bir yelpazede mühendislik uygulamalarına hizmet etmektedirler. Genellikle açık-kapalı(on-off) modunda lineer olmayan bir hareketle çalışırlar. Bu çıkış olarak verilen solenoid hareketi pistonun konumuna aşırı derecede bağlıdır. Solenoidin pozisyon kontrollü eyleyici olarak kullanılması üzerine son zamanlarda çalışmalar yapılmıştır. Bu yüzden solenoidin bu modda kullanılması için hassas modellenmesi gerekmektedir.

Bu tezdeki çalışma çerçevesinde piyasada bulunan bir solenoid modellenecek ve deneysel olarak karakterize edilmiştir. Modelleme çalışmaları içersinde korunum denklemleri ve sonlu eleman analizi bulunmaktadır. Solenoid karakterizasyonu için bir deney düzeneği tasarlanmış ve üretilmiştir. Karakterizasyon çalışmaları içersinde solenoidin çeşitli salınımlarda çeşitli yüklere karşı sürülmüştür. Pistonun deplasmanı ile yay ve sabit yüklere karşı uyguladığı kuvvet ölçülmüştür. Tezin amacı modellemin deneysel sonuçlarla doğrulanmasıdır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

c	Damping Constant
d	Wire Diameter
D	Mean Diameter
F	Force
F_{sol}	Solenoid Force
G	Shear Modulus
i	Current
k	Spring Constant
$L(x)$	Inductance
m	Mass
n	Number of Turns
R	Resistance
V	Voltage
x	Displacement
Φ	Flux
CET	Common Elapsed Time
COM	Communication Port
EMF	Electromotive Force
FEA	Finite Element Analysis
FEM	Finite Element Method
PWM	Pulse Width Modulation

1. INTRODUCTION

Solenoids are common actuators and they can be found in a variety of engineering applications. This is probably because of their simplicity of design which leads to reliability in the long term. They can be used as in direct solenoid actuation injector for diesel engine [1], or as automobile starter solenoid as solenoid valve, or as a door lock [2].

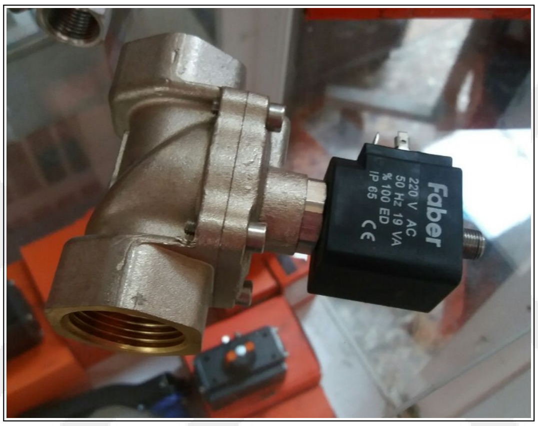

Figure 1.1. Solenoid Valve

Solenoid differ from each other by certain properties such as linear or rotary, short or long stroke etc. However certain components does not change. In Figure 1.2; solenoid components are showed for linear push pull type solenoid. In this type of solenoid; the plunger and the cap are the moving parts of the solenoid, while coil is wrapped with the housing and air gap is present between the cap and the housing.

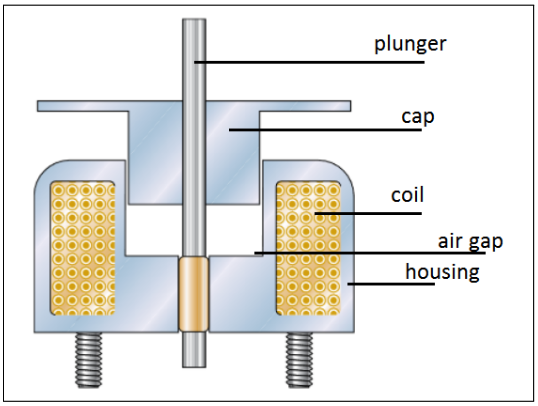

Figure 1.2. Solenoid Components

Working principle of solenoid is pretty straightforward. The current passing through the coil causes the magnetization, which pulls the plunger into the core. Therefore they traditionally operate in binary (on-off) mode. In Figure 1.3; states of the solenoid can be seen. De-energized state represents "off" state and energized state represents "on" state of the solenoid.

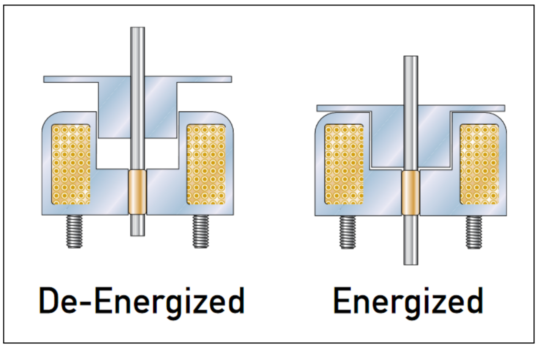

Figure 1.3. Solenoid States

Also there are studies on using solenoids as actuators in active engine mounts. This thesis has very close relationship with concept of solenoid usage in engine mount, since it is written under project for designing an active engine mount.

Engine mounts are used in isolating vibration in particularly in automotive industry.

Vibration occurs from two different sources. One is base excitement type which is caused from road conditions like bumps, other source rotating unbalanced type and this is occurring because of working engine itself. However there are multiple types of engine mounts in order to achieve their duty such as passive, semi-active and active. As it can be guessed easily active mounts are more effective and complicated than the passive ones. Actuators such as voice coils are usually used in active mounts.

Using solenoids instead of voice coils as actuator in engine mount is relatively new subject. A solenoid-based actuator is more compatible with the harsh environment of the active mount then voice coil [3].

The most common practice for solenoids is to use them as switching actuators. However using solenoids as proportional actuators is more challenging. This is a non trivial task because of the highly non linear magnetic characteristic of the device [4].

Traditionally solenoids are fed with PWM signal in order ability of drive them with higher voltage. But using them as proportional actuators is less popular action. In order to understand proportional actuator role of solenoid, there has to be an accurate mathematical model. Converting solenoid to proportional actuator mathematically has been studied before [5], [6]. However there is fewer literature on frequency response of solenoid as a proportional actuator.

In this thesis; solenoid will be modelled based on its governing equations. This dynamic model's frequency response will be validated with experiments. Magnitude ratio and phase angle will be under scope in frequency analysis. Also static response of solenoid also will be discussed. In that part; experiments will be compared with finite element method solutions. Displacement force relationship will be scoped in this part.

As mentioned before this thesis consists two major parts, simulations and experiments. The experiment setup used, the characterization tests performed and results gathered have some novel features. In that sense, present study exhibits uniqueness.

2. MODELLING AND SIMULATION

This chapter consists of two sections. In Section 2.1; static response of the system will be simulated and studied with finite element method. In Section 2.2; solenoid will be dynamically modelled after derivation of its governing equations.

2.1. STATIC RESPONSE OF THE SYSTEM

Static response of analysis of solenoid is studied under finite element method analysis. Finite element method provides a numerical solution to complex problems and it is widely used in academia and industrial field. With the power of finite element method; the relationship between distance of the plunger and the force applied by the solenoid is examined. This data will be compared with the experimental data from the future static tests. Therefore purpose of the fea modelling is to validate the results.

In order to solve the analysis an axissymetric model is created and solved in Finite Element Methods Magnetics (FEMM) software [7]. It should be noted that since the lack of opportunity to know the exact inside geometry and the materials of the solenoid parts, geometry and material selection is estimated.

The model which can be seen on Figure 2.1; is an axisymmetric view of the solenoid. It represents the actual solenoid used in experiments which is Geeplus Push Pull Solenoid Size 490. Model has the same number of turns of 360 and 24 AWG wire as the actual one. In model it is estimated that the plunger and surrounding block is made out of pure iron. Geometrical estimations can also be seen on Figure 2.1.

Figure 2.1 and Figure 2.2; shows model and meshed model in an respected order.

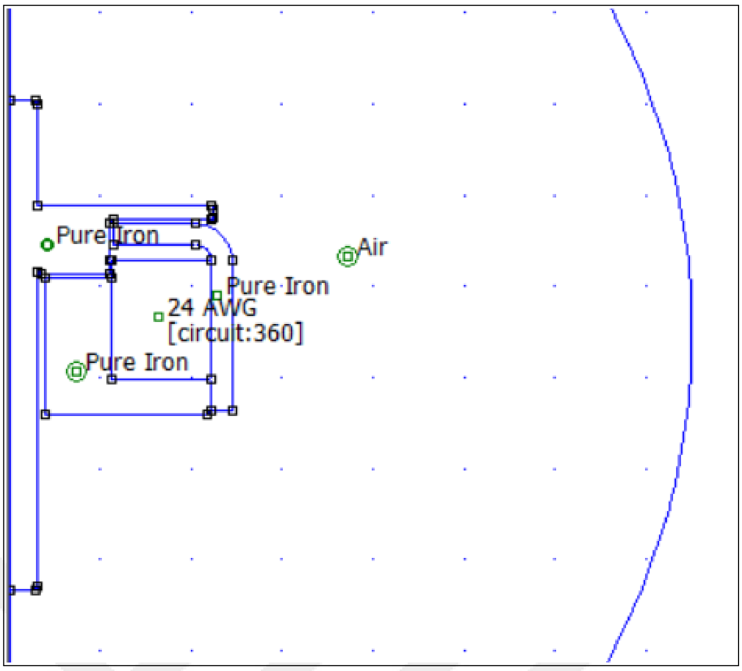

Figure 2.1. Model for Finite Element Analysis

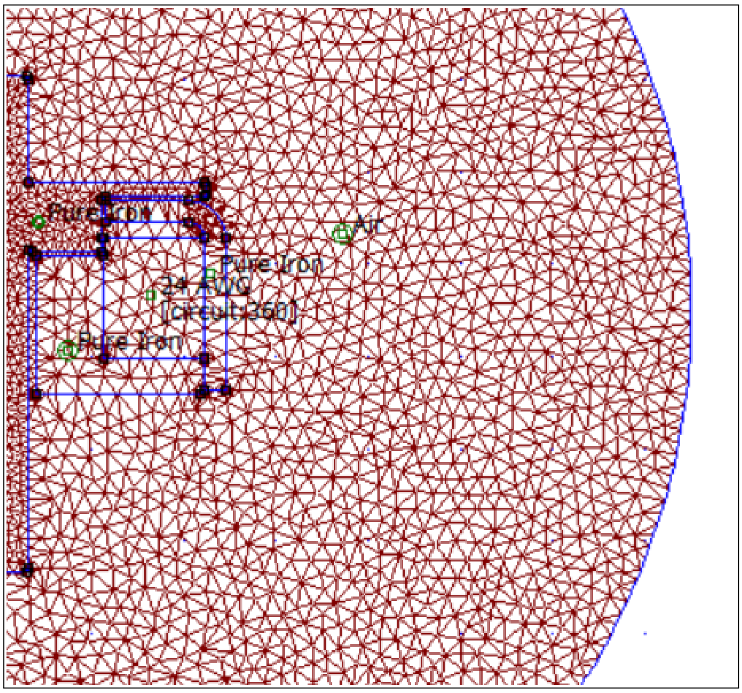

Figure 2.2. Meshed Model for Finite Element Analysis

Finite Element Method Magnetics software is run with precision of 1e-8. Using OctaveFEMM package [8] program under GNU/Octave program is run and solved from 0 displacement of armature to 4 mm displacement with 0.1 mm step size. Flowchart used for this process can be found on Figure 2.3.

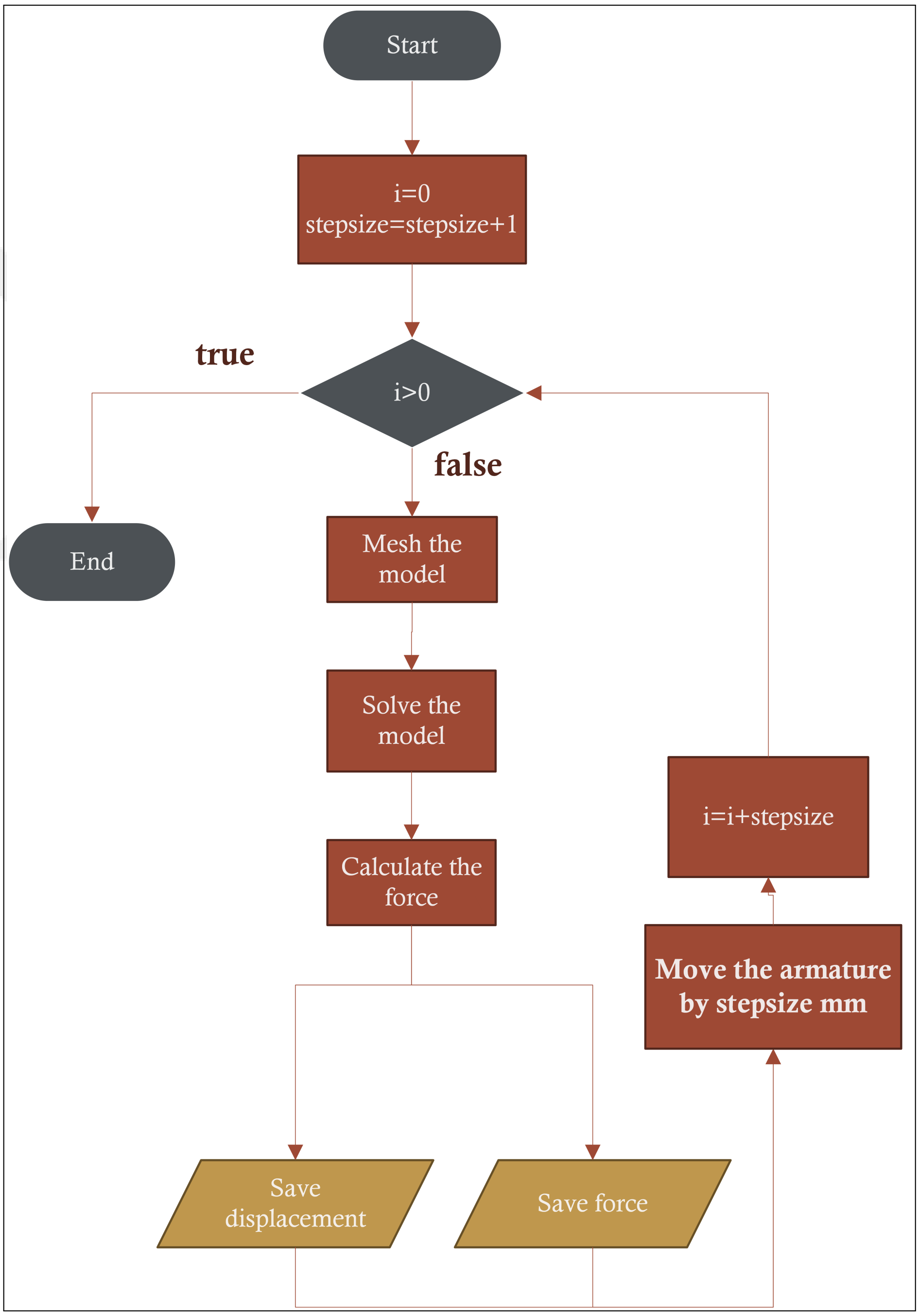

Figure 2.3. Finite Element Analysis Control Flowchart

If we follow the flowchart; we see that eight nine steps are present between its start and end points. This flowchart represents the control script for finite element analysis. Script start with stepsize of 0.1 and i of 0 in order to initialize which is represented in first box. Second box represents the if statement control of the script. This box orders that while i value is smaller than 4, script meshes, solves and calculates the force in a respected order and they are represented in third, fourth and the fifth boxes. After the calculation of the applied force, script saves displacement and force data. This saving data process is represented in sixth and the seventh boxes. Then the armature is moved by stepsize value which is very crucial point of the script and it is represented in eighth box. Which is followed by refreshing the i value in the last box. Lastly it is connected to the control box which means scripts is run till the value i reaches 4. Value i represents the travelled distance of plunger.

Force vs displacement relationship of solenoid according to finite element simulation can be seen on Figure 2.4. This figure is represented for visualizing purposes. It will be studied in discussion chapter along with experiment results.

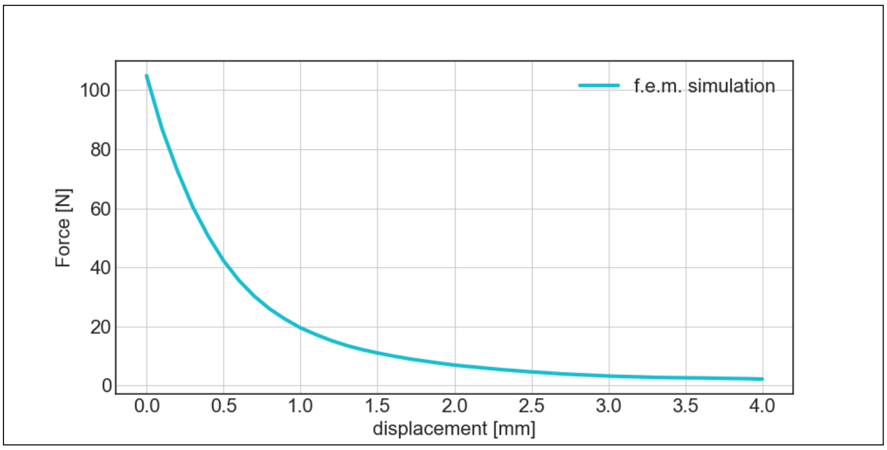

Figure 2.4. FEM Simulation Results

2.2. FREQUENCY RESPONSE OF THE SYSTEM

Defining complex systems as a group of simpler systems is a useful way and generally the first step to modelling. In problem of studying the solenoid these simpler subsystems would be electrical and mechanical. Therefore in this chapter first the governing equation of motion of solenoid will be derived and after that solenoid model will be constructed using those properties. The data from frequency response simulation will be compared with the future frequency experiment results. Therefore purpose of the the modelling is to validate the results.

If we look up solenoids working principle by step by step, we see that electrical input flowing through solenoids coil creates a magnetic force. This magnetic force pushes the plunger into the core of the body and therefore that force is mechanical output of the system. However after the cutting down the voltage input to the system, a spring returns the plunger to its original position which is out from the body. To understand the input and output relationship of the system; we need to understand its subsystems as well.

When we look up what happens inside of solenoid when there is voltage applied; we see the R–L circuit. However since a solenoid itself is a dynamic component, inductance value is also dynamic. In our case inductance is a function of plunger displacement instead of constant.

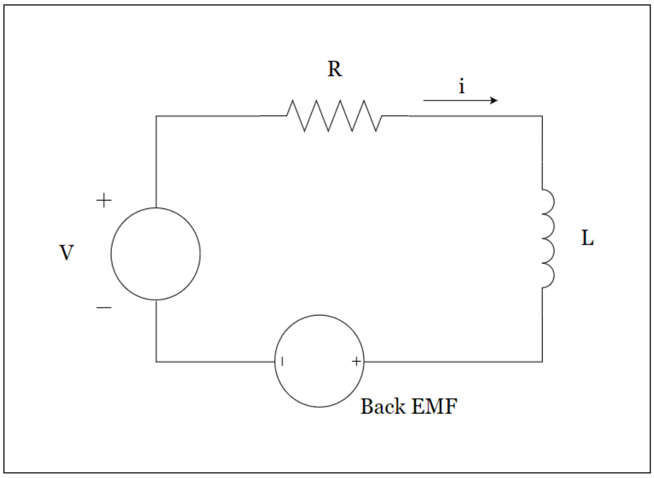

Figure 2.5. Electrical Subsystem

Figure 2.5 shows electrical subsystem of solenoid. If system is assumed as a magnetically linear; electrical equation for solenoid can be yield from Kirchoff's Voltage Law.

Therefore we can write;

$$V = R\,i + \frac{d\Phi}{dt} \tag{2.1}$$

$$\Phi = L(x)\,i \tag{2.2}$$

where V, R, i, Φ, t, $L(x)$ are voltage, resistance, current, magnetic flux, time and inductance in a respected order.

$$V = R\,i + L(x)\,\frac{di}{dt} + \underbrace{\boxed{i\,\frac{dL(x)}{dx}\,\frac{dx}{dt}}}_{\text{Back EMF}} \tag{2.3}$$

$$\frac{di}{dt} = \frac{1}{L(x)}\left[v_0 - R\,i - i\,\frac{dL(x)}{dx}\,\frac{dx}{dt}\right] \tag{2.4}$$

In our and very most systems push solenoid moves against spring, and after the cut of voltage supply spring force returns plunger to its initial position. Mechanical equation of solenoid can be derived for Newton's Second Law.

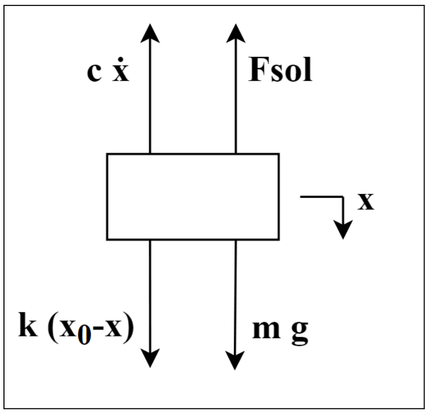

Figure 2.6. Free Body Diagram

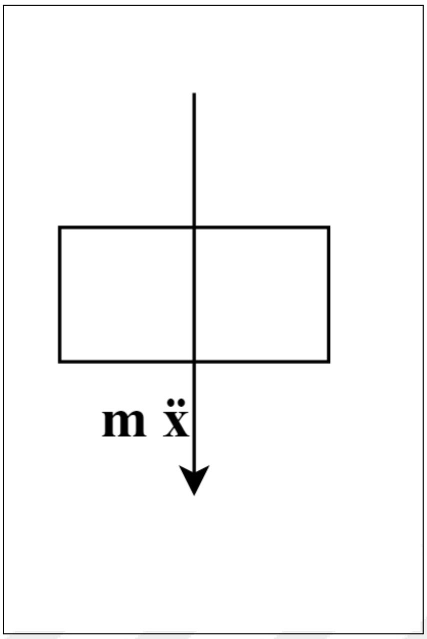

Figure 2.7. Kinetic Diagram

$$\Sigma F = m\,a \tag{2.5}$$

$$\Sigma F = m\,\frac{dv}{dt} = m\,\frac{d^2x}{dt^2} \tag{2.6}$$

where F_{sol}, k, x, c, m are solenoid force, spring constant, displacement, damping coefficent and mass of plunger in a respected order.

$$-F_{sol} - k\,(x - x_0) - c\,\frac{dx}{dt} + m\,g = m\,\frac{d^2x}{dt^2} \tag{2.7}$$

$$\frac{d^2x}{dt^2} = \frac{1}{m}\,\left[-F_{sol} - c\,\frac{dx}{dt} - k\,(x - x_0) + m\,g\right] \tag{2.8}$$

$$F_{sol} = \frac{1}{2} i^2 \frac{dL(x)}{dx} \tag{2.9}$$

Therefore the equation of motion of the system becomes:

$$m\,\frac{d^2x}{dt^2} = -\frac{1}{2}i^2\frac{dL(x)}{dx} - c\,\frac{dx}{dt} - k\,(x - x_0) + m\,g \tag{2.10}$$

In this point there are two ways to determine the inductance value; first one is calculating by its formula and the second one is measuring it at different positions. Since we don't have the opportunity to know the exact inside geometry and properties of the solenoid, second approach of determining the inductance value is chosen. In order to measure inductance accurately specific test equipment LCR meter is needed. This device is found available in RF Lab of Electric-Electronics Department.

Table 2.1. Inductance of solenoid at different dispalecements

disp.[mm]	**ind.[mH]**
0.5	3.411
1.5	3.417
3	3.439
4.5	3.460
6	3.468
7.5	3.471
9	3.475

In order the have a proper model; a polynomial function is fitted using data presented in Table 2.1. This function will be used in simulation model. The coefficients of second degree polynomial $ax^3 + bx^2 + cx + d$ is shown below. Python with Numpy and Scipy library is used in order to get polynomial fit function of inductance data [9],[10]. Same results could also be obtained with MATLAB or GNU/Octave with similar commands.

Table 2.2. Coefficients of Inductance Function

a	b	c	d	R^2
-8.545e-5	2.701e-4	1.241e-2	3.402	0.921

Figure 2.8 shows experimental measurements of inductance alongside with fitted 3rd degree function. This functions constants can be found on Table 2.2, and its properties can be found in Table 2.3.

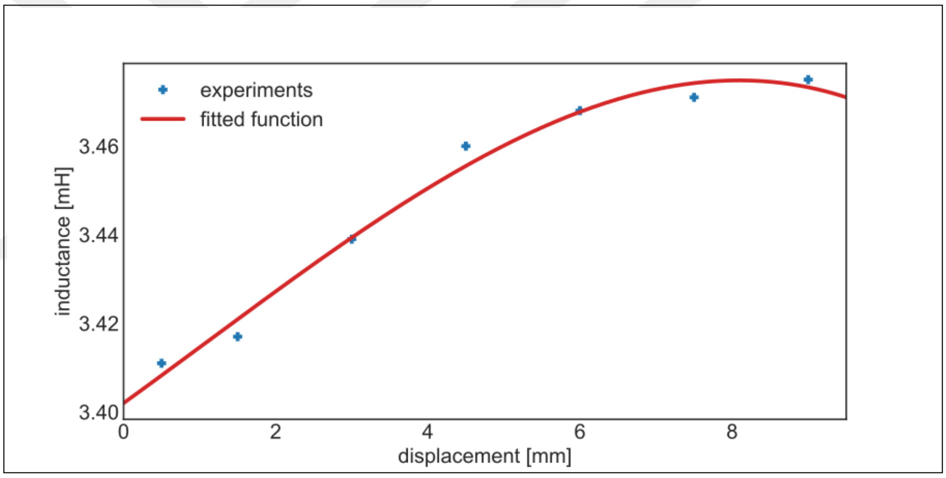

Figure 2.8. Inductance Measurements and Fitted Function

Table 2.3. Coefficients of Inductance Function

slope	intercept	r value	p value	standard error
0.008083	3.411759	0.954736	0.000817	0.001126

Figure 2.9 shows constructed block diagram of solenoid model. This figure is screenshot of

model used in MATLAB/Simulink.

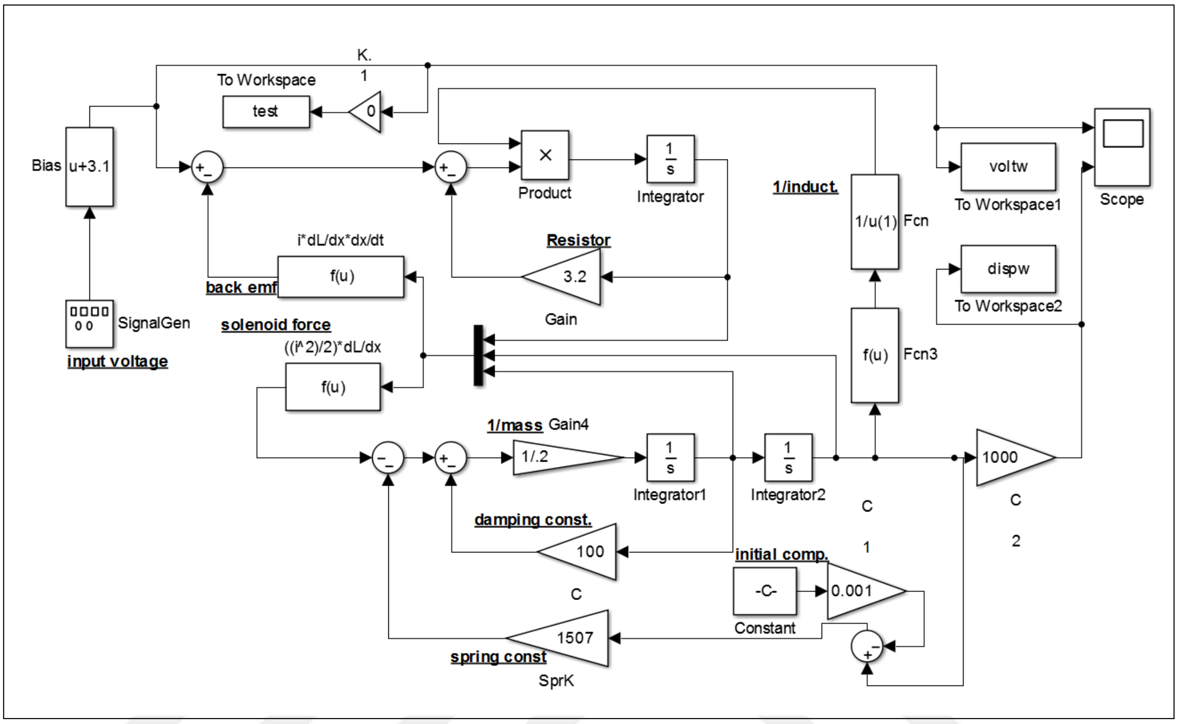

Figure 2.9. Block Diagram of Solenoid Model

Input of the system consists two parts; one is oscillating signal generated by SignalGen. In our system this part has sinusoidal with amplitude of 1 V. Its frequency is controlled within a script which will be discussed later. In blog diagram Product, Integrator, Gain are the elements for solving the differential equation for electrical subsystem. On the lower half of the block diagram where Gain4, Integrator1, Integrator2, C, SprK are the mechanical differential equation of solenoid which is mentioned earlier.

Inductance function is inserted in first derivation form with respect to displacement in back emf and solenoid force boxes. Mass of the moving part of the solenoid and spring constant is measured and calculated and putted into respective boxes. The value of initial displacement is measured from experimental setup, when there is lack of AC component of input signal which means only constant DC voltage supply is present there is no oscillations of plunger. Also it should be noted that in order to eliminate the transient effects, the simulation is run for fifty-five seconds and only last five seconds are scoped.

Since initial compression is used as a initial condition; respective integrator box which relates to displacement also should have the same value. Therefore initial condition value is inserted in to “Integrator2” box as well. Also it should be noted that as solver Runge-Kutta method with 1e-8 fixed step size is used.

Table 2.4. Parameters of the Model

$m\ [kg]$	0.200
$k\ [N/m]$	1507
$c\ [Ns/m]$	100
$R\ [\Omega]$	3.2

The used parameters in the model are summarized and shown in the Table 2.4.

This model is run inside of a script for determining its magnitude and phase results in changing frequency range. Since it is planned to perform experiments from 500 mHz to 20 Hz the code is ran in the same range with step size of 500 mHz. MATLAB/Simulink is used to run this block diagram and simulate the frequency response of solenoid. Code for analyzing magnitude and phase values of voltage and displacement can be found in Appendix 6.

Magnitude ratio vs frequency and phase difference vs frequency relationships of solenoid can be seen on Figure 2.10 and Figure 2.11. These figures are represented for visualizing purposes. They will be studied in discussion chapter along with experiment results.

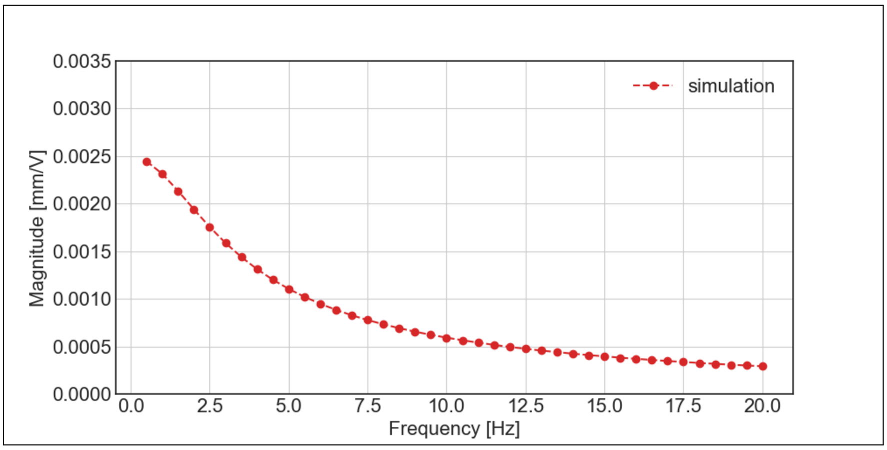

Figure 2.10. Simulated Magnitude Ratio

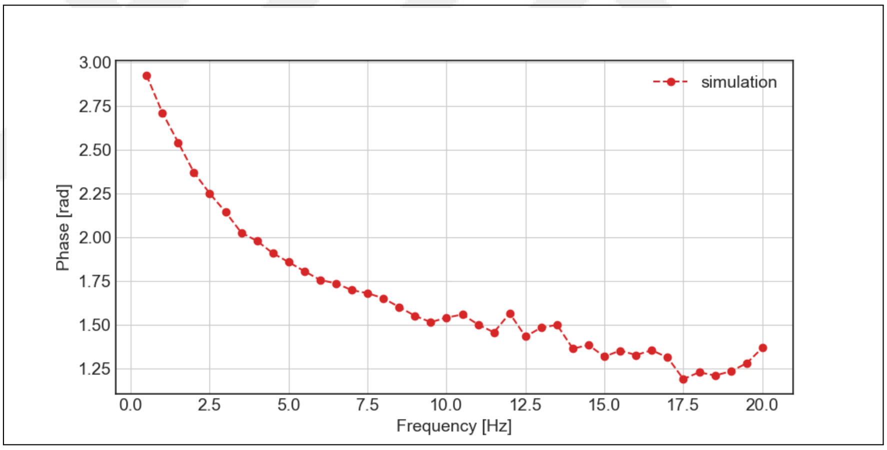

Figure 2.11. Simulated Phase Difference

3. EXPERIMENT SETUP

This chapter consists of three sections. In Section 3.1; mechanical design of the experimental setup will be explained under two subsections; for which are designing the setup itself and study of selecting springs. In Section 3.2; process for designing required electrical circuit will be explained. This section consists circuit designs in order to generate desired signal and simulation of these circuits. In Section 3.3; the software design process will be explained. This design is also necessary in order to perform solid experiments.

3.1. MECHANICAL DESIGN

This section consist of two subsections. In Subsection 3.1.1; design focus and process for experiment setup discussed from its 3D computer aided model to its assembly. In Subsection 3.1.2; the process for selecting spring with optimal rigidity will be explained.

3.1.1. Setup Design

The main idea behind the experiment setup is its similarity to the solenoid used in active engine mount model. In the active engine mount solenoid moves against static and rubber spring force while it is assembled upside down. This means when voltage is on plunger is on the highest position.

In experiment setup same idea is used. However in experiment setup is designed in order to perform various tests and experiments on a solenoid. The major concerns of the system are cost and ease in manufacture, the ability of the control the variety of parameters of the system i.e. initial length of the spring, stroke of the solenoid, etc.

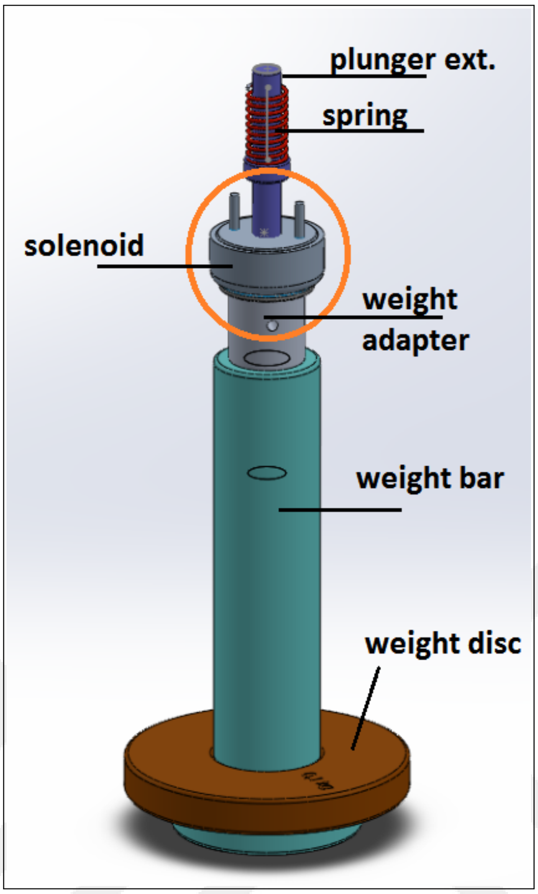

Figure 3.1. Solenoid Assembly

In the experiment setup; the solenoid moves against two force sources. One is static which is caused from weight of solenoid, weight bar and discs. The other force source is optional and caused from spring. The design allows wide range of initial spring length, limiting the movement of the plunger, existence of preload and ability to apply heavy loads on solenoid.

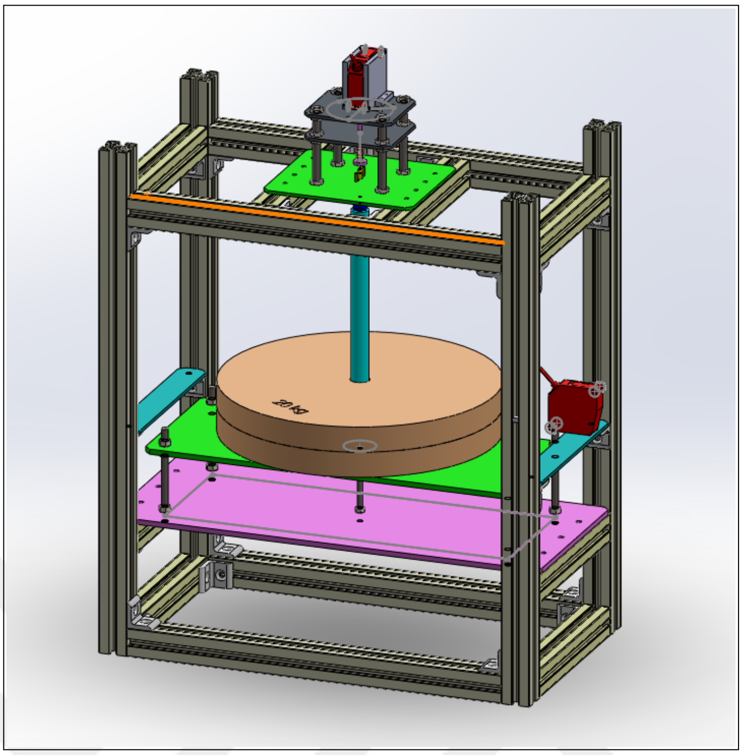

Figure 3.2. 3D Model of Experiment Setup

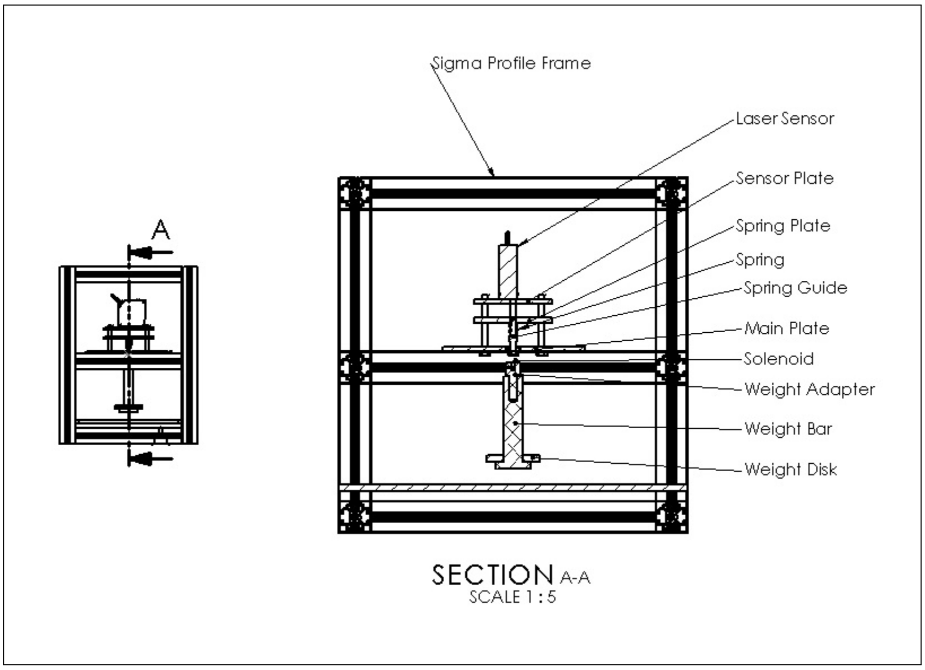

Figure 3.3. Section View of Experiment Setup

There are several changes made to the initial design during the thesis. For example; in the first design, the limitation of plunger movement is planned to achieve by limiting the movement of weight bar by blocking its travel at a particular point with another plate underneath.

However after manufacturing and assembly of the setup this was changed to simpler and more practical way. In that modification solenoid's movement limitation is achieved by nuts. This is better solution in terms of reliability and accuracy level.

Geeplus Linear Push Pull Solenoid Size 490 is used in experiments. Solenoid used has number of turns of 360 and AWG number of 24.

Figure 3.4. Geeplus 490

Sensor used for displacement readings is SICK Precision OD5-30T05 which is very accurate industrial short distance laser sensor. It has a measuring range of 25 to 35 mm and frequency of 10 kHz.

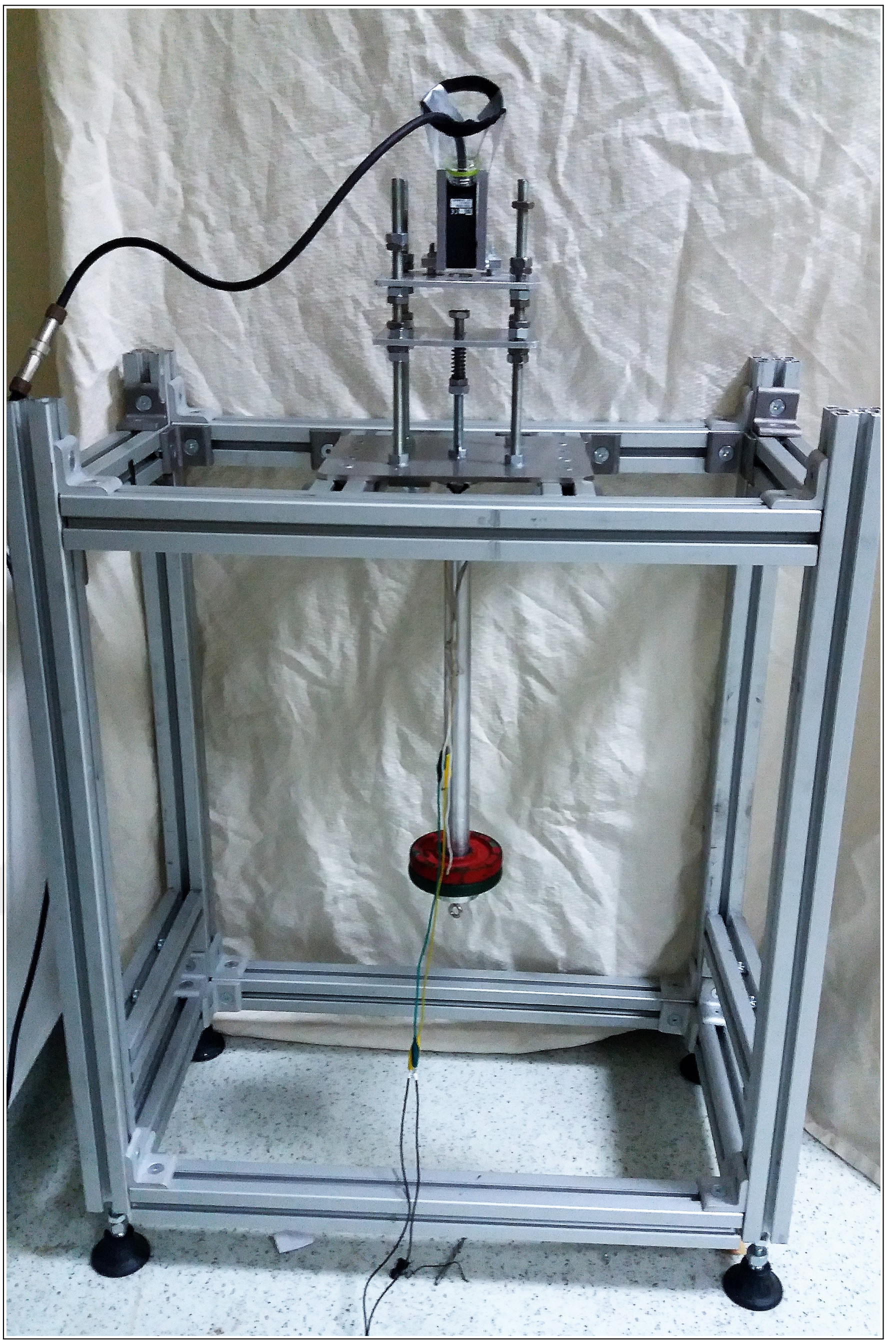

Figure 3.5. Assembled Experiment Setup

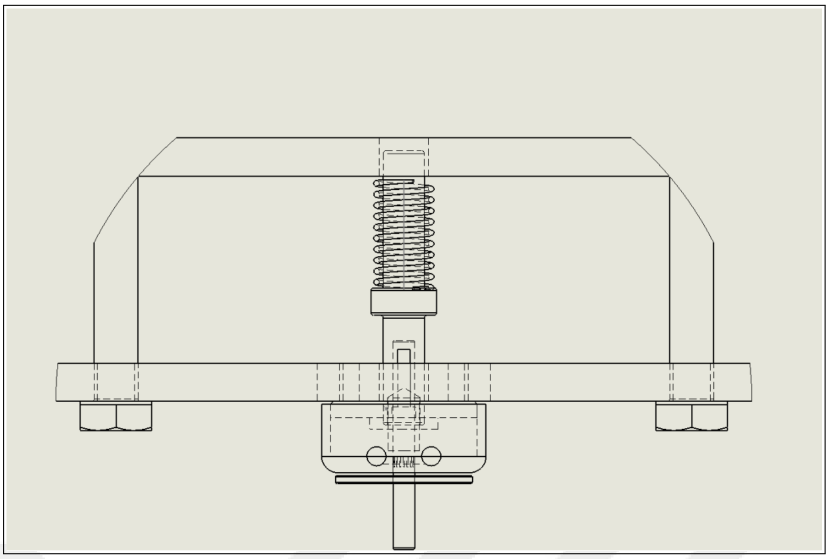

Figure 3.6. Plunger Extension and Spring Drawing

Figure 3.7. Plunger Extension and Spring

General work-flow of the setup is as follows. According to the applied signal, a magnetic force pulls the plunger into the core, and while it moves under the resistance forces from the weight and the spring; a laser sensor measures its distance as well as voltage is measured by micro-controller card. And change in input signal cause change in output. This work-flow is described on a flowchart which can be seen on Figure 3.8.

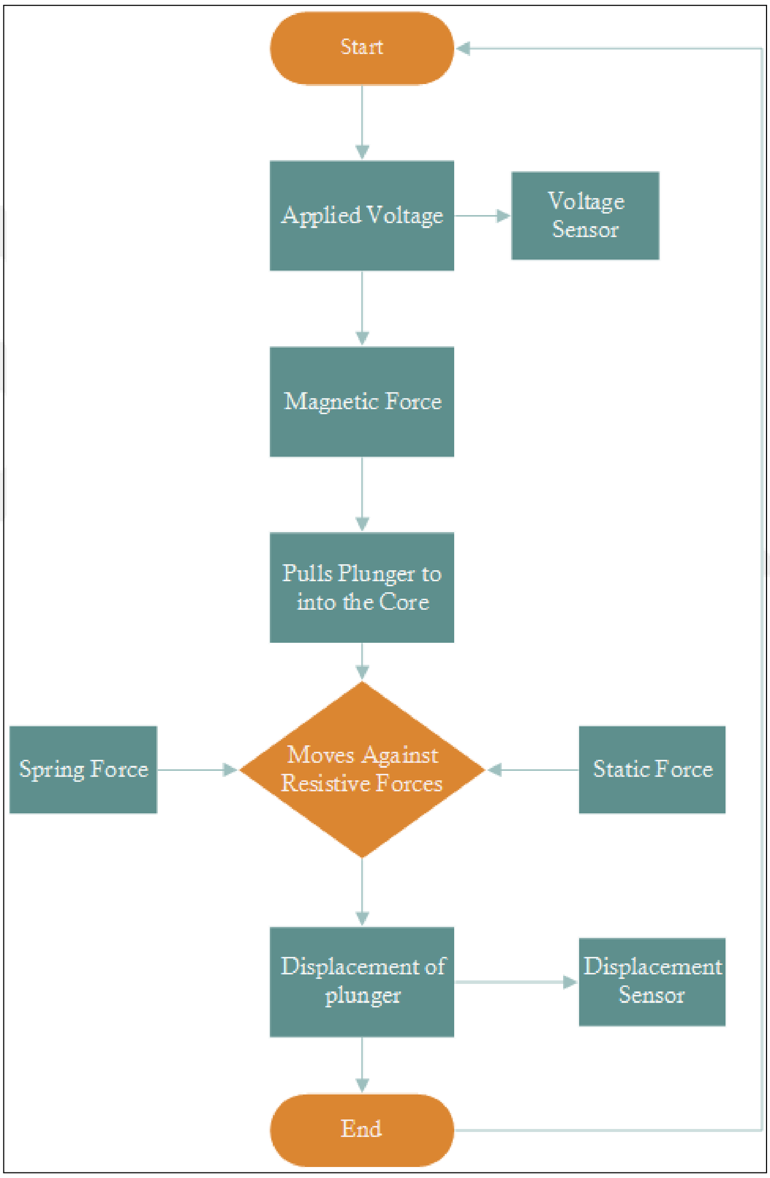

Figure 3.8. Solenoid Movement Flowchart

As we follow the flowchart; we see that main path between start and end points, contains five steps which are applied voltage, magnetic force, pulling the plunger, moving, displacement of the plunger. Measurements are done in applied voltage by voltage sensor step and displacement step by displacement sensor step. External forces are represented as inputs to the moving against resistive forces step. These external forces are represented in two different boxes of static force and spring force.

3.1.2. Selection of Springs

Springs are selected in a range after static tests are applied to solenoid. Spring selection is somehow very important for the results of the frequency response experiments. Stiffness of the spring should be high enough to prevent the solenoid's plunger to travel fully, while it needs to be low enough therefore oscillation of plunger can be detected. Two sets of 10 springs are purchased over the time of experiments. Among those 20 spring, the most optimal one is used in frequency response experiments. The springs of the first set have constants that vary between 0.98 and 20.32 N/mm while the second set has 0.15 and 4.64 N/mm. First set springs and second set springs can be seen on Figure 3.9 and Figure 3.10 respectively.

Figure 3.9. First Spring Set

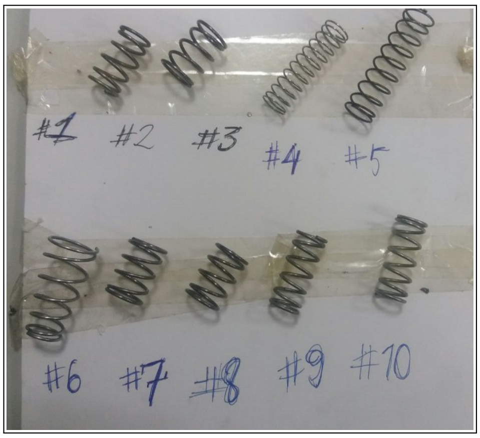

Figure 3.10. Second Spring Set

According to second set springs' potential use on the experiment setup their spring constant is also measured by using weight discs and displacement sensor. Measurements for spring constant is performed for each spring's displacement under two different weight added. Table 3.1 shows measured and calculated spring constants for second set springs. Measured column values are average of two different measurements. This measurements are collected by using famous Hooke's Law.

$$k = \frac{F}{x} \tag{3.1}$$

Calculated values of spring constant is obtained from geometrical formula of compression spring constant.

$$k = \frac{G\, d^4}{8\, D^3\, n} \tag{3.2}$$

G, d, D, n are the values for the shear modulus of the spring material, the diameter of the wire, the mean diameter which is measured from center to center of cross sections and the number of turns respectively. It should be noted that it is accepted normal that geometrical formula for spring constant can give 10 per cent more or less of the actual spring constant. Also since none of the springs are custom made, they are literally selected by hand from the basket; some of the springs' material is different than what it is assumed. That is the reason for some springs has very different values for measured and calculated.

Table 3.1. Spring Constants of Second Set

#	Calculated [N/mm]	Measured [N/mm]
1	1.625	1.448
2	4.317	5.979
3	2.860	10.108
4	0.146	0.213
5	0.146	0.435
6	1.381	1.185
7	2.637	3.918
8	2.626	3.781
9	3.424	4.222
10	4.640	4.725

3.2. ELECTRICAL CIRCUIT DESIGN

Solenoids are traditionally are fed with pulse width modulation (PWM) signals. Using this on-type of signal provides opportunity to supply solenoid with higher voltages. However if the desired output of solenoid is to oscillate the plunger around a fixed point things will get messier. In this point the input should be periodical signal with an off-set which also can be described as AC+DC. In the experiments the alternating part of that signal will be sinusoidal wave. Example signal with its components can be seen on Figure 3.11.

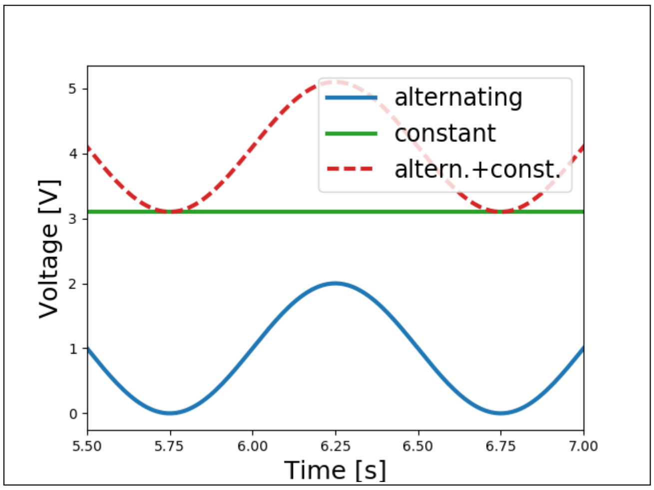

Figure 3.11. Desired Signal and its Components

This type of challenge was not the issue in static tests. Since static test did not require a electrical component other than direct connection to power supply, there was a need for electrical circuit in other tests.

However in our case; a few circuits are designed, simulated and applied in order to get desired output. The first circuit was applied to the system to make sure it is functioning properly.

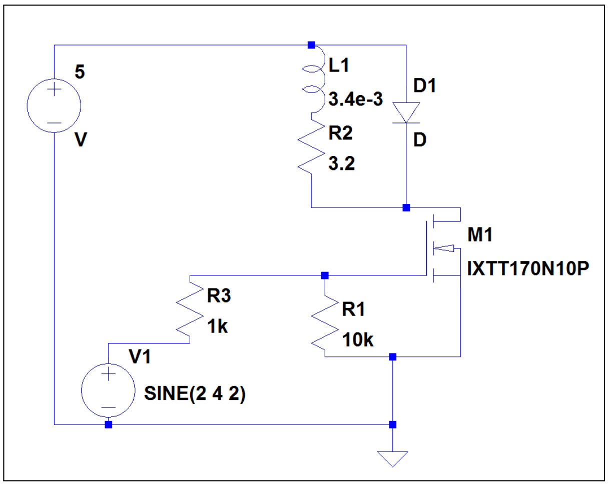

Figure 3.12. Switch Circuit

The circuit which can be seen on Figure 3.12, works as a switch. The voltage supplied from power supply, eliminates by function generator through the transistor After successful performance of this test, it is sure that solenoid acts like it supposed to and works with no problem with altering signals.

Major drawback for constructing an alternating voltage with offset is the high current need of solenoid to move. The current in experiments is foreseen to be high up to 1 Ampere.

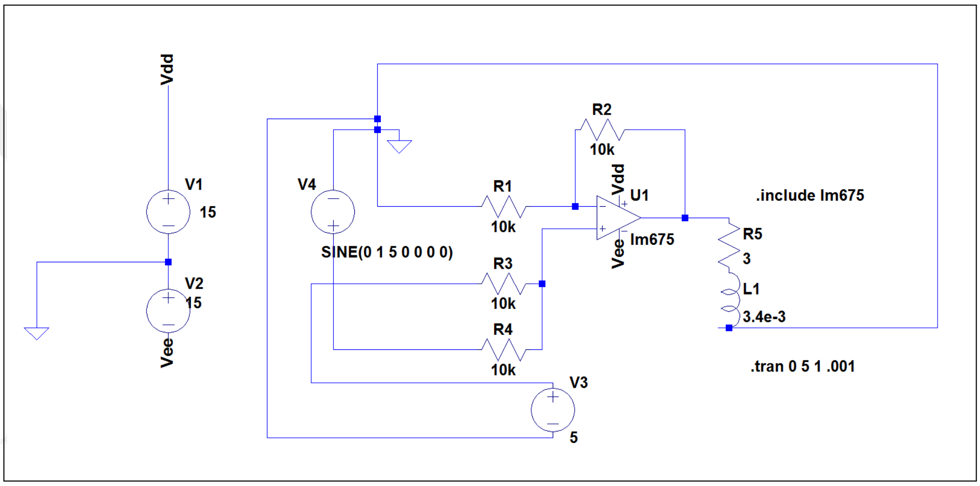

Figure 3.13. Non-Inverting Summer with Power Op-Amp

The circuit which can be seen on Figure 3.13; uses a power operational amplifier(power op-amp) to amplify input signal. The output of operational amplifier is capable of passing high currents. This type of circuit is called non-inverting summer and formula of output voltage can be yield from basic op-amp properties.

$$V_{OUT} = \left(1 + \frac{R2}{R1}\right) \cdot \left(V3 \cdot \frac{R3}{R3 + R4} + V4 \cdot \frac{R4}{R3 + R4}\right) \tag{3.3}$$

All of the resistors are selected same since what is wanted from the system is basically sum up the signals from function generator and power supply and making it capable of passing high current.

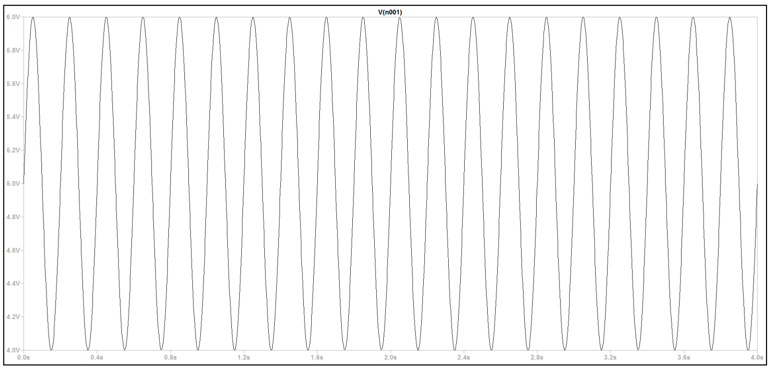

Figure 3.14. Simulation Output of Non-Inverting Summer with Power Op-Amp

As it can be seen on Figure 3.14; expected result is yield. However on the experiment setup; this circuit did not work. Output of system was not stabilized when solenoid took place on the circuit. This is probably due to changing inductance of solenoid. Therefore another circuit design is became necessary.

The final circuit designed is able to give desired output in simulation and in reality. This circuit can be seen on Figure 3.15.

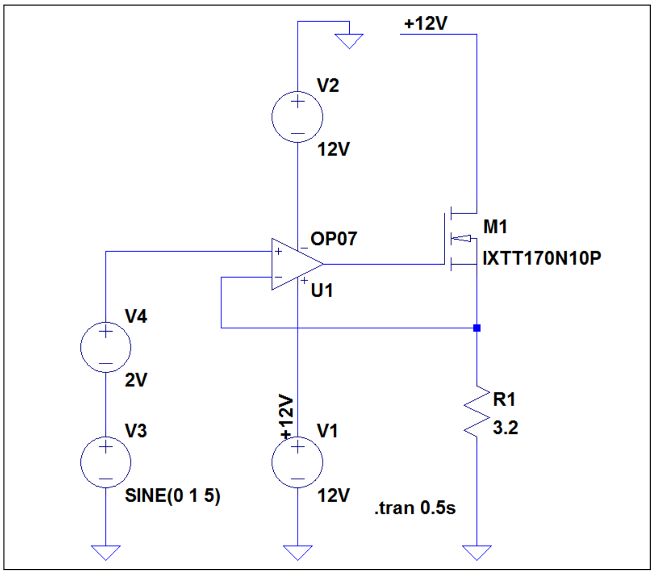

Figure 3.15. Current Sink Circuit

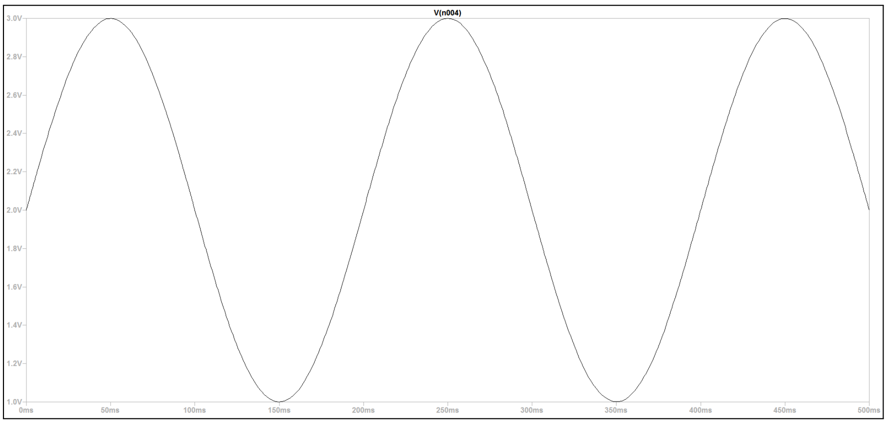

Figure 3.16. Simulation Output of Current Sink Circuit

This circuit is described and named as current sink circuit. In this circuit a regular operational amplifier is used, while its job is only keeping voltage same between its positive and negative inputs. However a major key role in this circuit belongs to transistor. It is selected as it is able to pass through high current. Voltage measurement is performed on R1 resistance where R1 represents solenoid.

3.3. SOFTWARE DESIGN

Usually it would be considered unusual having a section for software design in mechanical engineering thesis, but for in this thesis, it is necessary for explanation concerns. In frequency analysis tests; the focus will be on magnitude of output to input ratio and phase between output and input signals. Therefore major key role on this analysis would be the common time. Data of input voltage signal and data of output displacement signal necessarily start and end at the exactly same time. Only and only that would make analysis valid.

However both of those signals are measured from different sensors. Therefore there is need a trigger to start and end the readings from sensors at the same time. But as an another issue micro-controller cards do not have a real time clock by nature. It has a clock which can be used for measured time measurements, however it does not have the high level accuracy. Sensor for voltage reading is monitored under Windows OS's communication (COM) port; while reading by displacement sensor's is only possible under its specific program. Even though sensor reading box has different output ports which are USB and RS232; it does not send the readings right away. In order to get and save data from displacement sensor; its specific software SOPAS Engineering Tool v3 needs to be executed and send command to read data to the sensor on this software.

This problem is solved by saving both voltage and displacement data with timestamps of computer's BIOS time. SOPAS Engineering Tool software has ability to save data with BIOS timestamps and as serial port communication monitoring software Realterm has that ability with milliseconds accuracy in beta version. After saving BIOS time vs voltage and BIOS time vs displacement Python script is used for detecting the common type elapsed. Python script run can be found in Appendix 6. Flowchart of this process can be seen on Figure 3.17.

After the extraction of common elapsed time by Python script; analysis for frequency analysis is performed. In this analysis the output is text file that has data name, frequency, mean magnitude of displacement to mean magnitude of voltage ratio, mean value of phase difference between voltage and displacement. In order to get those mentioned parameters, calculation for zero crossing values and maximum and minimum values of voltage and displacement for relating zero crossings is also performed.

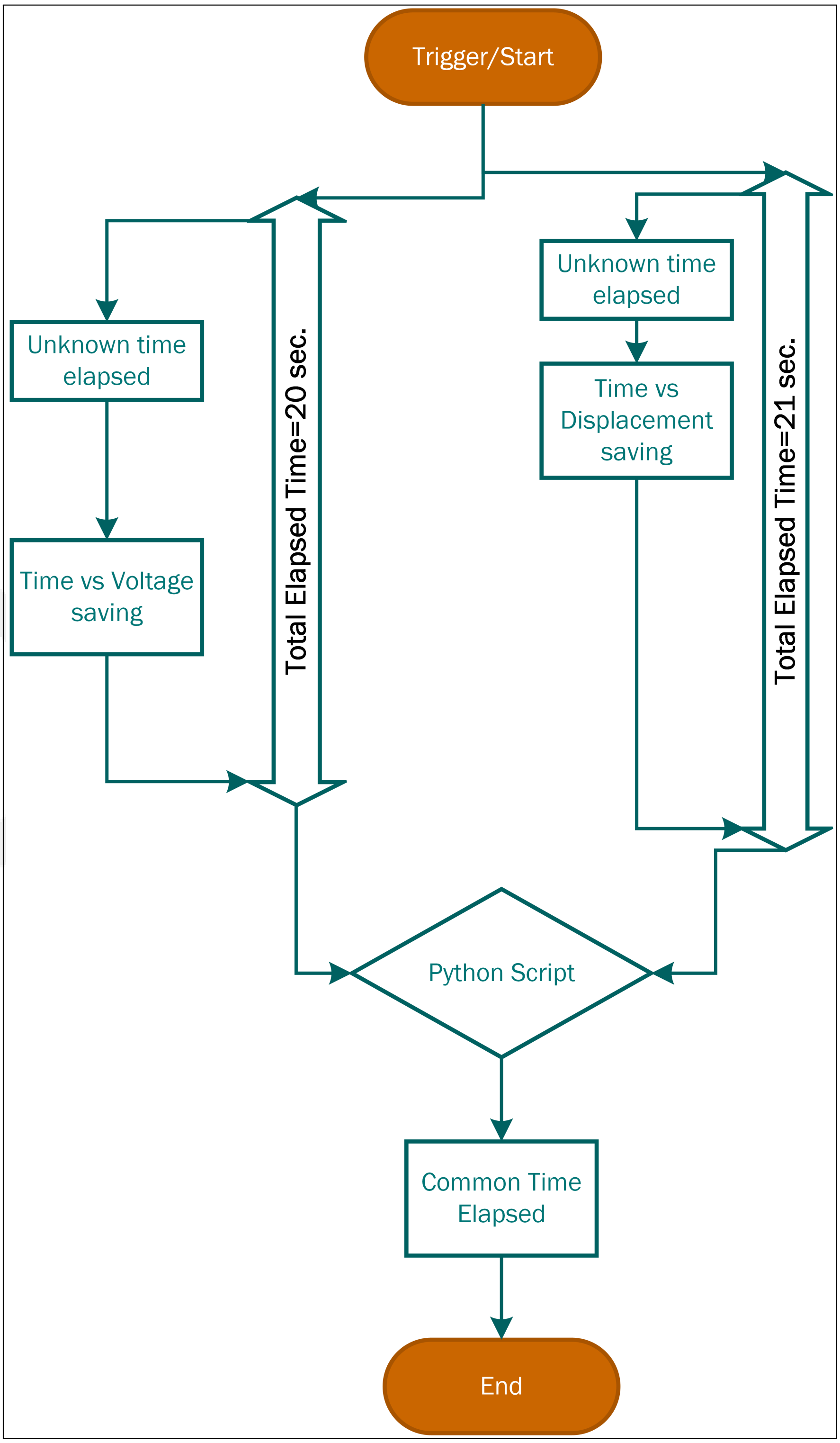

Figure 3.17. Sensor Control Flow Chart

4. EXPERIMENTS

This chapter is consists of two sections. In Section 4.1; procedure and details of experiments for static response is explained. In Section 4.2; procedure and details of experiments for frequency response is explained.

4.1. STATIC RESPONSE EXPERIMENTS

In static response experiments; after limiting the maximum travel of solenoid by nuts, voltage is applied in order to get its plunger to the highest position. While solenoid is hanging there stable, more weight is added to its hanger. The weight which it fails to carry is reported with voltage, current and maximum travel length values. The flowchart for static response experiment can be found in Figure 4.1.

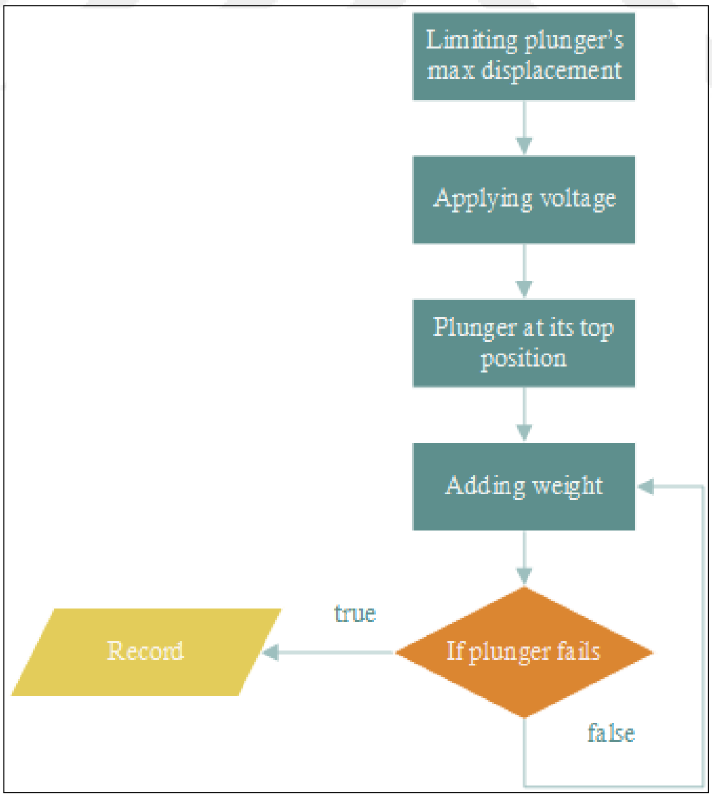

Figure 4.1. Static Test Flowchart

If we follow the flowchart; we see that it has six steps. They are limiting the plunger's maximum displacement, applying voltage, movement of the plunger to its top position, adding weight, checking its failure and recording. First step decides whereabouts of the plunger during the experiments. Second and third steps represents the moving of the plunger to that point. The fourth step is adding weight in order to make the plunger fail. This is checked and by fifth step. This step decides weather experiments should continue again from the fourth step or it should be recorded and end. Record process is represented in the last step.

This test is performed around 4.4 Watts applied to the solenoid itself and test is repeated for three times. This experiments results will be compared with force calculation at different travel distances by finite element analysis results.

Force vs displacement relationship of solenoid according to experiments can be seen on Figure 4.2. This figure is represented for visualizing purposes. It will be studied in discussion chapter along with finite element analysis results.

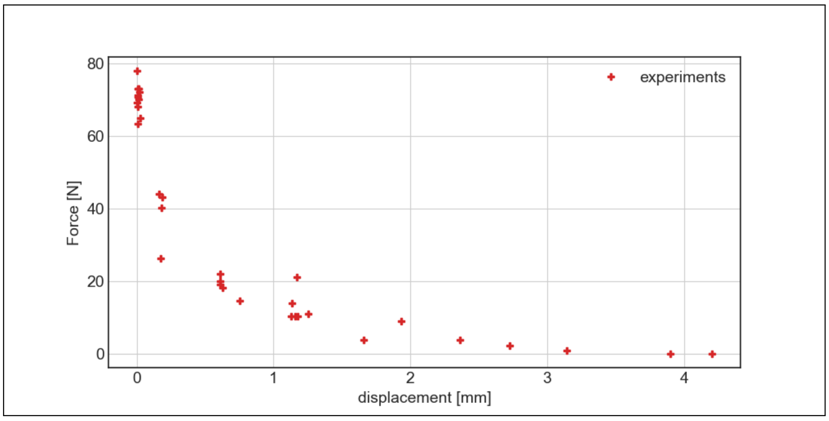

Figure 4.2. Static Experiment Results

4.2. FREQUENCY RESPONSE EXPERIMENTS

The flowchart for static response experiment can be found in Figure 4.3.

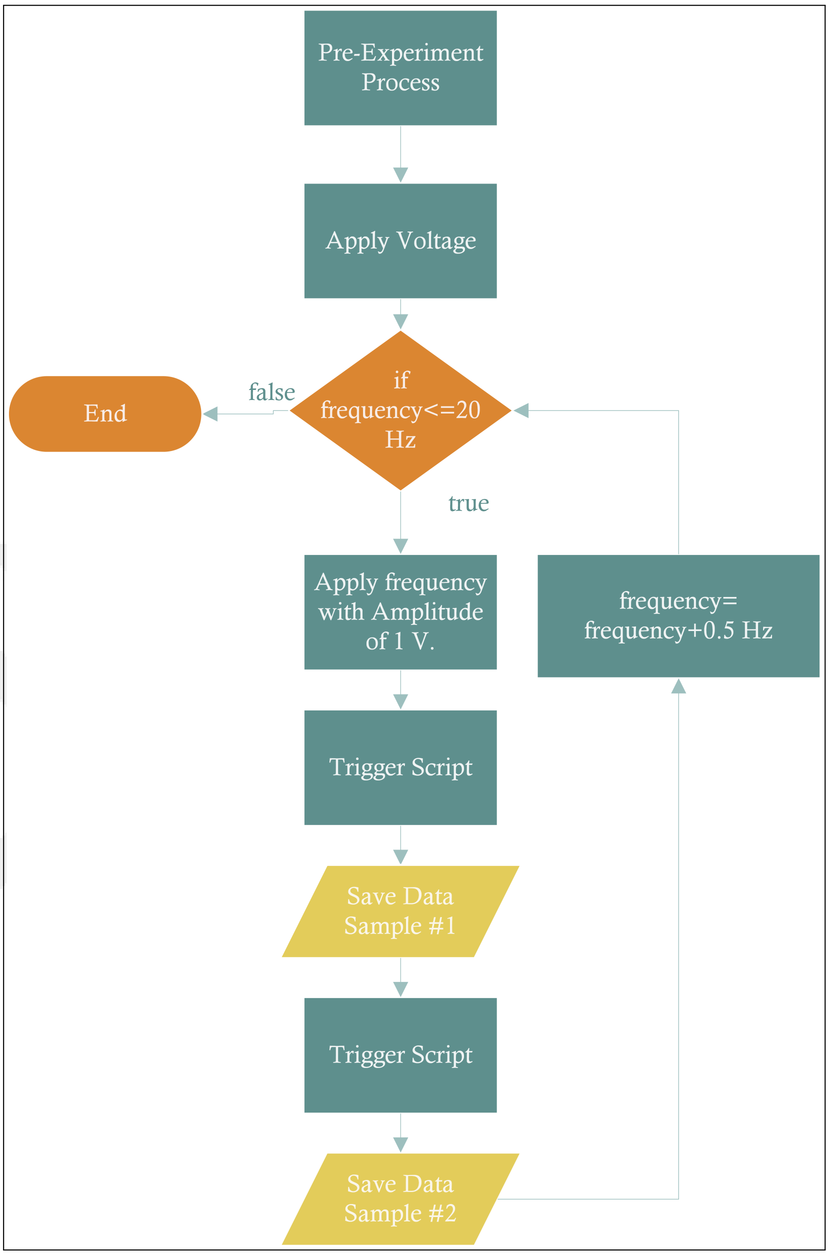

Figure 4.3. Frequency Test Flowchart

If we follow the flowchart, we see that it has nine steps. First two steps are done in order to get the solenoid up and running. Third step is the control step and it determines whether experiment is ongoing or finished. Forth step is changes the frequency of the signal. In order to store data twice at different times, fifth and seventh steps are performed. Data storage is represented right after those steps in the sixth and the eighth boxes. And then frequency value is refreshed in order to perform the experiment till the limit in our case 20 Hz.

In frequency response experiments, after selecting the optimal spring for the experiments by trial-and-error, the setup is assembled. Frequency response experiments starts with pre-experiment process which is making sure that displacement sensor reads zero displacement when the solenoid is at off mode and spraying oil to solenoid since the displacements are low and mechanical resistance can affect the results. After voltage is applied and the plunger of solenoid is reached its top position, frequency with an amplitude of 1 volt is applied and trigger function is ran for two times, which means two samples of data are stored at specific frequency. This test is performed between 0 and 20 Hertz. Also it should be noted that, at low frequencies like 500 mHz 1.5 Hz, data storing process delayed for a minute or two since transient effects are present. Those transient effects are causing plunger to not oscillate in a stable position. Instead of that, it was either decreasing or increasing to its stable position of oscillation.

Magnitude ratio vs frequency and phase difference vs frequency relationships of solenoid can be seen on Figure 4.4 and Figure 4.5. These figures are represented for visualizing purposes. They will be studied in discussion chapter along with simulation results.

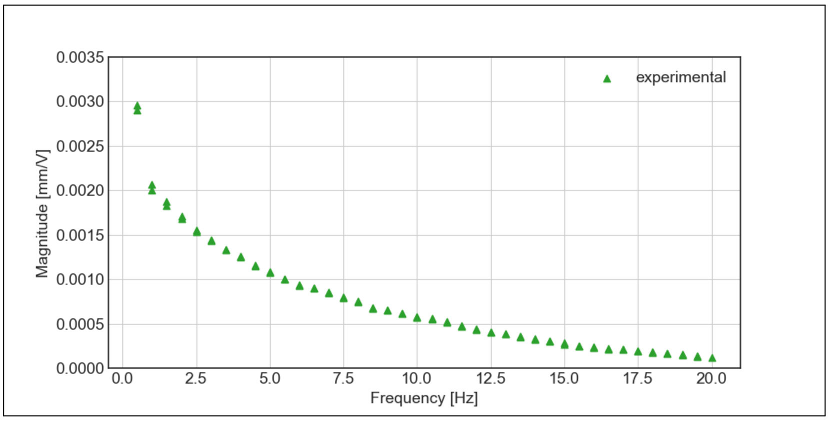

Figure 4.4. Experimented Magnitude Ratio vs Frequency

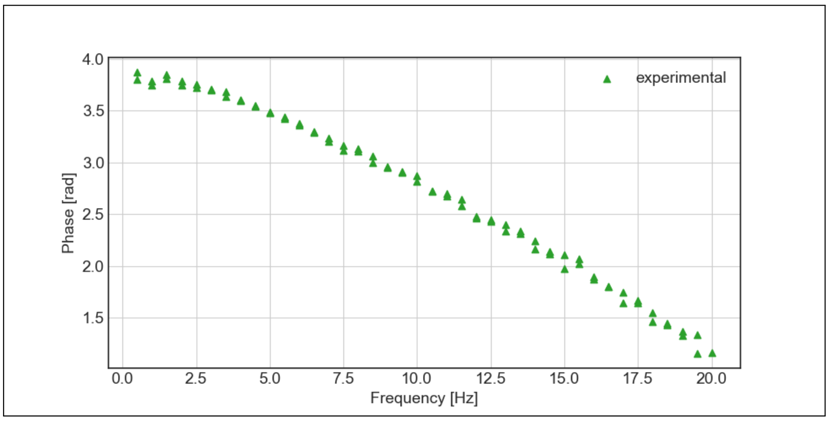

Figure 4.5. Experimented Magnitude Ratio vs Frequency

4.2.1. Example Data

In this subsection example data for frequency analysis experiemtes will be shown. Since the data collected from experiments constructed frequency result data, it is necessary to mention.

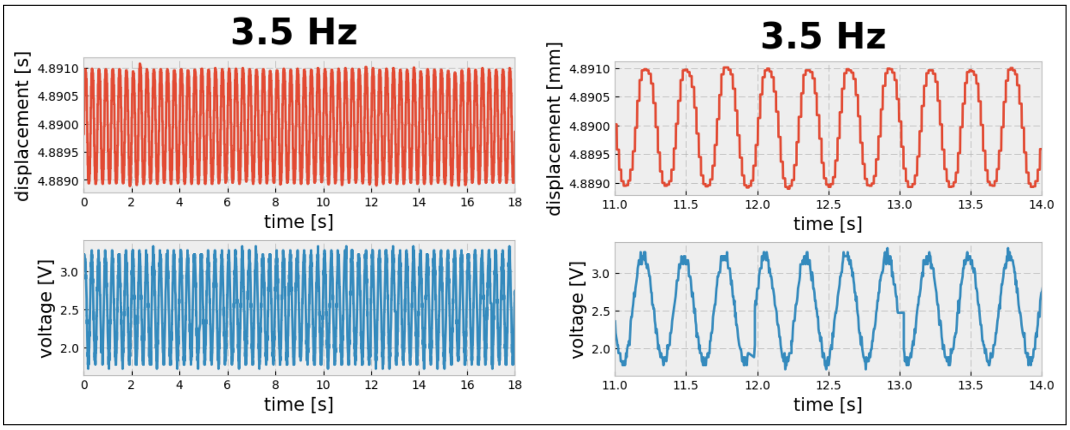

Figure 4.6. Over and Zoomed Views for 3.5 Hz

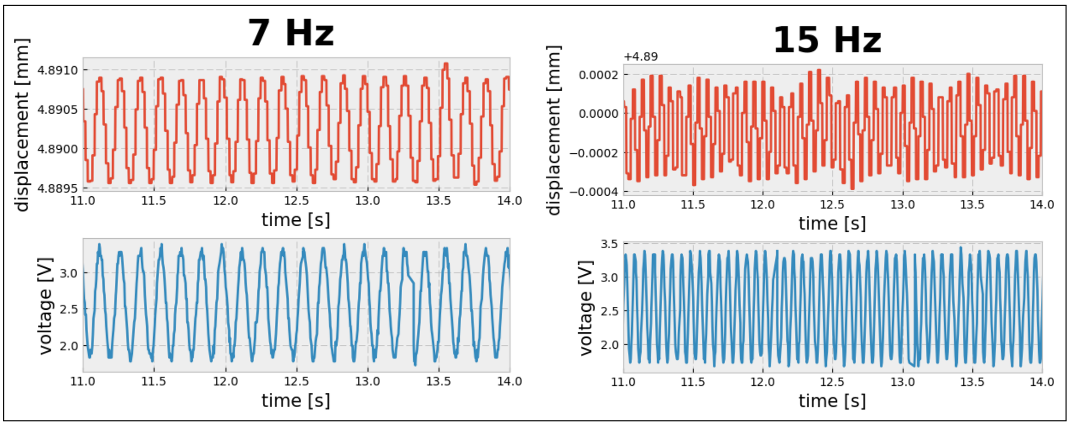

Figure 4.7. Over and Zoomed Views for 3.5 Hz

In Figure 4.6 and 4.7; some random picked, ordinary data are shown. Zoomed views are used for analysis purposes. Both voltage and displacement data narrowed down to common elapsed time.

In a regular data collection for the experiments; elapsed time is 17 to 18 seconds, and zoomed view has 3 to 4 seconds of common elapsed time. Some noise present in voltage data is eliminated by hand. For example; if zero crossings occur at 15,16 and 17th sample of data while second minimum difference is ; 15th and 17th of data are eliminated. It is possible to automate that process, however it is better to have a human touch in here, since over-looking to data can be consider as checking.

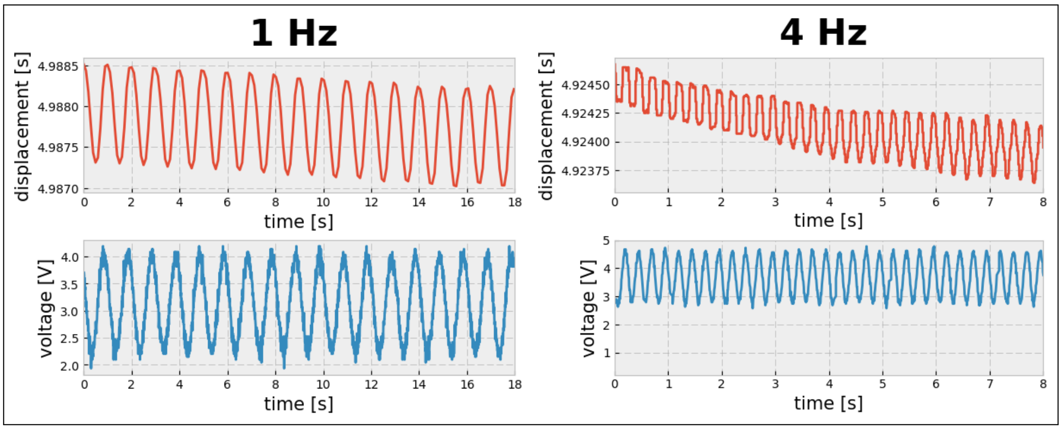

Figure 4.8. Example Data with Transient Effects

Also it should be mentioned that some detectable transient effects are present at low frequencies. Examples for that can be seen on Figure 4.8. In order to eliminate them, some time delay needs to be applied before data store.

5. RESULTS AND DISCUSSION

This chapter consists of two sections. In Section 5.1; static response results for simulation and experiments are presented and discussed. In Section 5.2; frequency response results for simulation and experiments are presented and discussed.

5.1. STATIC RESPONSE RESULTS

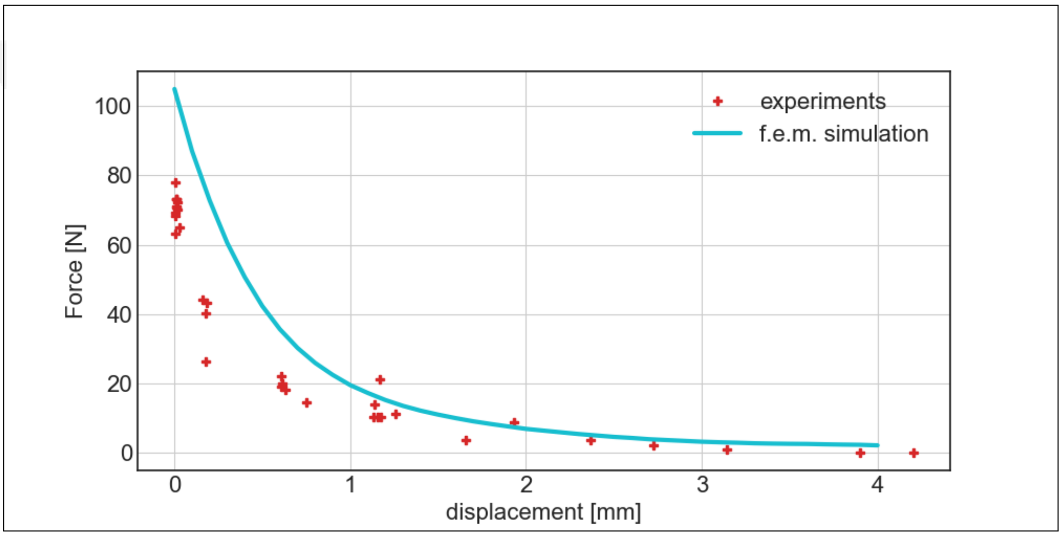

Figure 5.1. Static Response Analysis

In Figure 5.1; experimental results are compared with finite element method results for force applied by solenoid's plunger, along its travel.

It is seen that they both show the same behaviour as plunger goes further away. The reason behind the unevenly distribution of experiments is the lack of mechanism for moving the plunger.

In discussion for static response results; even though there is difference between simulation and experiment output, the results can be considered as expected. The force generated by the solenoid declines dramatically as travelled distance of solenoid goes higher. This is common sense since it gets further away from its core.

The reason behind the difference of force calculated and measured is the lack of knowledge of solenoid's inside geometry and material. The model used in finite element method is based on assumptions on those parameters.

5.2. FREQUENCY RESPONSE RESULTS

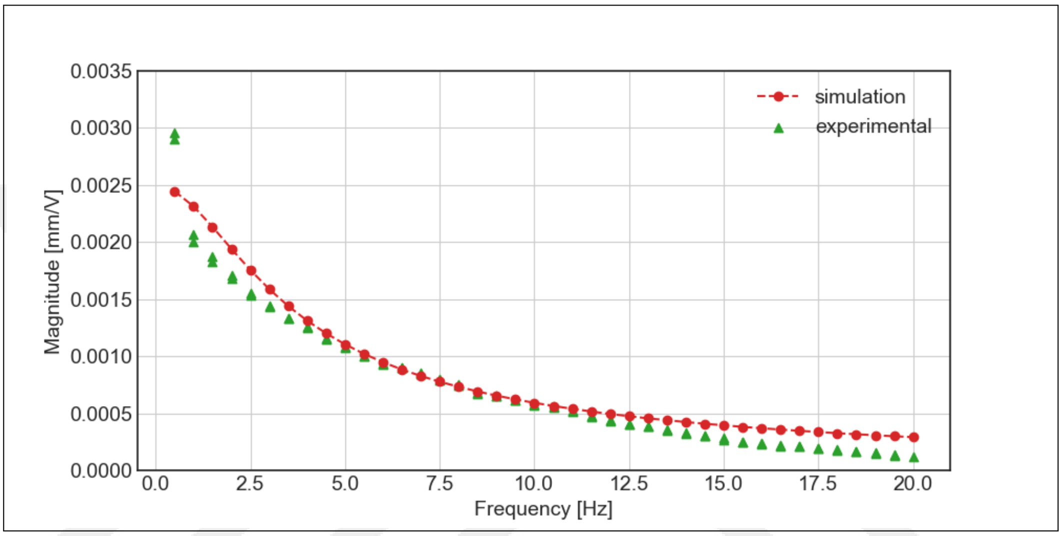

Figure 5.2. Simulation vs Experiment for Magnitude Ratio

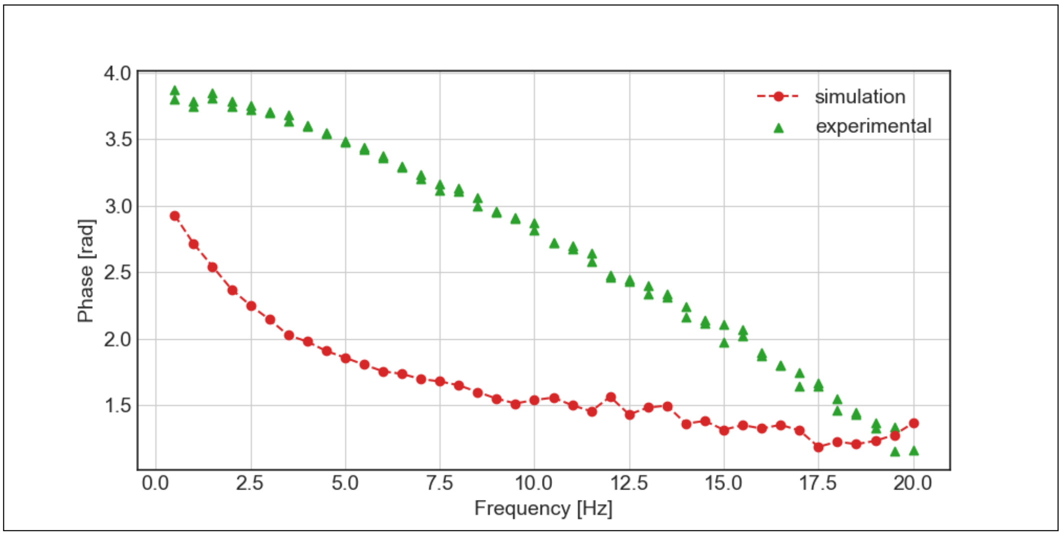

Figure 5.3. Simulation vs Experiment for Phase Difference

In Figure 5.2; comparison of simulation response for frequency and experiment results in terms of magnitude ratio can be seen. Also the comparison in terms of phase can be found on Figure 5.3.

When we look at the Figure 5.2; we see simulation results and experiment results validate each other. They share similar values from the lowest to the highest frequencies.

In Figure 5.3; even though there is a difference between experimental and simulation values, it is seen that they both show decline behaviour and match at similar phase values at high frequencies.

In frequency response it is seen that there is a match between experiment and simulation results, specially in terms of magnitude ratio results. According to solenoid's governing equations; solenoids are second order systems. The natural frequency should be at 13.82 Hz, however this is not visible in both Figure 5.2 and 5.3. The reason for that is damping constant is relatively high and prevents plunger to have a great magnitude peak at its resonance. Also it should be noted that damping constants value is determined by tuning since there was not any experiment for measuring it.

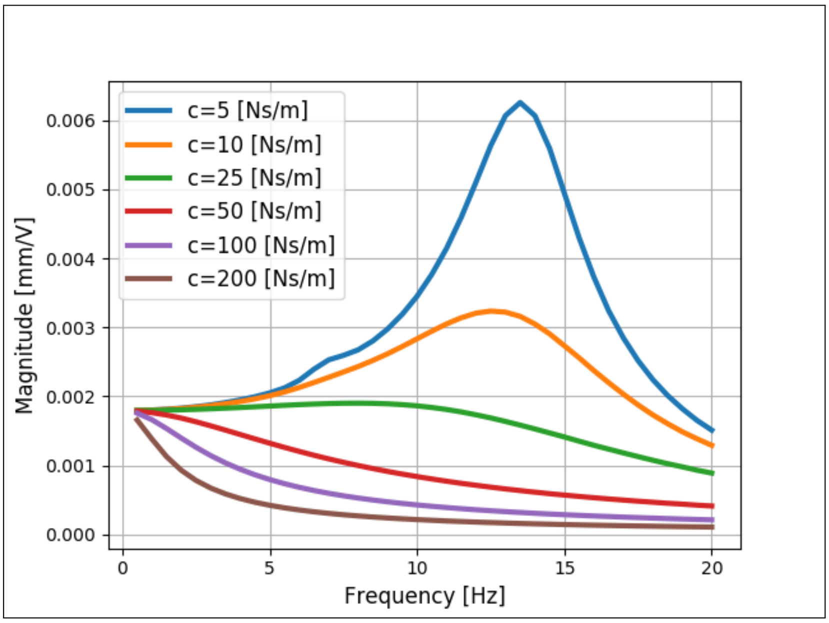

Figure 5.4. Simulation with Different Damping Coefficients

In Figure 5.4; simulation of magnitude ratio with different damping coefficients are shown. The purpose of this is to validate the points made before. Therefore in this graph we can see the second order characteristics, peak near natural frequency. Also fitted damping coefficient causes overdamped system since $\zeta = 2.88 > 1$.

In term of phase difference; simulation and experimental results show similarity. They both decline and share similar values at high frequencies. However when the whole range is considered, there is significant difference between simulation and experiment results. This can be explained with the difference of experiment and simulation. While in simulation everything is defined ideal. Which means system responses to input immediately.

However in reality, things are not as smooth. In order to pull the plunger, solenoid must be magnetized through passing current in coil, and also plunger can go to its lower position after solenoid is fully demagnetized. This takes some not negligible time. These principals also valid reasons for phase difference too. Presence of transient effects are highly likely to cause the difference.

6. CONCLUSION

In this thesis linear, short stroke solenoid is studied as a proportional actuator.

Solenoid is modelled for finite element analysis. Its force output is calculated for a range of plunger displacement. Also a mathematical model of solenoid is constructed and simulated. This model consisted of two subsystems, which are electrical and mechanical. A simulation is run for a range of frequencies in order to simulate solenoid's frequency response.

Static and frequency tests are performed on the solenoid. In order to achieve this an experiment setup is designed and manufactured from scratch. For frequency tests; a spring is selected, a circuit which can deliver AC+DC signal is designed and attached to the setup, software-wise control for independent sensors is performed.

Experiment results are compared with the relative model results. Static tests vs. finite element analysis results yield matching results as well as the magnitude ratio of frequency tests vs. magnitude ratio of simulation results.

However it should be noted that frequency model can be improved by considering transient effects.

It is seen that solenoid are capable of working in a range rather than just binary "on-off" mode.

REFERENCES

1. Postrioti L, Malaguti S, Bosi M, Buitoni G, Piccinini S, Bagli G. Experimental and numerical characterization of a direct solenoid actuation injector for Diesel engine applications. Fuel. 2014;118:316328

2. Wallace H. Solenoid locked push door. Google Patents; 1946. US Patent 2,411,266. Available from: https://www.google.com.tr/patents/US2411266.

3. Mansour H. Design and development of active and semi-active engine mounts. Applied Science: School of Engineering Science; 2010.

4. Cheung N, Lim K, Rahman M. Modelling a Linear and Limited Travel Solenoid. In: Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON93., International Conference on. IEEE; 1993. p. 15671572

5. Lim K, Cheung N, Rahman M. Proportional Control of a Solenoid Actuator. In: Industrial Electronics, Control and Instrumentation, 1994. IECON94., 20th International Conference on. vol. 3. IEEE; 1994. p. 20452050.

6. Vaughan N, Gamble J. The modelling and simulation of a proportional solenoid valve. American Society of Mechanical Engineers; 1990.

7. Femminfo. [Online]. Available from: http://www.femm.info/wiki/HomePage [Accessed 10 June 2017].

8. Meeker D. Finite Element Method Magnetics: OctaveFEMM.; 2010.

9. Numpyorg. [Online]. Available from: http://www.numpy.org/ [Accessed 10 June 2017].

10. Scipyorg. [Online]. Available from: https://www.scipy.org/ [Accessed 10 June 2017].

APPENDIX A: F.E.A. CONTROL OCTAVE CODE

```
openfemm
opendocument('modelov4.FEM');
mi_saveas('temp2cc.fem');
h=.1;a=1;b=1;ak=1.4

mi_modifycircprop('circuit',1,ak);

for i=0:h:4
mi_clearselected
mi_purgemesh
mi_createmesh
mi_analyze
mi_loadsolution
mo_groupselectblock(1)
F(a,1)=mo_blockintegral(19)
a=a+1

mi_clearselected
mi_selectgroup(1)
mi_movetranslate(0,h)
dist(b)=i;
b=b+1;
end

closefemm
```

APPENDIX B: SIMULATION CONTROL MATLAB CODE

```
% simumaster
copyfile SolSiModel_v5_manuel.slx SolSiModel_Temp.slx
ad='0713'
adf=strcat('OK_simu_',ad,'.txt');
fid=fopen(adf,'w');
fprintf(fid, [ '# Frekans[hz] ;' 'TF mag;' 'fazRAD'  '\n
    ']);
o=1;

model = 'SolSiModel_Temp';
load_system(model);

for frekans=.5:.5:20


    sfrekans=num2str(frekans);


    set_param('SolSiModel_Temp/SignalGen','Frequency',
        sfrekans);
    save_system(model,[],'OverwriteIfChangedOnDisk',true);
    sim(model);

    voltaj=ScopeData1.signals(1).values;
    deplasman=ScopeData1.signals(2).values;



    voltaj_gcc=voltaj-mean(voltaj);
    deplasman_gcc=deplasman-mean(deplasman);
```

```
disp2=test;
volt2=test;

volt2.signals.values=voltaj_gcc;
volt2.time=ScopeData1.time;

disp2.time=ScopeData1.time;
disp2.signals.values=deplasman_gcc;

x=deplasman_gcc;
dispX = find(abs(diff(sign(x)))==2);

x=voltaj_gcc;
voltX = find(abs(diff(sign(x)))==2);
q=length(dispX);
e=length(voltX);
a=min([q e]);

t=voltw.time(12999:end);
V=voltw.signals.values(12999:end);
D=dispw.signals.values(12999:end);

DMF = find(D==max(D),1,'last');
VMF= find(V==max(V),1,'last');

tersfaz=t(VMF)-t(DMF);
fazRAD=mod((tersfaz*2*pi*frekans),(2*pi));

for i=3:a
```

```
    xmaX(i-2)=max(deplasman(dispX(i-2):dispX(i)));
    xmiN(i-2)=min(deplasman(dispX(i-2):dispX(i)));
end

magX=abs(xmaX-xmiN);
meanMagX=mean(magX);

for i=3:a
    vmaX(i-2)=max(voltaj(voltX(i-2):voltX(i)));
    vmiN(i-2)=min(voltaj(voltX(i-2):voltX(i)));
end

magV=abs(vmaX-vmiN);
meanMagV=mean(magV);

meanMag=meanMagX/meanMagV;

fid=fopen(adf,'a+');

fprintf(fid, '%s\t;',sfrekans);
fprintf(fid, '%d;',meanMag);
fprintf(fid, '%d',fazRAD);
fprintf(fid, '\n');

fclose(fid);
frekans

fazMAT(o,1)=fazRAD;
magMAT(o,1)=meanMag;
freMAT(o,1)=frekans;
```

```
    o=o+1;

end

save_system(model,[],'OverwriteIfChangedOnDisk',true);
close_system(model)

close all



delete SolSiModel_Temp.slx
delete SolSiModel_Temp.slx.autosave

subplot(2,1,1);
plot(freMAT,magMAT);
ylabel('Mag');
subplot(2,1,2);
plot(freMAT,fazMAT);
ylabel('Phase');
```

APPENDIX C: FREQUENCY ANALYSIS PYTHON CODE

```
from __future__ import division
from matplotlib import pyplot as plt
from matplotlib import style
import numpy as np
import os
import sys
from matplotlib import interactive
import matplotlib.pylab as pylab
import pandas as pd
from shutil import copy2
from IPython import get_ipython

isimraw='0603_183411_7HZ'
isim=isimraw[:11]

def isimdegistir():
  os.rename(isim+'.csv',isimyeni+'.csv')
  os.rename(isim+'.log',isimyeni+'.log')

def kapat():
  plt.close('all')

def kapan():
  plt.close('all')

def isimdegistir2():
  os.rename(isim+'.log',isimyeni+'.log')
```

```
os.chdir('C:\\Users\\cc\\Desktop\\DData')

frekans=isimraw[12:-2]
HZ='_'+frekans+'HZ'
isimyeni=isim+HZ

isimdegistir2()
del HZ, isim

#%%
# SENSOR GLOBAL TIME



df = pd.read_csv(isimyeni+'.csv', delimiter=";",skiprows
    =1,names=['onemsiz1','GT','onemsiz2'])
sGT=list(df.GT)

for i in range(0,len(df.GT)):
   q=df.GT[i]
   q=q[11:]
   q=q.replace("EEST(+0300)","")
   q=q.replace(":","")
   qsaat=q[0:2]
   qdkk=q[2:4]
   qsny=q[4:]
   sGT[i]=float(qsny)+60*float(qdkk)+60*60*float(qsaat)



sGT=np.asarray(sGT)
x_time,x = np.loadtxt(isimyeni+'.csv',unpack=True,skiprows
```

```
    =1,usecols=(0,2),delimiter= ';')

del q,qsaat,qdkk,qsny
del df
del i



# ARDUINO GLOBAL TIME

df2 = pd.read_csv(isimyeni+'.log', delimiter=";",skiprows
    =1,names=['GT','vlt'])

vlt=df2.vlt.values
aGT=list(df2.GT)

for i in range(0,len(df2.GT)):
  q=str(df2.GT[i])
  qsaat=q[0:2]
  qdkk=q[2:4]
  qsny=q[4:]
  aGT[i]=float(qsny)+60*float(qdkk)+60*60*float(qsaat)

aGT=np.asarray(aGT)

print (len(aGT))
print (len(vlt))

del df2,i
del q,qsaat,qdkk,qsny
print ("checkpoint charlie")
```

```
#%%
### FULL RANGE ESIT ARALIK BULMA

kd=-1
ku=-1

while kd==-1:
  for i in range(0,len(sGT)):
    vsg=round(sGT[i],3)
    asg=round(aGT[0],3)
    fark=(abs((asg-vsg)))
    if fark==0:
      kd=i
      print("Kesme Alt Degeri")
      print(i)
  if kd==-1:
    aGT=aGT[1:]
    vlt=vlt[1:]



while ku==-1:
  for i in range(0,len(sGT)):
    vsg=(sGT[i])
    asg=(aGT[-1])
    fark=(abs((asg-vsg)))
    if fark==0:
      ku=i
      print("Kesme Ust Degeri")
      print(i)
```

```
    if ku==-1:
        aGT=aGT[:-1]
        v1t=v1t[:-1]
#%%

sGT=sGT[kd:ku+1]
x=x[kd:ku+1]
x_time=x_time[kd:ku+1]

print ([min(sGT), min(aGT), max(sGT), max(aGT)])
if [0,0]==([min(sGT)-min(aGT), max(sGT)-max(aGT)]):
    ok1=True
    print ("ok1")

# %%

zamanA=aGT-max(aGT)
zamanS=sGT-max(sGT)
zamanA=zamanA+abs(min(zamanA))
zamanS=zamanS+abs(min(zamanS))
V=v1t*(11.)*(5./1023.)
v1t=V
X=abs(x)
kapan()

kirmizi='#e24a33ff'
mavi='#348abdff'

print ("____VNIMANIE____")
```

```
#%%
# FULL RANGE

style.use('bmh')

fig1 = plt.figure()
ax1 = fig1.add_subplot(2,1,1)
ax2 = fig1.add_subplot(2,1,2)

ax1.grid(True)
ax2.grid(True)

ff=15

ax1.set_xlim([round(min(zamanS)), round(max(zamanS))])
ax2.set_xlim([round(min(zamanA)), round(max(zamanA))])

ax1.set_xlabel('time [s]',fontsize=ff)
ax2.set_ylabel('voltage [V]',fontsize=ff)
ax2.set_xlabel('time [s]',fontsize=ff)
ax1.set_ylabel('displacement [s]',fontsize=ff)

ax1.plot(zamanS,X,kirmizi)
ax2.plot(zamanA,V,mavi)

fig1.suptitle(frekans+' Hz', fontsize=30, fontweight='bold
    ')
plt.subplots_adjust(left=.12, bottom=.11, right=.90, top
    =.88, wspace=.12, hspace=.355)
```

```
# ZOOMED

kd2=-1
ku2=-1

limit_alt=10
limit_ust=14

for i in range(0,len(zamanA)):

    if zamanA[i]>limit_alt:
        if zamanA[i]<limit_alt+1:
            vtmin=i
    if zamanA[i]>limit_ust:
        if zamanA[i]<limit_ust+1:
            vtmax=i



sGT2=zamanS
aGT2=zamanA[vtmin:vtmax]
vlt2=vlt[vtmin:vtmax]

while kd2==-1:
  for i in range(0,len(sGT2)):
    vsg=round(sGT2[i],3)
    asg=round(aGT2[0],3)
    fark=(abs((asg-vsg)))
    if fark==0:
      kd2=i
```

```
        print("Kesme Alt Degeri")
        print(i)
    if kd2==-1:
      aGT2=aGT2[1:]
      vlt2=vlt2[1:]



while ku2==-1:
    for i in range(0,len(sGT2)):
      vsg=round(sGT2[i],3)
      asg=round(aGT2[-1],3)
      fark=(abs((asg-vsg)))
      if fark==0:
        ku2=i
        print("Kesme Ust Degeri")
        print(i)
    if ku2==-1:
      aGT2=aGT2[:-1]
      vlt2=vlt2[:-1]



sGT2=sGT2[kd2:ku2+1]
x2=x[kd2:ku2+1]



print ([min(sGT2), min(aGT2), max(sGT2), max(aGT2)])
if [0,0]==([min(sGT2)-min(aGT2), max(sGT2)-max(aGT2)]):
    ok7=True
    print ("ok7--zommed aral k")
```

```
  # %%
## ZOOMED

Xzum=abs(x2)
zamanSzum=sGT2
zamanAzum=aGT2
Vzum=vlt2

fig2 = plt.figure()
ax1 = fig2.add_subplot(2,1,1)
ax2 = fig2.add_subplot(2,1,2)

ax1.grid(True)
ax2.grid(True)

ff=15
ax2.set_xlabel('time [s]',fontsize=ff)
ax2.set_ylabel('voltage [V]',fontsize=ff)

ax1.set_xlim([np.ceil(min(zamanSzum)), np.floor(max(
   zamanSzum))])
ax2.set_xlim([np.ceil(min(zamanAzum)), np.floor(max(
   zamanAzum))])

ax1.set_xlabel('time [s]',fontsize=ff)
ax1.set_ylabel('displacement [mm]',fontsize=ff)

ax1.plot(zamanSzum,Xzum,kirmizi)
ax2.plot(zamanAzum,Vzum,mavi)
fig2.suptitle(frekans+' Hz', fontsize=30, fontweight='bold
```

```
    ')
plt.subplots_adjust(left=.12, bottom=.11, right=.90, top
    =.88, wspace=.12, hspace=.355)
plt.show



def kaydet():
  isimyeni1='OK_'+isimyeni
  isimyeni_tmp=isimyeni1+'_full'
  fig1.savefig('OK/'+isimyeni_tmp+'.png', bbox_inches='
      tight')
  isimyeni_tmp=isimyeni1+'_zoomed'
  fig2.savefig('OK/'+isimyeni_tmp+'.png', bbox_inches='
      tight')
  copy2(isimyeni+'.csv', 'OK/'+isimyeni1+'.csv')
  copy2(isimyeni+'.log', 'OK/'+isimyeni1+'.log')
  os.rename(isimyeni+'.csv','@'+isimyeni+'.csv')
  os.rename(isimyeni+'.log','@'+isimyeni+'.log')
  print ('\nKaydedildi!')

def sil():
  os.rename(isimyeni+'.csv', 'silinenler/'+isimyeni+'.csv
      ')
  os.rename(isimyeni+'.log', 'silinenler/'+isimyeni+'.log
      ')
  print ('\nlog ve csv dosyalari silinenler klasorune
      tasindi.')
  get_ipython().magic('reset')

#%%
```

```
# ZERO CROSSINGLERI BULMA ZOOOMED

V_0=Vzum-np.mean(Vzum)
zero_xingsV = np.where(np.diff(np.sign(V_0)))[0]

X_0=Xzum-np.mean(Xzum)
zero_xingsX= np.where(np.diff(np.sign(X_0)))[0]

#%%
tzc_x=[]
for i in range(0,len(zero_xingsX)):
    tzc_x.append(zamanSzum[zero_xingsX[i]])

tzc_x=np.array(tzc_x)

tzc_v=[]
for i in range(0,len(zero_xingsV)):
    tzc_v.append(zamanAzum[zero_xingsV[i]])

tzc_v=np.array(tzc_v)

if len(tzc_x)==len(tzc_v):
  print "e it tzc_v ve tzc_v"
  faz=tzc_x-tzc_v
else:
  temporar=[]
  nerdebir=[]
  print "kontrol et tzc_v ve tazc_x"
  for i in range(0,len(zero_xingsV)-1):
    temporar.append((zero_xingsV[i+1]-zero_xingsV[i]))
```

```
   if abs(temporar[i])==1:
     if (i) not in nerdebir:
       nerdebir.append(i)
     if (i+1) not in nerdebir:
       nerdebir.append(i+1)

print("min fark zero xing")
print(min(temporar))
print([len(tzc_x),len(tzc_v)])

raw_input("kontrol et")
def silulen(*args, **kwargs):
 a=[]
 for ar in args:
    a.append(ar)
 print a
 global zero_xingsV;
 global temporar
 global nerdebir
 zero_xingsV=np.delete(zero_xingsV,a)
def dvy():
 global tzc_x
 global tzc_v
 global ortak
 global faz
 ortak=min(len(tzc_x),len(tzc_v))
 faz=tzc_x[:ortak]-tzc_v[:ortak]

testDur()
print('testDur')
```

```
#%%## SIL FONKSIYONU

def silfonk(*args, **kwargs):
    a=[]
    for ar in args:
        a.append(ar)
    print a
    global zero_xingsV;
    global temporar
    global nerdebir
    zero_xingsV=np.delete(zero_xingsV,a)

#%%

meanfaz=np.mean(faz)
faz_aciD=(meanfaz*float(frekans)*360)%360
faz_aciRAD=(meanfaz*float(frekans)*2*np.pi)%(2*np.pi)

if len(zero_xingsV)-len(zero_xingsX)==0:
   print('hurraaa (2)')
   ok2=True
   print("hurraa2")
else:
   ok2=False



#%%
## ZERO CROSS MAX MIN VE MAGNITUDE
```

```
XzumL=np.ndarray.tolist(Xzum)
xmaX=[]
xmiN=[]
for i in range(2,len(zero_xingsX)):
    xmaX.append(max(XzumL[zero_xingsX[i-2]:zero_xingsX[i
        ]]))
    xmiN.append(min(XzumL[zero_xingsX[i-2]:zero_xingsX[i
        ]]))

xmaX=np.array(xmaX)
xmiN=np.array(xmiN)
magX=abs(xmaX-xmiN)
meanMagX=np.mean(magX)


VzumL=np.ndarray.tolist(Vzum)
vmaX=[]
vmiN=[]

for i in range(2,len(zero_xingsV)):
    vmaX.append(max(VzumL[zero_xingsV[i-2]:zero_xingsV[i
        ]]))
    vmiN.append(min(VzumL[zero_xingsV[i-2]:zero_xingsV[i
        ]]))

vmaX=np.array(vmaX)
vmiN=np.array(vmiN)
magV=abs(vmaX-vmiN)
meanMagV=np.mean(magV)
meanMagTF=meanMagX/meanMagV
```

```
def akaydetcsv():
  with open('OK/'+'OK_csv_'+isimyeni+'.dat', "w") as
     text_file:
    text_file.write(" Data Name          ;\
                     Frequency [Hz];\
                      Mean Mag. of TF [mm/V];\
                       Mean Mag. of Phase [rad] \
                     \n")
    text_file.write(" %s;" % (isimyeni))
    text_file.write(" %s;" % (frekans))
    text_file.write(" %s;" % (meanMagTF))
    text_file.write(" %s" % (faz_aciRAD))

    print('\nALL GOOD csv')

def akaydet():
  with open('OK/'+'OK_'+isimyeni+'.dat', "w") as text_file
     :
    text_file.write("#Data Name\n")
    text_file.write("isimyeni=%s  \n \n" % (isimyeni))

    text_file.write("# Frequency [Hz]\n")
    text_file.write("frekans=%s  \n \n" % (frekans))

    text_file.write("# Mean Mag. of Voltage [V]\n")
    text_file.write("meanMagV=%f  \n \n" % (meanMagV))

    text_file.write("# Mean Mag. of Displacement [mm]\n")
    text_file.write("meanMagX=%f \n \n" % (meanMagX))
```

```
    text_file.write("# Mean Mag. of Transfer Func. [mm/V]"
        )
    text_file.write("meanMagTF=%f \n \n" % (meanMagTF))

    text_file.write("# Mean Phase [s] \n" )
    text_file.write("meanfaz=%f  \n \n" % (meanfaz))

    text_file.write("# Phase Degree [*] \n" )
    text_file.write("faz_aciD=%f \n \n" % (faz_aciD))

    text_file.write("# Phase Degree [rad] \n" )
    text_file.write("faz_aciRAD=%f \n \n" % (faz_aciRAD))

    print('\nALL GOOD')


def apy():

    copy2('OK/'+'OK_'+isimyeni+'.dat', 'OK/pyimport/'+
       isimyeni+'.py')
    print("py dosyalari hazir.")

def pp():
    kaydet()
    akaydetcsv()
    kapat()
    get_ipython().magic('reset')

if ok1*ok2*ok7 is 1:
```

```
    print ("\n OK1OK2OK7")


    print("pp")
else:
    print ("ss")
```