A NEW TMR BASED SENSING TECHNIQUE FOR ELECTRIC GUITAR PICKUP

A THESIS

submitted by

DEBJYOTI KUMAR MANDAL
(EE15M051)

for the award of the degree
of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2017

THESIS CERTIFICATE

This is to certify that the thesis titled "**A NEW TMR BASED SENSING TECHNIQUE FOR ELECTRIC GUITAR PICKUP**", submitted by **Mr. Debjyoti Kumar Mandal**, to the Indian Institute of Technology Madras, Chennai for the award of the degree of **Master of Technology**, is a bona fide record of research work done by him under my supervision. The contents of this thesis, in full or a part has not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Boby George
Research Guide
Associate Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude towards several people who enabled me to reach this far with their timely guidance, support and motivation. First and foremost, I offer my earnest gratitude to my guide Dr. Boby George, Associate Professor, Department of Electrical Engineering, IIT Madras, for his support and guidance has inspired me to work efficiently throughout the project, and I thank him for being friendly as well as helping me with all the obstacles I faced in the project and allowing me the freedom and flexibility while working on the project. He has been a constant source of encouragement and motivation and I consider myself fortunate to work under his guidance.

I would also like to thank Dr. V Jagadeesh Kumar, Head, Measurement and Instrumentation Laboratory, IIT Madras, for his valuable guidance and suggestions throughout the course of the work done for this project.

I would like to thank Mr. B Umaithanu Pillai, Technical Superintendent, and Mrs. V Rekha, Junior Technical Superintendent, from the Measurement and Instrumentation Laboratory, IIT Madras for their support and assistance with the logistics throughout the course of the project. I would also like to thank the staff at the Central Electronics Centre, IIT Madras, for their kind-hearted help they rendered at various stages of the project.

I must thank all my lab mates, batch mates and friends who were very supportive to me and encouraged me constantly. I would especially like to thank A S Ramkumar, Anil Kumar, Ashwani Kumar, Dinesh Reddy, Shashank Anand and everyone in the Measurements and Instrument Lab for always being there for me, making lab a fun place to be in, adding loads of laughter to it, each day, making it all the more difficult for me to leave this place. Special thanks to Mr. Saikat Jana, PhD scholar, whose guidance and suggestions have been very valuable.

Last but not the least, I would like to thank my parents for their continuous support and care without which I would not have been able to get through my life in IIT Madras.

ABSTRACT

KEYWORDS: TUNNEL MAGNETORESISTANCE, ELECTRIC GUITAR, PICKUP

This thesis deals with the development of a completely new type of electric guitar pickup technology based on Tunnel Magnetoresistive sensor. The method involves passing current of a particular frequency through the guitar string, thereby generating magnetic field which is sensed by the Tunnel Magnetoresistive sensor producing a voltage output. As the string is vibrated, the output from the TMR sensor based pickup circuit produces an amplitude modulated waveform after which signal processing algorithms are applied producing accurate output waveform which replicates the original sound of the vibrating string.

In the hardware implementation of the project, the TMR sensor was placed right below the test string and the output from the prototype circuit was connected to the Elvis board. The signals were acquired using a Virtual Instrument using LabVIEW and the waveforms were transferred to MATLAB for signal processing. Here the envelope was extracted from the amplitude modulated signal from the prototype circuit and was compared with a Look-Up Table which finally produced the waveform corresponding to the vibration of the test string.

The results obtained from the prototype system developed has proven that there is a huge scope and potential in the music industry by using the TMR sensor for electric guitar pickup. The prototype circuit which was designed around the TMR sensor can be easily built and the signal processing algorithms used in the project is also not computationally tough to implement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

TMR	TUNNEL MAGNETORESISTANCE
GMR	GIANT MAGNETORESISTANCE
PCB	PRINTED CIRCUIT BOARD
LUT	LOOK UP TABLE
LDS	LASER DISTANCE SENSOR
DAQ	DATA ACQUISITION
AM	AMPLITUDE MODULATION
SSE	SUM OF SQUARES DUE TO ERROR
RMSE	ROOT MEAN SQUARED ERROR
TI	TEXAS INSTRUMENTS
NI	NATIONAL INSTRUMENTS
FFT	FAST FOURIER TRANSFORM

CHAPTER 1

1.1 INTRODUCTION

The electric guitar is one of the popular instruments of the last half-century in the music industry. Certainly, its introduction brought a major change to musical technology and has shaped the sound and direction of modern musical styles and revolutionized the music industry. Before the invention of electric guitar in the 1930's, people have been playing acoustic guitars for thousands of years, for which such instruments depended on the acoustic property to create the required sound.

Today majority of the electric guitar manufacturers use inductive pickups. A pickup is a transducer that captures or senses mechanical vibrations produced by stringed instruments such as guitar and converts them to an electrical signal. Because of their natural inductive qualities, all magnetic pickups tend to pick up ambient, usually unwanted electromagnetic interference or EMI, resulting in hum. Another disadvantage of inductive pickups is that they very bulky.

In the recent years, developments have been going around to create active guitar pickups based on adaptive inductive coins, Hall Effect and GMR. This project aims to create a new technology for guitar pickup using TMR sensors. The discovery of the TMR effect was originally discovered in 1975 by M. Jullière. TMR sensor works on the principle of spin-dependent tunnelling, which is the tendency of a material to change the value of its electrical resistance when externally applied magnetic field.

1.2 TMR SENSOR

The TMR sensor used in this work is AAT001-10E TMR Angle Sensor. The AAT001-10E angle sensor is a unique array of four Tunnelling Magnetoresistance (TMR) elements rotated at 90 degree intervals in the package and connected in a bridge configuration. The output can be configured to represent the sine and cosine functions of the magnetic field applied to the sensor. Each TMR sensor element has a resistance of approximately 1.25 MΩ. Outputs are proportional to the supply voltage and peak-to-peak output voltages are comparatively large than other sensor technologies.

Tunnel magnetoresistance is a magnetoresistive effect that occurs in a magnetic tunnel junction, which is a component consisting of two ferromagnets separated by a thin insulator. If the insulating layer is thin enough (typically a few nm), electrons can tunnel from one ferromagnetic layer to the other. The direction of the two magnetizations of the ferromagnetic films can be switched individually by an external magnetic field. If the magnetizations are in a parallel orientation it is more likely that electrons will tunnel through the insulating film than if they are in the antiparallel orientation. Consequently, such a junction can be switched between two states of electrical resistance, one with low and one with very high resistance.

1.3 OPERATION OF TMR SENSOR

Each sensor elements contain two magnetic layers: a "pinned," or fixed direction layer; and a movable direction, or "free" layer. The diagram below illustrates the configuration, using arrows to represent the two layers [1]:

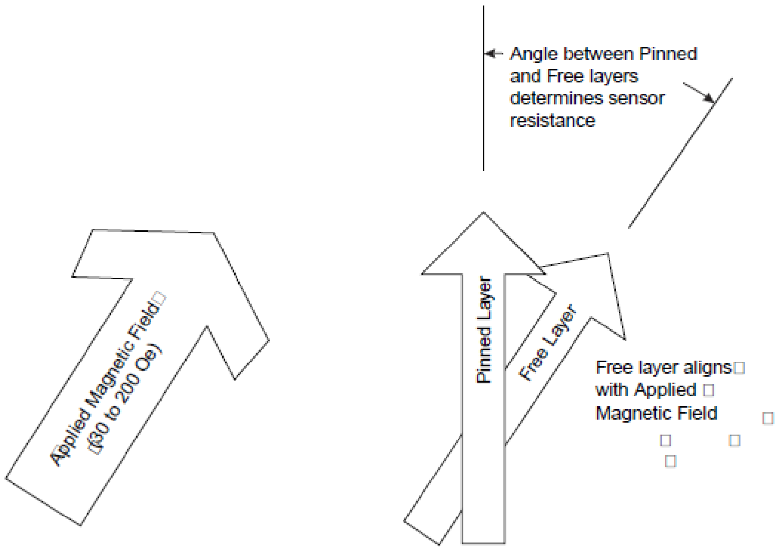


Figure 1.1: Shows the direction of applied magnetic field and the angle between that and the pinned layer to determine the sensor resistance [1]

The sensor element free layers will align with the external field. As the applied field changes direction, the angle between the free layer and the pinned layer changes, changing the resistance of TMR elements, which changes the device output voltages.

1.4 OBJECTIVE OF THE PROJECT

The main objective of this project is to develop a TMR sensor based pickup for electric guitar and proper signal processing algorithms for producing reliable and accurate sound capable of replicating the sound qualities of an inductive pickup which is predominantly used in the music industry now. The TMR sensor array is mounted in PCB along with signal conditioning circuit so that it can be mounted easily on electric guitar.

1.5 ORGANIZATION OF THE THESIS

Chapter 1 of the thesis gives an overview of TMR sensor and its application as an electric guitar pickup.

Chapter 2 deals with the theoretical aspects of the project including theory about string vibration and magnetic field generated by a current carrying wire. This chapter also tells about the theory of the signal conditioning circuit used in the project.

Chapter 3 details with the practical implementation of the project, capturing the waveform from the circuit, using signal processing algorithms and developing Look-up Tables.

Chapter 4 tells about the final stage of the project, verification of the results using Laser Distance Sensor, generation of Look up Tables using Laser Distance Sensor and different algorithm for signal processing.

Chapter 5 includes the results obtained from the project, the future scope of work in addition to the conclusion.

CHAPTER 2

2.1 PHYSICS OF GUITAR STRING VIBRATION

BASIC PRINCIPLE

If a string is tied between two fixed supports, pulled tightly and sharply plucked at one end, a pulse will travel from one end of the string to the other. When you pluck the string, you put energy into an elastic medium, and this energy travels through the medium as a transverse pulse. Transverse means that the amplitude is at right angles to the direction of propagation. The speed of the pulse through the medium, in this case the string, is a function of the properties of the string. Specifically, it is a function of the linear density of and the tension in the string. Making the string tighter and lighter increases the pulse speed, and making the string looser and heavier slows the pulse speed down. The relation between pulse speed, tension and linear density is given by the following equation [2]:

$$v = \sqrt{\frac{T}{\mu}} \tag{2.1}$$

where v is the pulse speed, T the tension in the string, and μ the string's linear density.

For a guitar string, the string is fixed at both ends; hence reflection occurs at both ends of the guitar string. The two wave pulses formed produce a stationary wave, i.e. a wave which is non-travelling.

When plucking the string, it will be initially displaced from its equilibrium state. When released, the wave will vibrate from its initial shape, to its mirror shape, and back again.

Suppose a string of length L is kept fixed at the ends x = 0 and x = L and sine waves are produced on it. For certain wave frequencies, standing waves are set up in the string. Due to the multiple reflection at the ends and damping effects, waves going in the positive x-direction interfere to give a resultant wave [2]:

$$y1 = Asin(kx - \omega t) \tag{2.2}$$

Similarly, the waves going in the negative x-direction interfere to give the resultant wave:

$$y2 = Asin(kx + \omega t + \delta) \tag{2.3}$$

The resultant displacement of the particle of the string at position x and at time t is given by the principle of superposition as:

$$y = y1 + y2 \tag{2.4}$$

$$= A[sin(kx - \omega t) + \ sin(kx + \omega t + \delta)] \tag{2.5}$$

$$= 2Asin\left(kx + \frac{\delta}{2}\right)\cos(\omega t + \frac{\delta}{2}) \tag{2.6}$$

If standing waves are formed, the ends x = 0 and x = L must be nodes because they are kept fixed. Thus, we have the boundary conditions y= 0 at x= 0 for all t and y = 0 at x = L for all t.

The first boundary condition is satisfied by equation if:

$$\sin\left(\frac{\delta}{2}\right) = 0 \tag{2.7}$$

The second boundary condition will be satisfied if:

$$\sin(kL) = 0 \tag{2.8}$$

Therefore,

$$L = \frac{n\lambda}{2} \tag{2.9}$$

If the length of the string is an integral multiple of $\lambda/2$, standing waves are produced. Again writing λ= vT = v/f, equation becomes:

$$f = \frac{nv}{2L} = \frac{n}{2L}\sqrt{\frac{F}{\mu}} \tag{2.10}$$

The lowest possible frequency is:

$$f_0 = \frac{nv}{2L} = \frac{n}{2L}\sqrt{\frac{F}{\mu}} \tag{2.11}$$

This is the fundamental frequency of the string.

The other natural frequencies with which standing waves can be formed on the string are [2]:

$f_1 = 2f_0 = \frac{2v}{2L} = \frac{2}{2L}\sqrt{\frac{F}{\mu}}$, 1st overtone or 2nd harmonic

$f_2 = 3f_0 = \frac{3v}{2L} = \frac{3}{2L}\sqrt{\frac{F}{\mu}}$, 2nd overtone or 3rd harmonic

$f_3 = 4f_0 = \frac{4v}{2L} = \frac{4}{2L}\sqrt{\frac{F}{\mu}}$, 3rd overtone or 4th harmonic

In general, any integral multiple of the fundamental frequency is an allowed frequency. These higher frequencies are called overtones. Thus, v1 = 2 vo is the first overtone, v2 = 3 vo is the second. overtone etc. An integral multiple of a frequency is called its harmonic. Thus, for a string fixed at both the ends, all the overtones are harmonics of the fundamental frequency and all the harmonics of the fundamental frequency are overtones. This property is unique to the string and makes it so valuable in musical instruments such as guitar.

When a string vibrates according to equation (15.24) with some natural frequency, it is said to vibrate in a normal mode. For the nth normal mode, the equation for the displacement is:

$$y = 2Asin\left(\frac{n\pi x}{L}\right)cos\omega t \qquad (2.12)$$

For fundamental mode, n = 1 and the equation of the standing wave is [2] :

$$y = 2Asin\left(\frac{\pi x}{L}\right)cos\omega t \qquad (2.13)$$

For our experiment, we used the A string. For a typical A string in a standard guitar, where L=0.63m, μ=0.0166 Kg/m, tension T=318.88 N we can calculate the fundamental frequency:

$$f_0 = \frac{n}{2L}\sqrt{\frac{F}{\mu}} \qquad (2.14)$$

By putting the above values, we find f_0 = 110 Hz, which is the standard for open A string [16].

2.2 MODELING STANDING WAVES OF 'A' STRING

Here is a snippet of the code used to simulate the standing waveform of the A string in MATLAB :

```
figure('color','white')
x=0:0.01:10;
A=1;
L=0.63;
F=110;
for t=0:0.1:5
   y=2*A*sin((pi*x)/L)*cos(2*pi*f*t);
   plot(x,y,'b');
   hold all;
   ylim([-2 2]);
   title('Standing Wave')
   getframe;
end
```

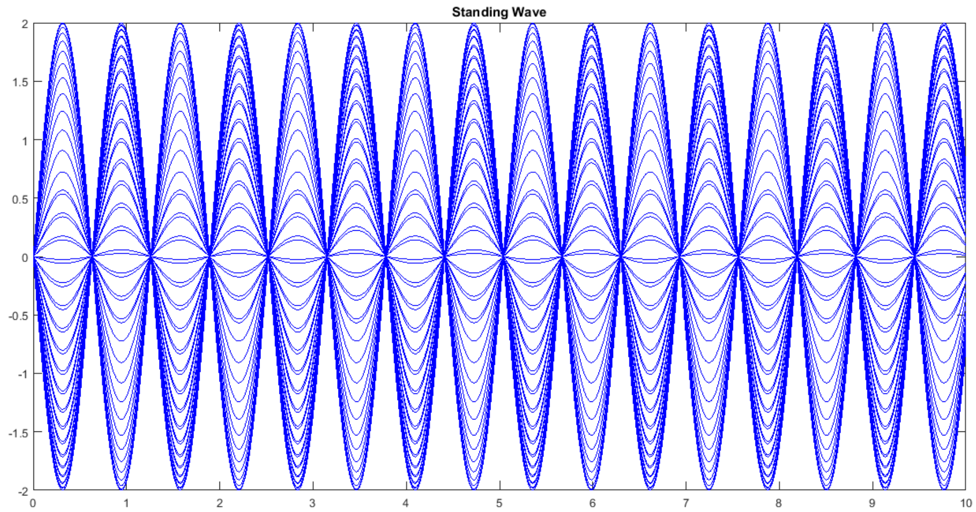


Figure 2.1: Simulation of standing waves produced by 'A' string vibrating at 110 Hz

One characteristic of the standing wave pattern is that there are points along the medium that appear to be standing still. These points, sometimes described as points of no displacement, are referred to as nodes. There are other points along the medium that undergo vibrations between a large positive and large negative displacement. These are the points that undergo the maximum displacement during each vibrational cycle of the standing wave. In a sense, these points are the opposite of nodes, and so they are called antinodes. A standing wave pattern always consists of an alternating pattern of nodes and antinodes.

2.3 MODELLING MAGNETIC FIELD GENERATED BY THE CURRENT CARRYING STRING

We now simulate the magnetic field generated by current carrying string. We know that a magnetic field can be produced by moving charges or electric currents. The basic equation governing the magnetic field due to a current distribution is Biot-Savart law.

From Biot-Savart law, the magnetic field due to a long straight wire, for our case, a string is[3]:

$$B = \frac{\mu_0 I sin(\omega t+\varphi)}{2\pi r} \tag{2.3.1}$$

Where, $Isin(\omega t + \varphi)$ is the alternating current through the string, and r is the distance from the string to the TMR sensor. Therefore, we take two things into account:

i. The magnetic field is proportional to the applied current.
ii. The magnetic field sensed by the TMR sensor is inversely proportional to the distance between them. We will use this property later in the project to derive the exact relationship between the output voltage from the TMR circuit and the distance between the sensor and the string.

A MATLAB script was used [4] to see how the magnetic field lines would look like for a string which is left unplucked:

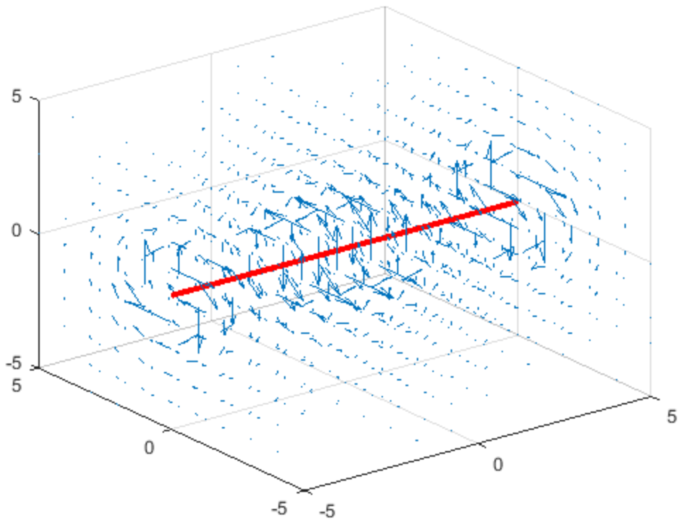


Figure 2.2: Simulation of magnetic field lines generated by the 'A' string carrying sinusoidal current at 10KHz. The string here is unplucked.

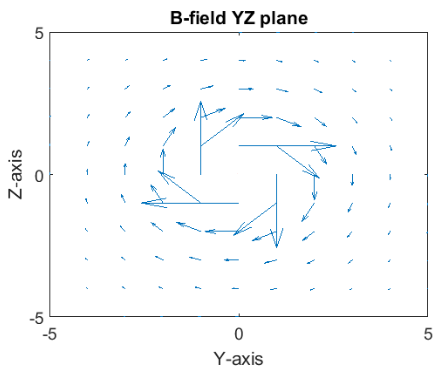


Figure 2.3: Simulation of the magnetic field lines at the cross section of the string

2.4 SIGNAL CONDITIONING CIRCUIT: THEORY OF OPERATION

The complete circuit diagram is shown below:

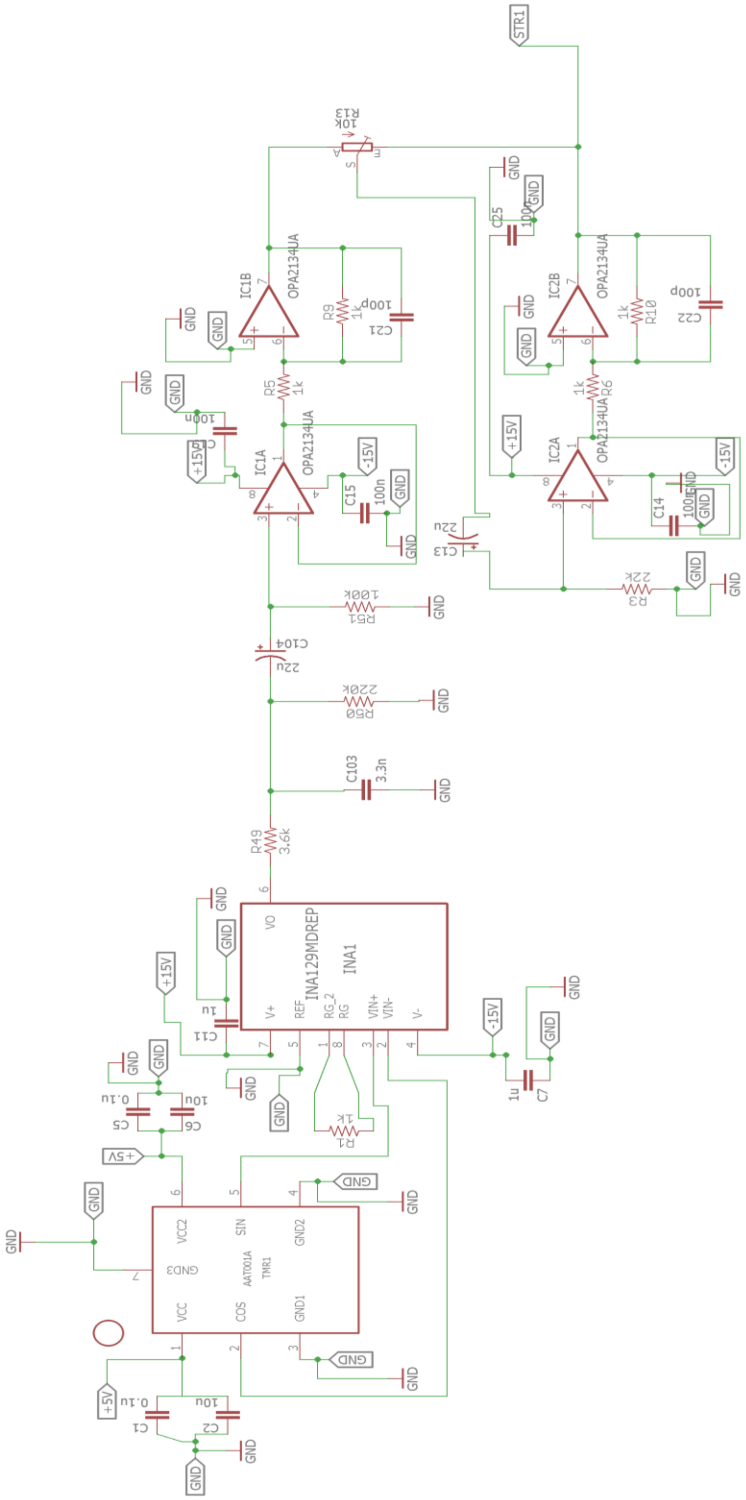
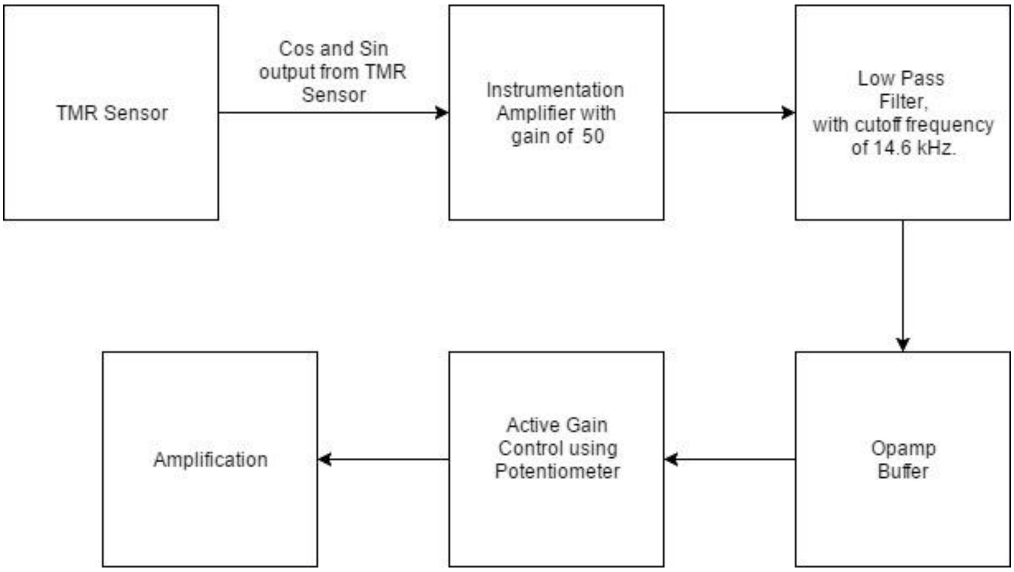


Figure 2.6: The complete circuit diagram used in the project including TMR sensor and INA in the first stage, the low pass filtering in the second stage and active gain control in the third stage.

Block diagram for the prototype circuit:



2.4.1 FIRST STAGE

Below is the circuit diagram for the initial stage of the circuit. From the TMR datasheet, we find that the TMR sensor contains four sensors with the pinned layer which is 90 degrees apart. The resistors are connected as two half bridges which provide the sine and cosine outputs which we see in the circuit diagram below:

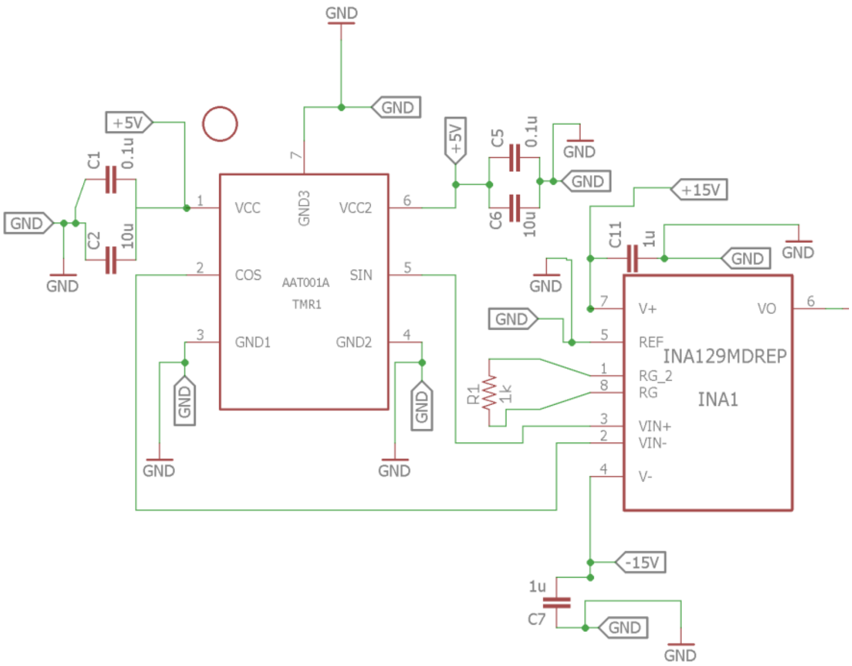


Figure 2.7: Circuit diagram of the TMR and INA setup

We have chosen a dedicated instrumentation amplifier, the INA 129 because it has been outfitted with input protection circuit and input buffer amplifiers, which eliminate the need for input impedance matching and make the amplifier particularly suitable for use in measurement and test equipment. Additional characteristics of the INA129 include a very low DC offset, low drift, low noise, very high open-loop gain, very high common-mode rejection ratio, and very high input impedances. The INA129 is used where great accuracy and stability of the circuit both short and long term are required. The gain for INA129 is given by [13]:

$$G = 1 + \frac{49.4k\Omega}{R_G} \tag{2.4.1}$$

Where, R_G in our case is $1k\Omega$, therefore the total gain from the INA is G = 50.4

2.4.2 SECOND STAGE

A passive first-order low pass filter at the input of the circuit attenuates unwanted noise from outside the required frequency. This filter is made up of the input source impedance RIN, as well as R5 and C5.

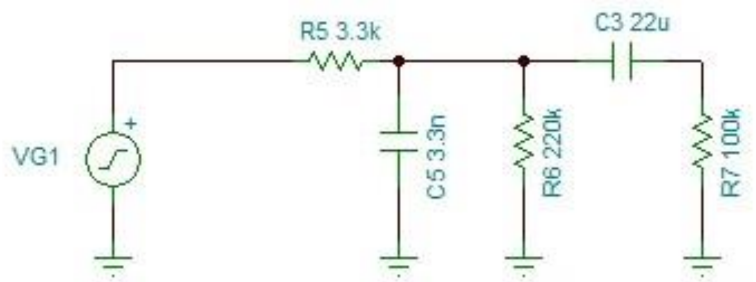


Figure 2.8: Circuit diagram of the low pass filtering stage

The RF filter at the input of the circuit should provide significant attenuation at frequencies outside the required range while preserving the circuit's gain and phase performance. 14.6 kHz is selected as the cut-off frequency for the filter.

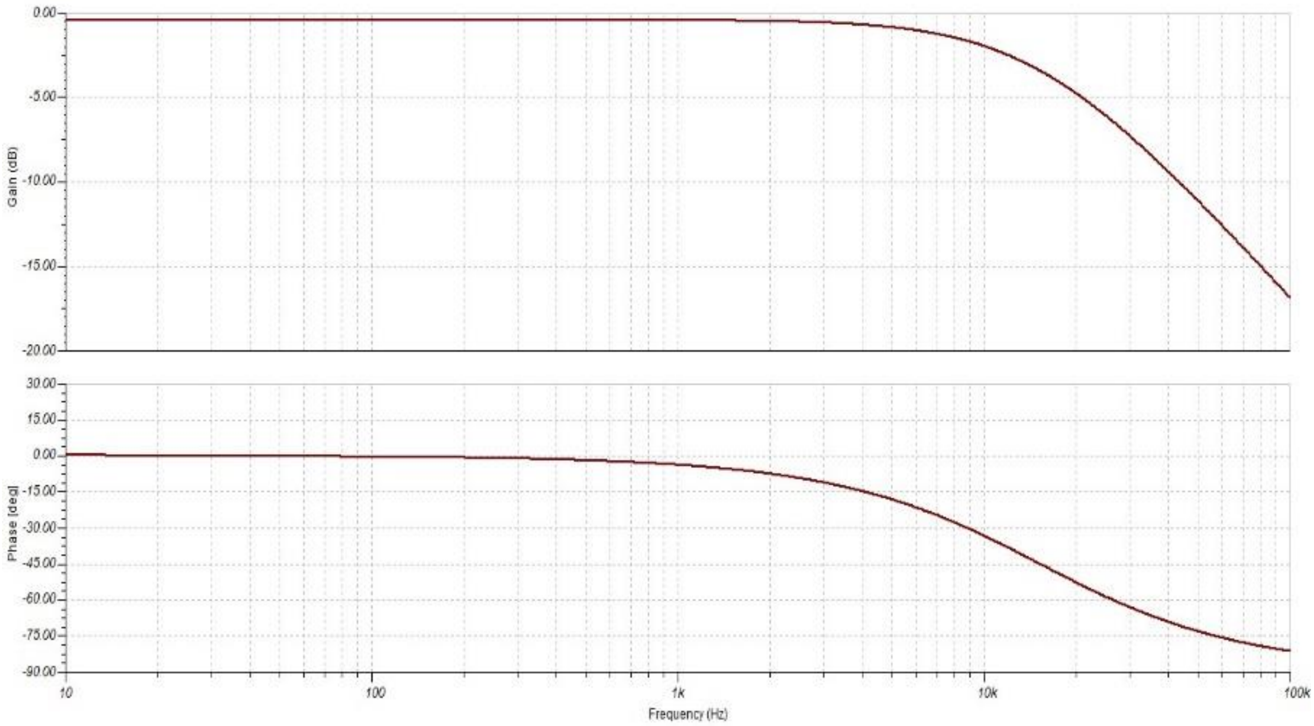


Figure 2.9: Bode plot simulated in TINA for the low pass filter

Next in the signal path is amplifier U1A, configured as a non-inverting buffer. While the input source impedance in this design is well-defined, in practical the system will have to interface with many different types of inputs. A buffer amplifier is necessary in order to preserve the behaviour of the system independent of the input source impedance RIN.

The last stage of the input circuit consists of amplifier U1B, configured as an inverting amplifier with gain equal to R3/R4. This amplifier serves to counteract the inherent phase inversion of the active volume control stage and to provide additional gain.

4.2.3 THIRD STAGE

Let us now look at the mid stage of the circuit.

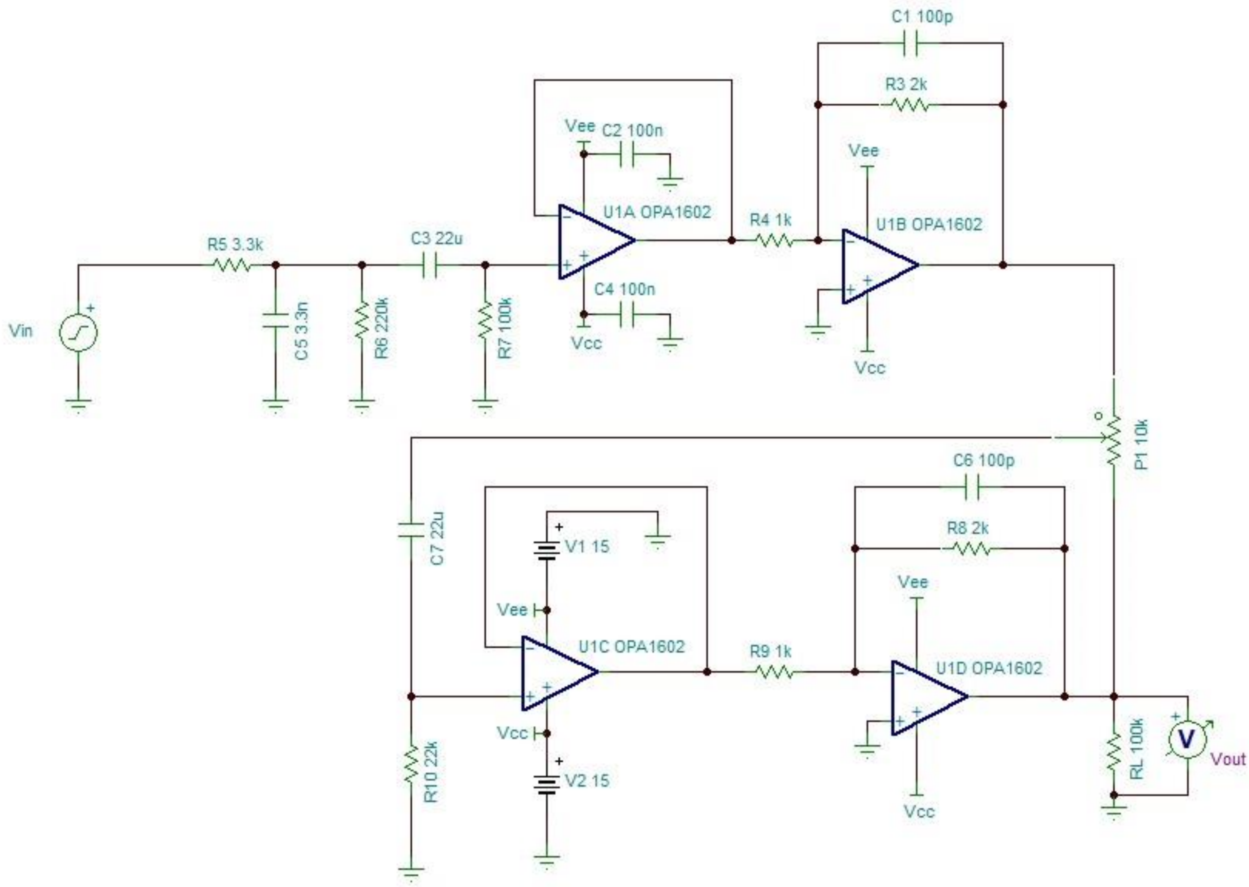


Figure 2.10: Circuit diagram for the buffer amplifier and active gain control

As we calculate the transfer function of the circuit, we find that it is based on the first stage gain set by the resistance ratio of R4 and R3, the resistance/position of potentiometer P1 and the second stage gain which is set by the ratio of resistance R9 and R8.

At the input stage of the circuit, there is a first order low pass filter which attenuates signal in the RF frequencies created by the R5 and C5. The remaining parts of the circuit are used for ac coupling, amplifier input biasing, stability compensation, and power supply decoupling.

The transfer function is calculated for this circuit as:

$$V_{OUT} = V_{IN}\left(\frac{R3}{R4}\right)\left[\frac{P1_{Rotation}}{(1-P1_{Rotation})+\frac{R9}{R10}}\right] \qquad (2.4.2)$$

The active gain control circuit is a type of shunt-feedback control which has two main advantages over other solutions [5]. Firstly, it achieves excellent logarithmic behaviour with a linear potentiometer, and secondly, its transfer function is not a function of the potentiometer track resistance or any other discrete resistances. In fact, the gain characteristic is only a function of the percentage of potentiometer rotation and the maximum gain set by the ratio of two resistors. This allows a standard-tolerance potentiometer to be used with excellent results. This is very necessary at the input stage as we do not want any noise in the signal due to the potentiometer.

Active gain controls in general have several other advantages over other passive gain controls. The use of amplifiers allows for gain early in the signal chain, improving noise performance. The current drive capabilities of the amplifiers allow low resistance values to be used, which minimizes Johnson noise and capacitive crosstalk. Finally, the high input impedance and low output impedance of the amplifiers ensures a robust design which, with some minor exceptions, maintains its performance independent of the source and load impedance.

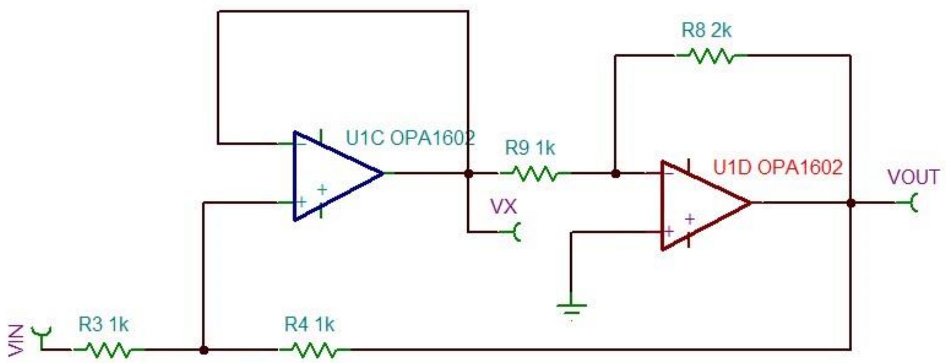


Figure 2.11: Circuit for simulation of active gain control

The simplicity of this signal conditioning circuit allows for a straightforward transfer function analysis. To find out the transfer function, the potentiometer P1 have been replaced with two discrete resistors: R_X, which represents the percentage of potentiometer resistance after the wiper, and $R_{1\text{-}X}$, which represents the percentage of pot resistance before the wiper. Together these two resistors represent the total possible span of potentiometer rotation. Figure 2.11 shows the modified circuit schematic.

To analyze the transfer function, we first define the voltage at the wiper as V_X. Since amplifier U_{1C} is configured as a non-inverting buffer, the voltage at the output of U_{1C} is also equal to V_X. V_X is applied as the input to amplifier U_{1D}, which is configured as an inverting amplifier with gain G. Therefore, the output voltage V_{OUT} must equal the product of V_X and G. This relationship can be solved for V_X and written as shown in Equation 2.4.3. A negative sign is necessary to account for the inverting configuration.

$$V_X = -\frac{V_{OUT}}{G} \qquad (2.4.3)$$

Next, Kirchoff's current law is used to analyze the current flow at the wiper. This law states that the sum of the currents flowing into a node equal the sum of the currents flowing out of a node, so these currents can be written as shown in Equation 2.4.4.

$$i_1 = i_2 + i_3 \qquad (2.4.4)$$

Current i_3 is the current flow into the positive input of amplifier U_{1C}. Assuming that U_{1C} is an ideal amplifier, it has infinite input impedance and therefore no current can flow in that direction. Since i_3 must equal zero, Equation 2.4.4 can be simplified to Equation 2.4.5.

$$i_i = i_2 \qquad (2.4.5)$$

Ohm's Law can now be applied to currents i_1 and i_2, and Equation 2.4.5 can be rewritten in terms of V_{IN}, V_X, V_{OUT}, $R_{1\text{-}x}$, and R_x as shown in Equation 2.4.6.

$$\frac{V_{IN}-V_X}{R_{1-x}} = \frac{V_X-V_{OUT}}{R_X} \qquad (2.4.6)$$

Using the relationship from Equation 2.4.3, V_X can be substituted for ($-V_{OUT}/G$). Equation 2.4.7 shows the new equation after making the substitution.

$$\frac{V_{IN}+\frac{V_{OUT}}{G}}{R_{1-x}}=\frac{\frac{-V_{OUT}}{G}-V_{OUT}}{R_X} \qquad (2.4.7)$$

Equation 2.4.7 can be solved for V_{OUT}/V_{IN}, and the transfer function can be written as shown in Equation 2.4.8.

$$\frac{V_{OUT}}{V_{IN}}=-\frac{R_X}{R_{1-X}+\frac{1}{G}} \qquad (2.4.8)$$

Amplifier U_{1D} is configured as an inverting amplifier, so its gain G is equal to R8/R9. After substituting in for G in Equation 2.4.8, the transfer function can be written as shown in Equation 2.4.9.

$$\frac{V_{OUT}}{V_{IN}}=-\frac{R_X}{R_{1-X}+\frac{R_9}{R_8}} \qquad (2.4.9)$$

CHAPTER 3

3.1 PHASE 1 OF PROJECT

After designing the signal conditioning circuit and simulating in Texas Instruments TINA, I built the circuit on breadboard. In the initial stage of the project, it was chosen to send sinusoidal current through the wire while placing the wire just above the TMR sensor to study the output waveform from the TMR sensor. Even the TMR circuit board was attached to the breadboard to make the prototyping setup more compact. The components used for signal condition circuit was:

i. AAT001: This was the TMR sensor under which the whole project is based on.
ii. INA 129: The two cos and sine outputs from the TMR sensor were directly connected to this dedicated instrumentation amplifier IC.
iii. OPA 2134: These are low distortion, low noise opamps specifically designed for audio applications. [14]

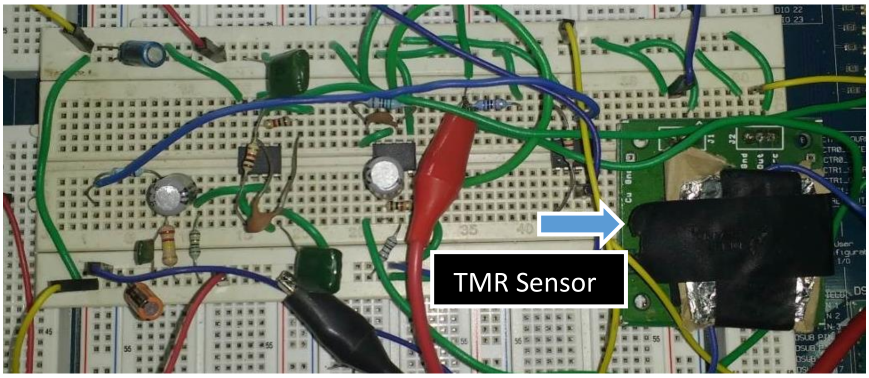


Figure 3.1 Prototype circuit at the initial stage of the project including TMR sensor board

iv. OPA 541: The current through the wire was provided by OPA 541, which is a power opamp, which can deliver continuous output current up to 5A. The input voltage to the opamp circuit was provided by the function generator of NI ELVIS board. [15]

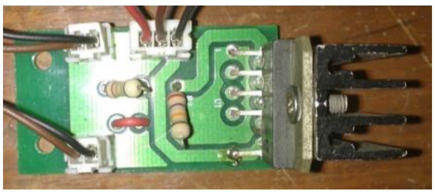


Figure 3.2: OPA 541 power amplifier circuit board used to provide current to the wire

After the hardware setup was complete, using the function generator of the Elvis board, we simulated various frequencies through the wire passing over the TMR sensor and compared the output from the signal condition circuit with the original signal.

In two figures shown below, the figure on the left shows the waveform of the original sinusoidal signal of 50Hz at the top and the waveform of the signal from the circuit at the bottom. It was found that the frequency as well as the shape of the waveform were correctly captured by the TMR sensor. Similarly, for the figure shown in the right we passed a sinusoidal current of frequency 1kHz which is shown at the top and the waveform from the circuit is shown below. In this case too, both the frequency and the shape of the waveform are same.

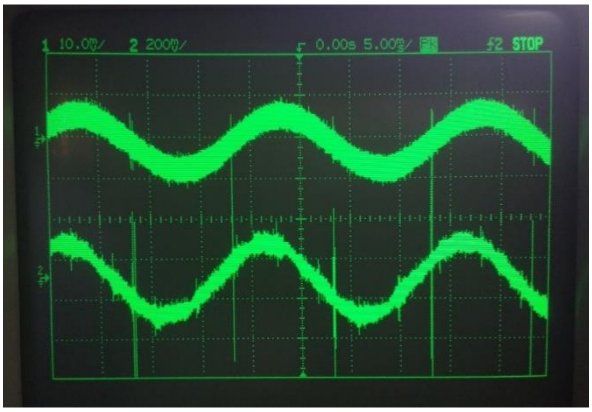


Figure 3.3: 50Hz signal at the top and signal captured from the TMR sensor at the bottom

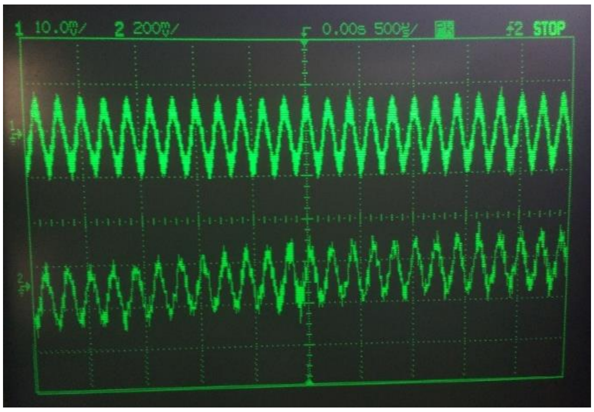


Figure 3.4: 1kHz signal at the top and signal captured from the TMR sensor at the bottom

Thus, the first phase of the project was successful in producing the following results which would be useful in the later stages of the project:

i. The signal conditioning circuit build around the TMR sensor was successfully able to sense the frequency of the signal which was passed through the wire.

ii. It was successfully able to capture the shape of the original waveform which was sent through the wire.

iii. There was a distortion as well as in the signal from the circuit when the current through the wire was lower than 90mA. Thus, at 100mA and above, the signal was perfect.

3.2 PHASE 2 OF PROJECT

After verifying in the first phase of the project that the frequency of the current signal passed through a single strand wire was accuretly detected by the TMR sensor, we finally chose to test the setup with an acoustic guitar. The TMR sensor board was placed directly below the 5th string of the guitar or more specifically, the open 'A' string of the guitar. All the simulations and experiments as well as the results obtained in the further part of the project has been carried out on the open 'A' string of the guitar.

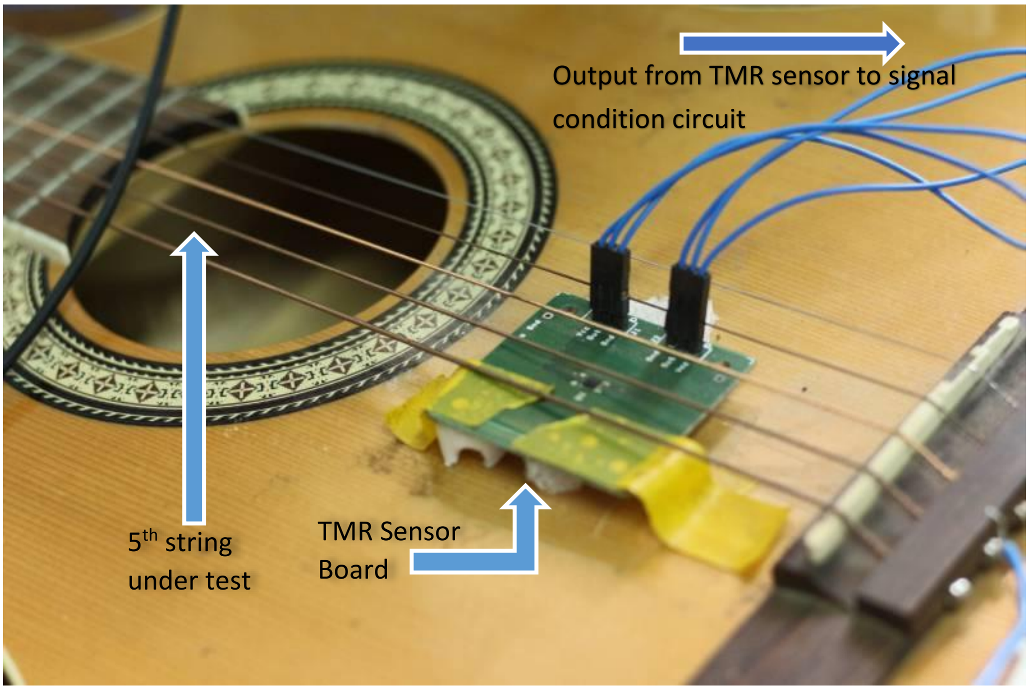


Figure 3.5: TMR sensor placed just below the 'A' string

These are the major highlights of the 2nd stage of the project:

i. 10kHz was chosen as the frequency of the current signal which was passed through the open 'A' string of the guitar.
ii. The output from the signal condition circuit was connected to the 'analog input 0' and was captured using DAQ in LabVIEW at the sampling rate of 500,000 Hz capturing 500,000 samples.

iii. The LabVIEW VI was made in this stage which exported the incoming signal to MATLAB for signal processing after passing through bandpass filter.

iv. In MATLAB, the peaks of the modulated signal were extracted to get the amplitude waveform of the signal.

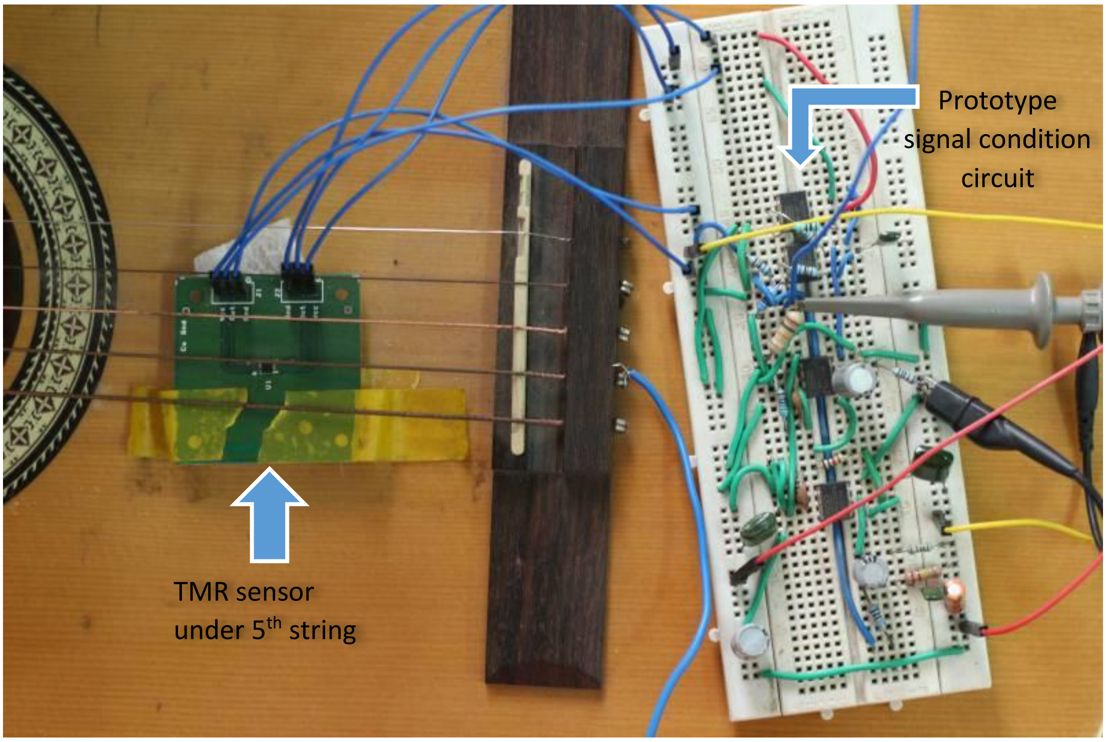


Figure 3.6: Top view of the TMR sensor and the signal conditioning circuit placed on the Guitar

3.3 MODELING THE WAVEFORM FROM THE CIRCUIT AS AM

Let the amplitude of the string vibration 0.1 (Amplitude of the modulating signal) and let the frequency with which it is vibrating be 110Hz. The equation of the modulating signal can be written as:

$$y_m = A_m \sin(2\pi f_a t) \quad (3.3.1)$$

Where, A_m is the amplitude of the modulating signal and f_a is the frequency of the modulating signal.

Now, we define the equation of the carrier signal, which is the signal through the guitar string. Let the frequency of vibration be 10KHz and the amplitude of vibration be 1V. The equation of carrier signal can be written as:

$$y_c = A_c \sin(2\pi f_c t) \quad (3.3.2)$$

Where, A_c is the amplitude of the carrier signal and f_c is the frequency of the carrier signal.

Therefore, the equation of the amplitude modulation can be written as:

$$y = A_c(0.5 + A_m \sin(2\pi f_a t))\sin(2\pi f_c t) \quad (3.3.3)$$

Now we compare the waveform we got from simulation of the above equation vs the waveform that we got from the signal condition circuit.

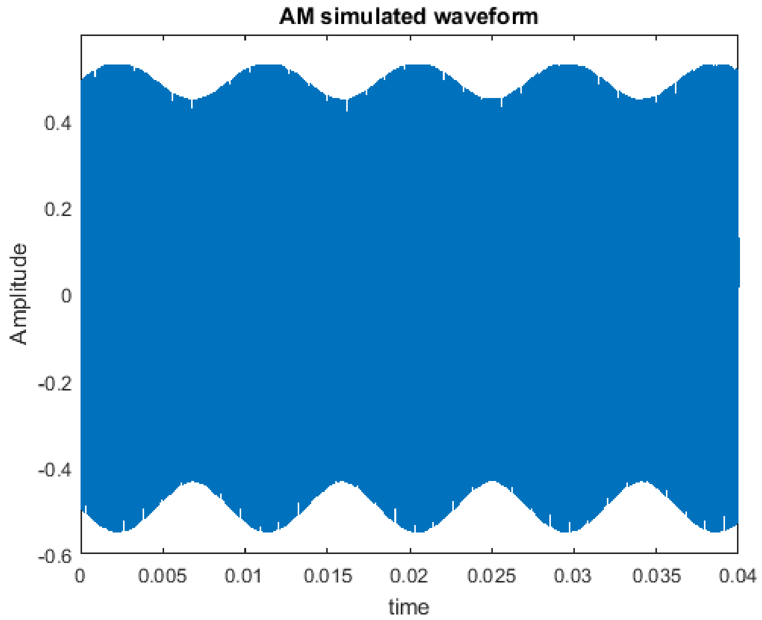


Figure 3.7: Simulation of the AM waveform from the given equation

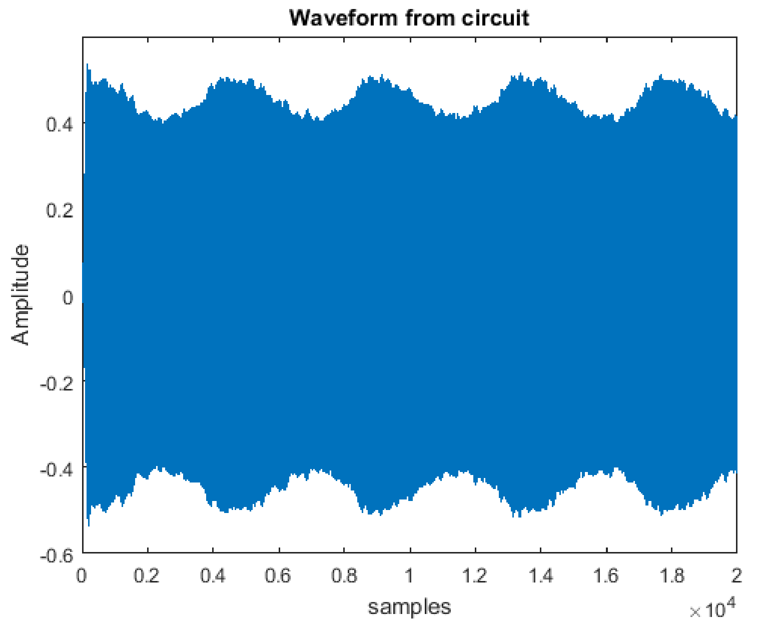


Figure 3.8: The waveform we got from the signal conditioning circuit

As the above two waveforms look strikingly similar, we can approximate the equation of the signal coming from the signal condition circuit as:

$$y = A_c(p + A_m \sin(2\pi f_a t))\sin(2\pi f_c t) \tag{3.3.4}$$

Where, A_c, p, A_m, f_a and f_c are the parameters which needs to be modified accordioning to get the closest approximation

3.4 MODELING THE DISPLACEMENT VS AMPLITUDE CHARACTERISTICS

As the current carrying string is brought nearer to the TMR sensor, the output voltage from the signal conditional circuit does not change linearly. Thus, there is a nonlinear relationship between the displacement from current carrying string (to the sensor) and the output voltage from the TMR sensor. Therefore, it is very important to find the output characteristics of the TMR sensor as the current carrying string is moved towards/away from the sensor. To do this, a hardware setup was made comprising of flat head screw and the string passing through just beneath the flat head. The setup was built in such a way such that as the screw rotates, the flat head of the screw pushes the string down linearly, thereby there is a linear motion of the string with the screw being rotated.

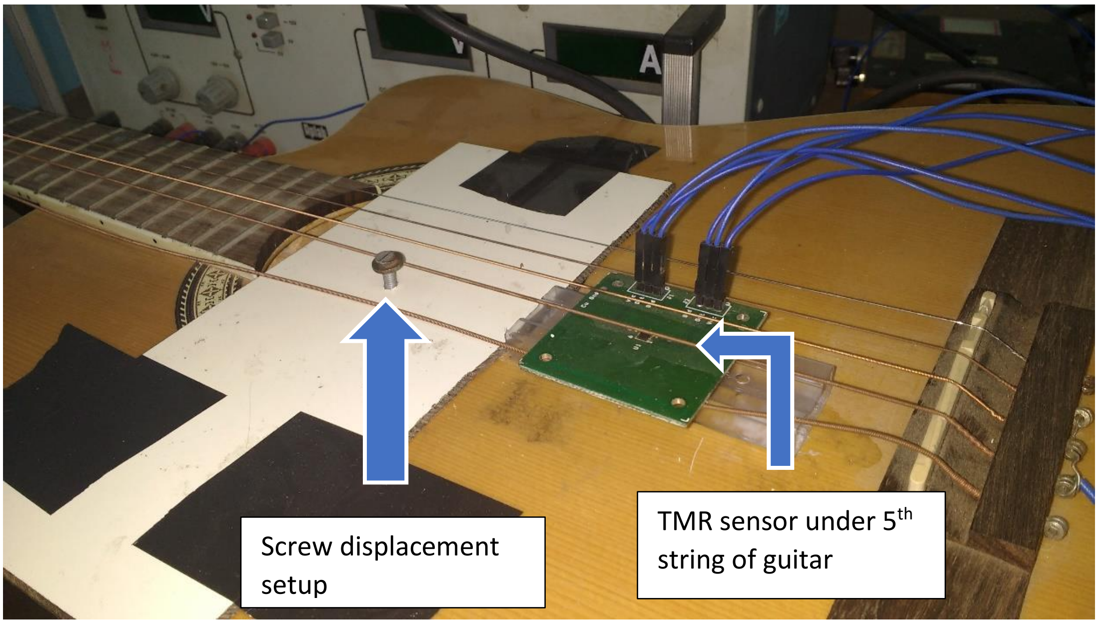


Figure 3.9: Hardware setup for generating LUT using screw displacement method

Now, we should consider the factor of how much the string is displaced for one complete revolution of the screw. To do this, we knew the initial height from the PCB to the mean position of the string to be 2.08 mm with the help of Vernier callipers. From the datasheet of the TMR sensor, the height of the IC is 0.85 mm. The diameter of the string was found to be 0.88 mm and the radius is 0.44 mm. Therefore, the height from the sensor to the mean position of the guitar is 1.24 mm. The transition relation of the screw is:

$$x = pitch(\frac{\theta}{2\pi}) \tag{3.4.1}$$

where, pitch of the screw is the distance moved by the spindle per revolution.

From the experiment, it is found that it takes roughly around three complete revolutions and one quarter rotation of the screw to move the string from its mean position to the position where it is just about to touch the sensor. Therefore, from calculations the string displaces 0.3815mm/revolution or 0.09538mm/quarter of a revolution.

I used this property to capture 14 datasets of displacement of string at 0.09538mm/displacement by rotating the screw one quarter at a time and measured the mean output voltage at the same time.

Table 1.1: This table shows the displacement from sensor to the string and its corresponding amplitude value from the circuit.

Sl.	Distance from the string to the sensor (mm)	Amplitude (V)
1	1.24	0.9187
2	1.14	0.9146
3	1.05	0.9237
4	0.95	0.9491
5	0.86	0.9926
6	0.76	1.0259
7	0.66	1.0259
8	0.57	1.0528
9	0.47	1.0870
10	0.38	1.1293
11	0.28	1.1800
12	0.19	1.2945
13	0.09	1.3583
14	0.0004	1.4413

Now as the data for the displacement and their corresponding amplitude have been taken, I have used the curve fitting toolbox in MATLAB to get the function between displacement vs amplitude.

The Curve Fitting app in MATLAB provides a flexible interface where we can interactively fit curves and surfaces to data and view plots. We can create, plot, and compare multiple fits, use linear or nonlinear regression, interpolation, smoothing, and custom equations, view goodness-of-fit statistics, display confidence intervals and residuals, remove outliers and assess fits with validation data, automatically generate code to fit and plot curves and surfaces, or export fits to the workspace for further analysis.

From the data points, we got from the above table, we created two 1-D arrays in MATLAB for displacement and amplitude data respectively.

The graph for the amplitude vs displacement which we got from the screw experiment looks like this, which proves that there exists a non-linear relationship between them.

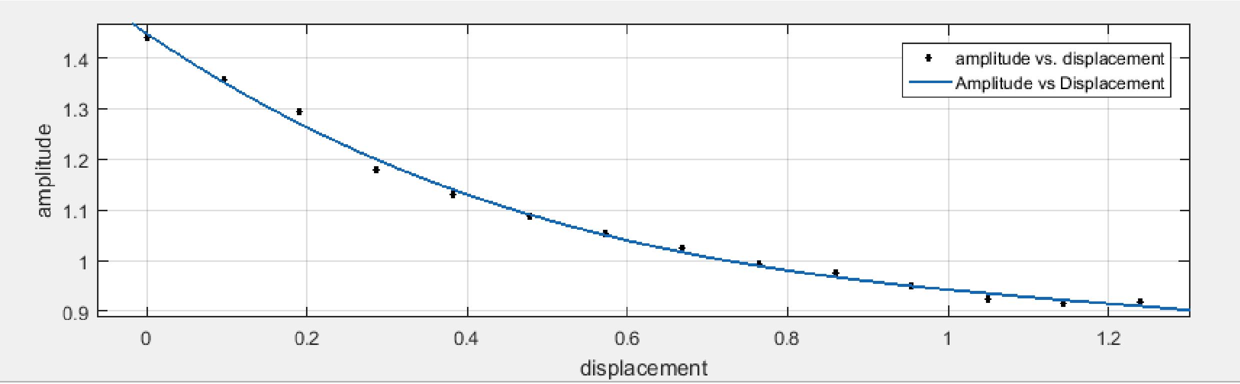


Figure 3.10: The amplitude vs displacement plot to generate the corresponding polynomial function

In order the generate the appropriate polynomial function, we used the 'fit' function to fit polynomials to data. In our case, we used 'poly3' to fit the curve with cubic polynomial. The curve fit polynomial equation which we got is [6]:

$$f(x) = p_1x^3 + p_2x^2 + p_3x + p_4 \quad (3.4.2)$$

Where,

p1 = -0.2173

p2 = 0.7858

p3 = -1.075

p4 = 1.449

The other details which we got from the curve fitting toolbox:

Goodness of fit:

SSE: 0.001653

R-square: 0.9957

Adjusted R-square: 0.9944

RMSE: 0.01286

Therefore, we got the 3rd degree polynomial equation from the chart. Now the problem was to generate the inverse function from the polynomial which we got, because in practical, the input will be in amplitude data from the circuit and the output will be desired displacement data, which is the actual string vibration data.

We therefore use the finverse function in MATLAB to find the inverse of the polynomial. We finally calculated the inverse function as:

$$g(x) = 57199130/(21930489*((((50000*x)/4683-141656625800/11411164443)^2-1871407086274833381497000/105473885895601875201 69)^{(}1/2)-(50000*x)/4683+141656625800/11411164443)^{(}1/3))+((((50000*x)/4683-141656625800/11411164443)^2-1871407086274833381497000/10547388589560187520169)^{(}1/2)-(50000*x)/4683+141656625800/11411164443)^{(}1/3)+4570/1561 \quad (3.4.3)$$

Now, initially our aim was to put this inverse function as one of the blocks right after the input stage, where the amplitude data will feed into the block generating the output as displacement data. The inverse function that we got was not only computationally tough to calculate but the results we got after sending experimental inputs were complex. It would be very difficult to implement such a function in hardware, and therefore we tried to look for an alternating approach.

The approach we came up with was a Look-up Table based approach. A lookup table is an array that replaces runtime computation with a simpler array indexing operation. The savings in terms of processing time can be significant, since retrieving a value from memory is often faster than undergoing an expensive computation or input/output operation. The tables may be precalculated and stored in static program storage, calculated as part of a program's initialization phase, or even stored in hardware in application-specific platforms.

Our look-up table is a matrix of output voltages from the signal conditioning circuit, and the corresponding displacement values from the base of the TMR sensor to the string. The use of a look-up table in computer data acquisition and analysis greatly enhances the speed of acquisition and accuracy of experimental data. In any experiment where look-up tables are used, the digitized voltages, prior to statistical analysis, are routed through the look-up table in the computer for voltage-displacement conversion.

3.4.1 PROCEDURE TO GENERATE A LOOK-UP TABLE:

Initially, voltage-displacement data (of previously obtained 14 voltage-displacement pairs) is fit to a mathematical curve shape using cftool in MATLAB. This curve is a third-order polynomial such as

$$f(x) = p_1x^3 + p_2x^2 + p_3x + p_4 \tag{3.4.4}$$

where p1, p2, p3, p4 are constants determined for best curve fit.

Once the best fit to the voltage-displacement data is obtained, an entire matrix of voltage-displacement pairs can be generated simply by calculating displacement for assumed voltage values across the range of input data. Though the number of voltage-displacement pairs is not limited in a look-up table, the maximum can be determined by available computer memory and the resolution of the analog-to-digital converter. For example, the A/D converter of National Instruments Elvis 2+ has 16 bits of resolution, the number of pairs in the lookup table need not exceed 65536.

3.4.2 ADVANTAGES OF A LOOK-UP TABLE

Speed of Data Analysis:

Since the look-up table is generated in the computer prior to an experiment, displacement for any digital voltage point is readily available. This greatly enhances the speed of on-line statistical data analysis. For example, if the polynomial in above equation should be employed for voltage-displacement conversion instead of a look-up table, all of the arithmetic operations involved in that polynomial reduce the speed of on-line analysis [7].

Accuracy of Voltage-Displacement Conversion:

For the entire velocity range of an experiment, use of a single polynomial may not give the best fit. But the look-up table need not be restricted to one functional relationship. The displacement range may be divided into several "subranges." Segmental fits can be made to obtain "best" fits in these subranges, and then a look-up table can be generated covering all of the subranges. This enhances the accuracy of voltage displacement conversion [7].

3.5 ALGORITHM FOR EXTRACTING PEAKS FROM WAVEFORM

At the stage of the project, the hardware setup is ready as well as the Look-up Table that we got from the earlier stage. We will now proceed to do signal processing in MATLAB to extract out the actual displacement waveform of string vibration waveform.

We used the same OPA 541 opamp circuit which we used at the initial stage of the project to provide sinusoidal current through the guitar string. The excitation signal to the opamp circuit was provided by the function generator of the ELVIS board. With the help of NI function generator, we provided a sinusoidal signal of frequency 10Khz and an amplitude of 1.5V peak to peak with no DC offset. We also connected an ammeter to the guitar string to measure the amount of current passing through the string, which was found to be 98.8 mA. By experiment, it was found that a current of 100mA and above through the string produced a perfect waveform from the TMR circuit and anything less than 90mA, waveform from the TMR circuit was distorted due to noise. Therefore, I chose 100mA to be sent through the string as standard.

The A string in the guitar is plucked gently and the waveform was recorded. The output from the signal conditioning circuit is applied to the analog input 0 of the ELVIS board. The signal is then recorded in the LABVIEW at 500,000 Hz sampling rate, capturing 500,000 samples in one second. The waveform is then passed through a bandpass filter with lower cutoff frequency of 8kHz and upper cutoff frequency of 12kHz. The waveform was then imported to MATLAB and it was plotted as shown below for full 500,000 samples:

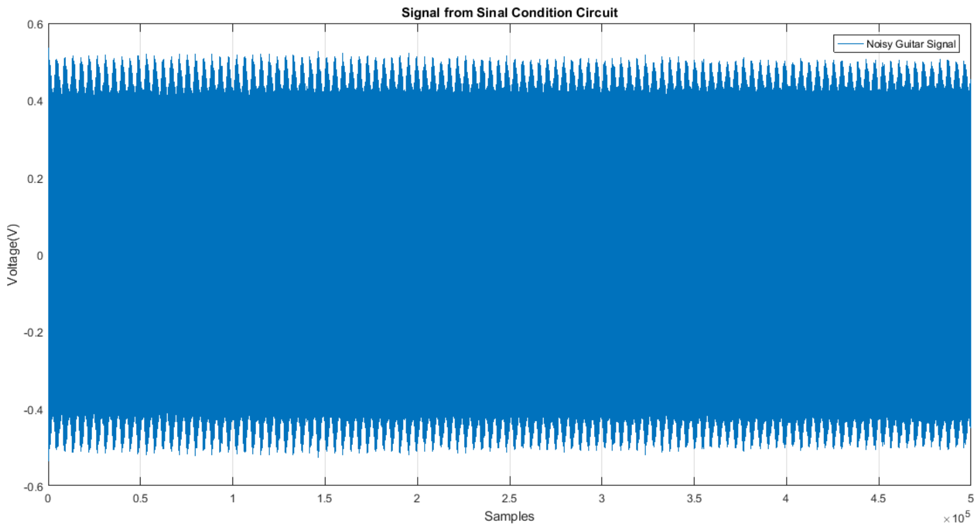


Figure 3.11: The waveform from the signal conditioning circuit, sampled at 500,000Hz, containing 500,000 samples

To view the waveform more distinctively, the plot below shows the waveform for 100,000 samples:

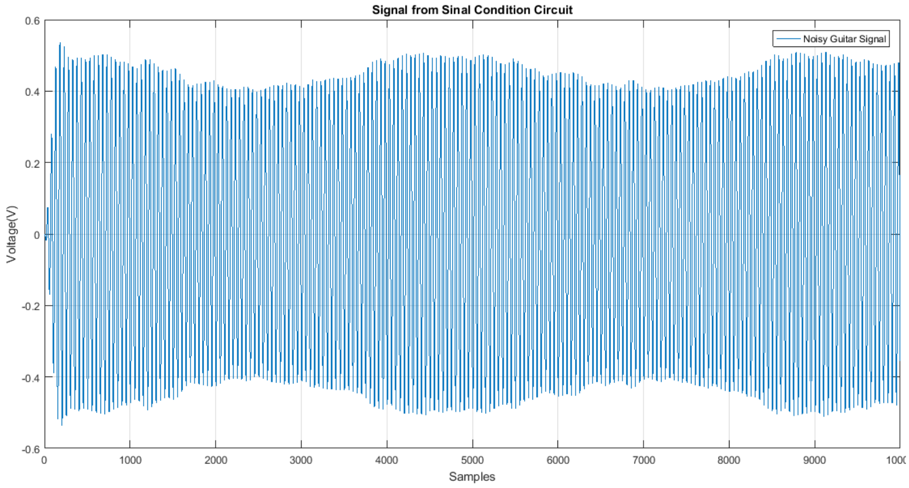


Figure 3.12: The same waveform from the signal conditioning circuit, showing 100,000 samples for comparison

Now, the next task was to extract out the peaks which are enveloping 10Khz carrier signal. By inspection, I have found that the two adjacent positive peaks of the modulated waveform are separated by more than 50 samples. We use this information to remove unwanted peaks by specifying a 'MinPeakDistance'. Then we use the MATLAB command findpeaks(VarName1,'MinPeakDistance',40) to find the location of the positive peaks, where VarName1 was the original waveform imported to MATLAB [8].

The above command returns the location of the positive peaks of the signal, which is shown below:

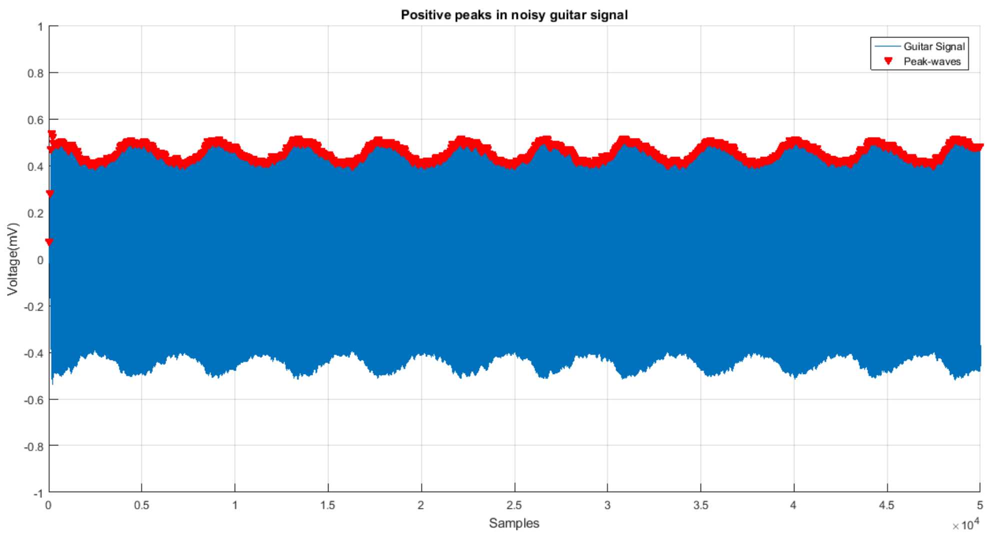


Figure 3.13: Red plot showing the positive peaks of the waveform of 50,000 samples

To find the negative peaks, we invert the original signal and apply thresholds appropriately. To do this, we multiply the original signal by -1 and then use the command findpeaks(VarName1_inv,'MinPeakDistance',40), where VarName1_inv is the inverted waveform of the original waveform. The plot for negative peaks is shown below:

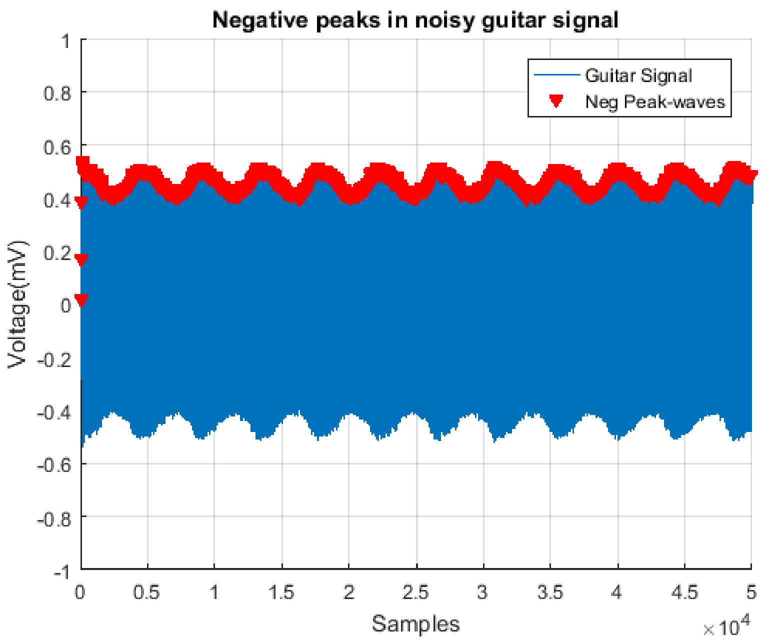


Figure 3.14: Red plot showing the negative peaks of the inverted original waveform of 50,000 samples

Now the locations of the positive occurring peaks are stored in an array ‘loc_peaks’ while the locations of the positive peaks of the inverted signal (negative peaks) are stored in another array ‘loc_negpeaks’.

To make calculations in the further part of the project possible, we need to have the length of both the array same. To do this, we store the length of the two arrays in ‘n’ and ‘p’ respectively. We now compare the ‘n’ and ‘p’. If ‘n’ is greater than ‘p’, this means that the number of positive peaks are greater than the number of negative peaks. In such case, we extend the array length of the negative peaks till the length of positive peaks, so that both lengths are matched and we fill those remaining indexes with the last legitimate position of the negative peaks.

Similarly, if ‘p’ is greater than ‘n’, then the number of location of negative peaks are greater than the number of location of positive peaks. In this case, we extend the length of the array for positive peaks till the length of positive peaks, so that both lengths are matched and we fill those remaining indexes with last legitimate position of the positive peaks. The code snippet for the above algorithm is shown below:

```
n = length(locs_Rwave2)
p = length(locs_Rwave_m2)

if (n > p)
    locs_Rwave_m2(numel(locs_Rwave_m2):numel(locs_Rwave2)) = locs_Rwave_m2(p);
elseif (p > n)
    locs_Rwave2(numel(locs_Rwave2):numel(locs_Rwave_m2)) = locs_Rwave2(n);
end
```

Next, in order to extract the envelope which is formed over the carrier signal, and the peaks of the waveform being extracted we calculate the peak to peak difference as:

```
peaks2=(VarName1(locs_Rwave2) + VarName1_inv(locs_Rwave_m2));
```

By computing the above expression in MATLAB, we actually extract out the envelope in the input waveform which is actually the amplitude waveform. An important point to be noted here is that this waveform is not the actual string vibration waveform, but rather distorted waveform sensed by the TMR sensor as the magnetic field sensed by the TMR sensor is inversely proportional to the distance between them (TMR sensor gives voltage output proportional to the magnetic field). Therefore, we need to use the LUT developed earlier to get the displacement waveform.

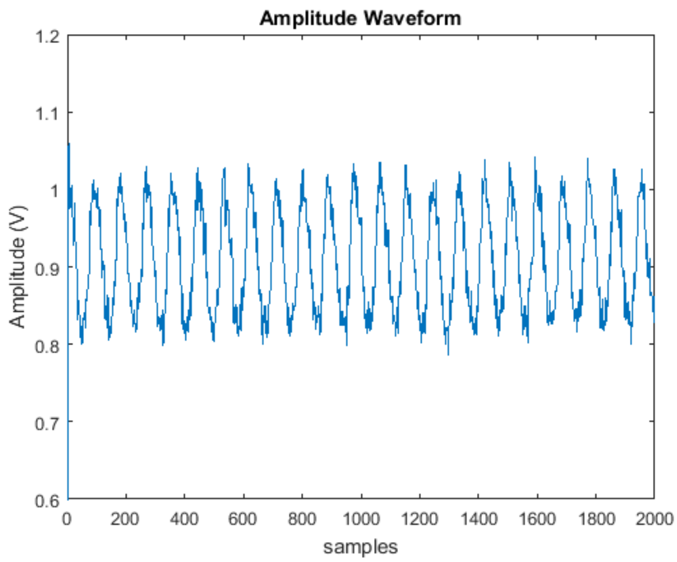


Figure 3.15: Plot showing the extracted amplitude waveform (envelope) from the original waveform

3.6 USING THE LOOK UP TABLE

The code snippet for the generation of displacement waveform from the amplitude waveform using LUT is give below.

```
for i = [1:len];
  if(peaks2(i)>0.07)
    k2=find(amplitude_lut>(abs(peaks2(i))-tolerance) &
amplitude_lut<(abs(peaks2(i))+tolerance));
    if(k2)
        disp_index2(1,i)=k2(1);
        disp_waveform2(1,i) = displacement_lut(k2(1));
    else
       disp_waveform2(1,i) = 0;
       count=count+1;
    end
  else
      disp_waveform2(1,i) = 0;
  end
end
```

'len' is the length of the peaks array. In the LUT searching algorithm used in this project, a for loop is run to search through the LUT from index 1 to 'len', until the index is found. Low amplitude peaks are removed as they are transient strays. We then use find() function in MATLAB which returns the indices of the 'amplitude_lut' array which satisfies the condition.

Since the 'peaks2()' has floating point values, the find() function cant directly find the proper index. To overcome the problem, we have defined a tolerance variable such that we create a search window with the ith value of peaks2() +- the tolerance, and we expect the find function to return the indices matching in that range. We therefore take the 1st index and look up the same index in the displacement LUT if more indexes are returned. The value which matches with the index is then stored in a new array 'disp_waveform' of length 'len'. This 'disp_waveform' stores the displacement data which is required for the project.

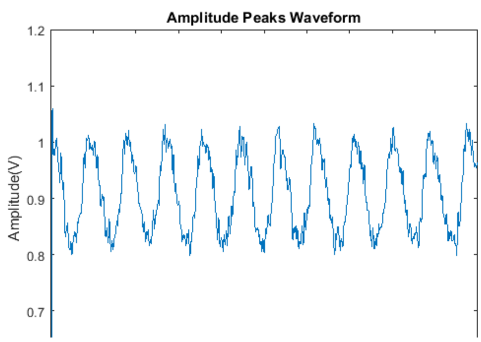


Figure 3.16 Plot showing amplitude waveform of 1000 samples

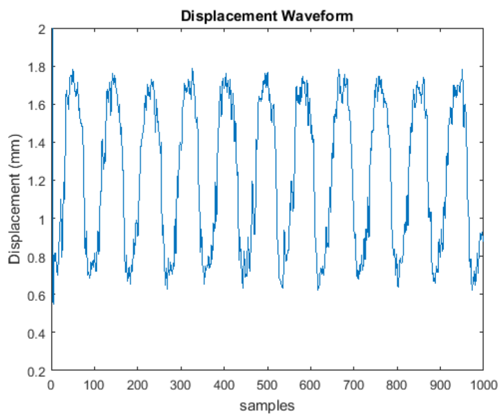


Figure 3.17 Plot showing the displacement waveform after processing through the LUT

3.7 FILTERING THE WAVEFORM

A Savitzky–Golay filter is a digital filter that can be applied to a set of digital data points for the purpose of smoothing the data, that is, to increase the signal-to-noise ratio without greatly distorting the signal. This is achieved, in a process known as convolution, by fitting successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares. When the data points are equally spaced, an analytical solution to the least-squares equations can be found, in the form of a single set of "convolution coefficients" that can be applied to all data sub-sets, to give estimates of the smoothed signal, (or derivatives of the smoothed signal) at the central point of each sub-set [9].

y = sgolayfilt(x,order,framelen) applies a Savitzky-Golay FIR smoothing filter to the data in vector x. If x is a matrix, sgolayfilt operates on each column. The polynomial order, order, must be less than the frame length, framelen, and in turn framelen must be odd. If order = framelen-1, the filter produces no smoothing.

Below is the comparison of the two waveforms, one without filtering on the left, and one after filtering on the right.

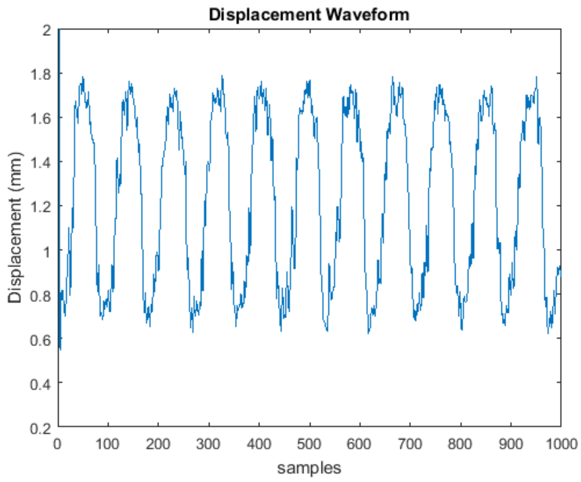


Figure 3.18: Plot showing displacement waveform right after processing through LUT algorithm

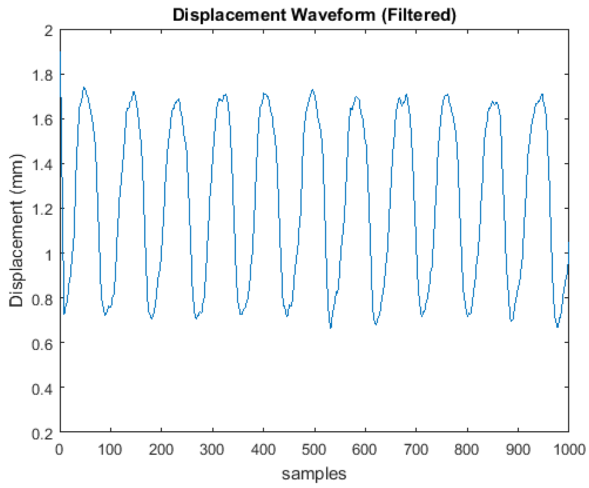


Figure 3.19: Plot of displacement waveform after filtering

CHAPTER 4

4.1 PHASE 3 OF THE PROJECT

It was required to verify the displacement vibration data obtained from the previous experiments and for this purpose, we used the Laser Distance Sensor to measure the string vibration and compare it with previous results.

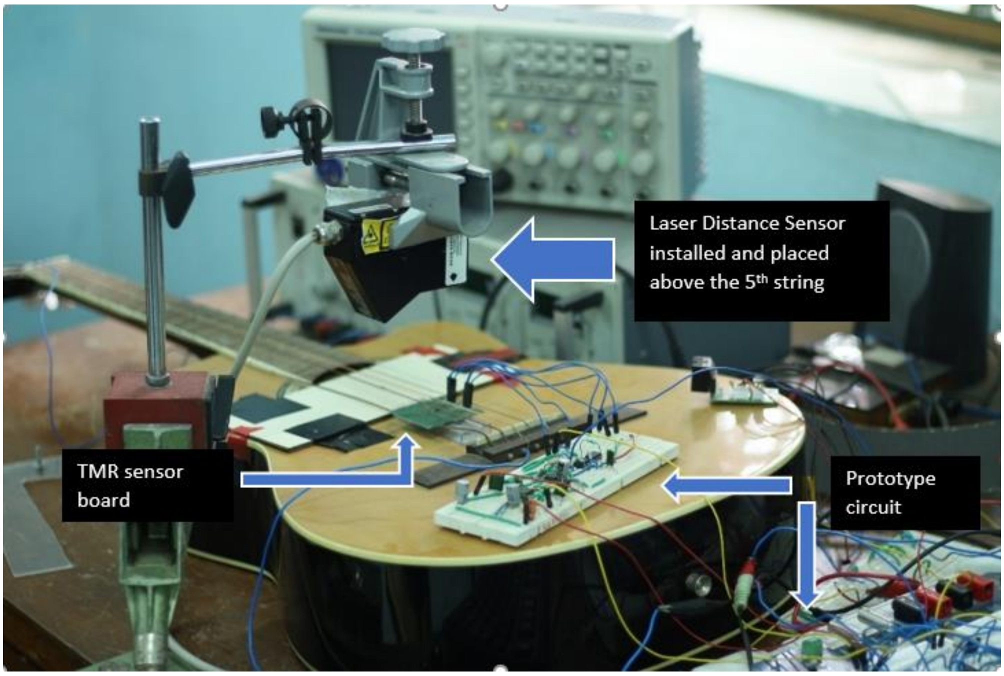


Figure 4.1: Hardware setup of the project along with the Laser Distance Sensor installed

The major highlights of the 3rd phase of the project is:

i. Due to the specifications mentioned in the LDS datasheet, the LDS was placed 8cm above the open ‘A’ string of the guitar.

ii. The LDS had an analog output of 0-10V DC corresponding to 0-2cm distance, therefore the analog output signal from the LDS was connected to the Elvis Board for data acquisition.

iii. With the help of LabVIEW DAQ, we converted the voltage output of the LDS into corresponding distance value in millimetres

iv. Verification of the displacement results obtained in the previous stage.

v. The algorithm from MATLAB was ported to LabVIEW and VI was made which produced audio output from the speakers

4.1.1 LABVIEW INTERFACE

It was in this stage of the project that the final LabVIEW interface was made as shown below:

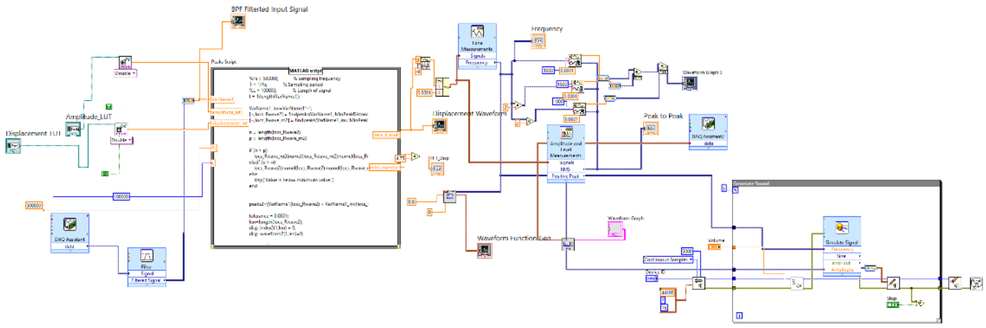
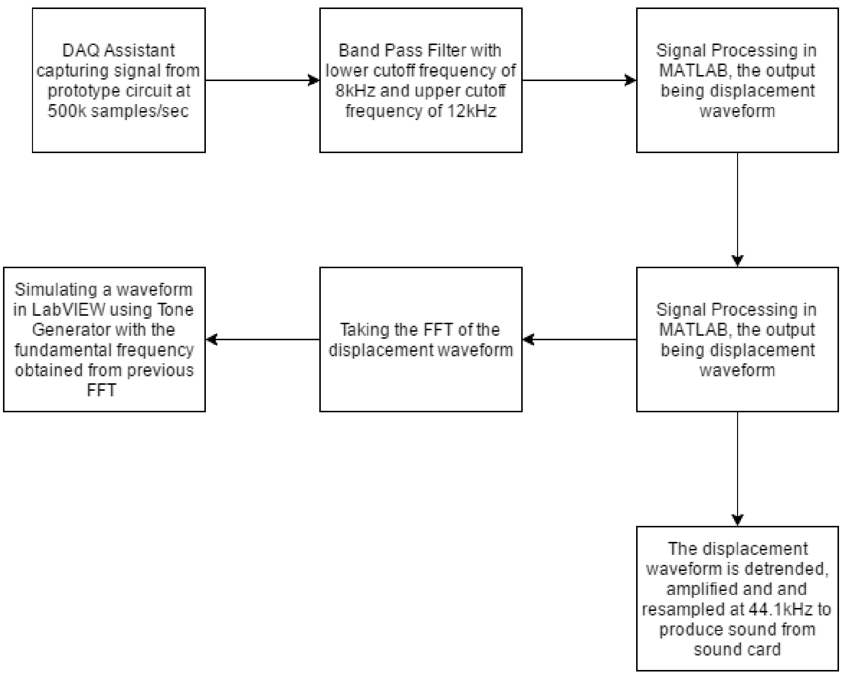


Figure 4.2: LabVIEW interface of the project

The flow chart of the LabVIEW interface in the 3rd phase of the project is shown below:



4.2 VERIFYING WITH LASER DISTANCE SENSOR

With the help of the LUT obtained by the methods stated earlier, we got the displacement vibration data from the amplitude data. Now in order to verify whether the waveform is correct or not, we used a Laser Distance Sensor to directly get the string vibration waveform.

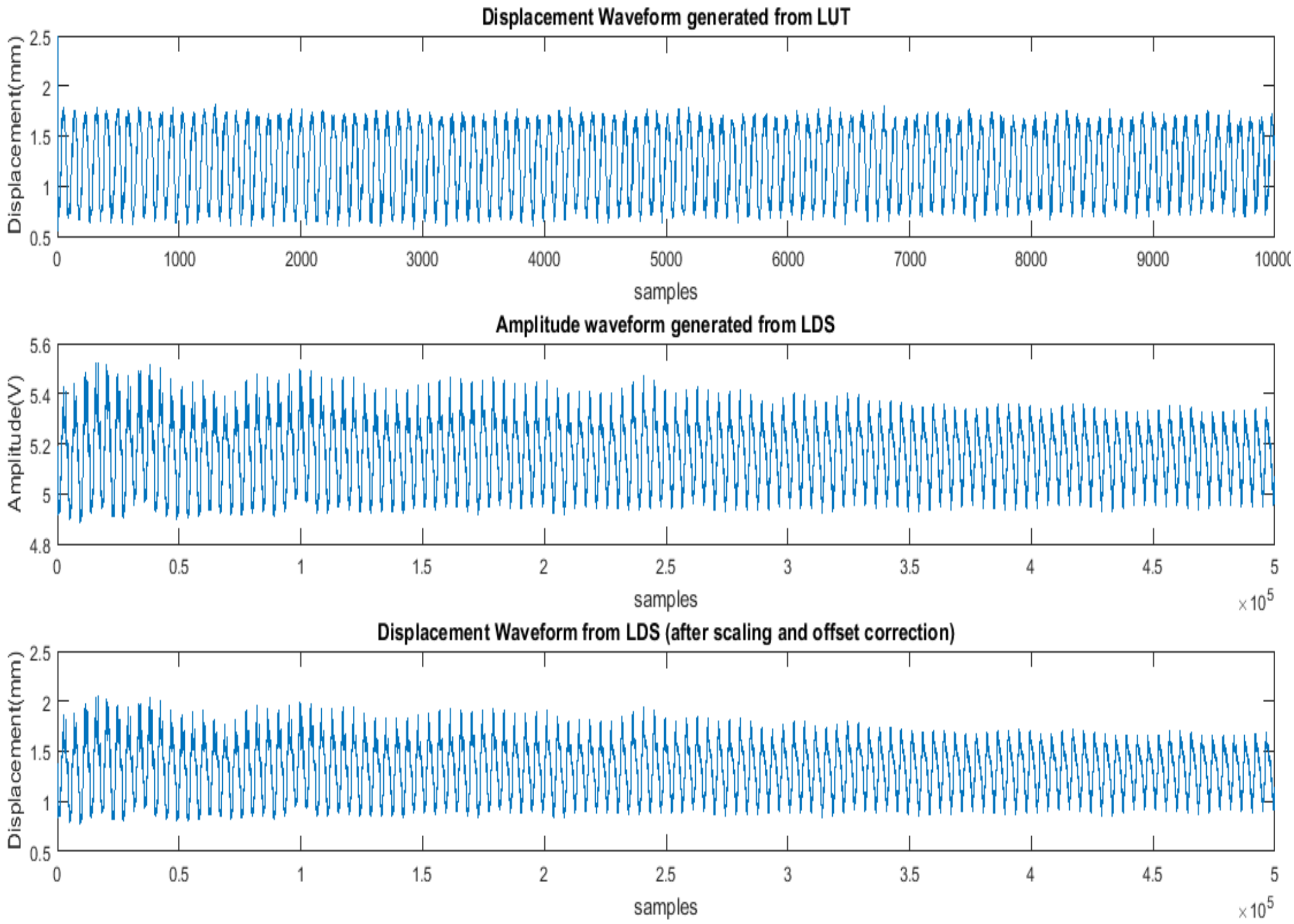


Figure 4.3 The first plot shows the displacement waveform generated from the LUT, the 2nd plot shows the amplitude waveform obtained from the LDS and the 3rd plot shows the displacement waveform from LDS after scaling

The reference sensor that we are using to verify the results which we got from the screw displacement setup is LMI Laser Distance Sensor 80/20. The sensor is based on high speed PSD based triangulation principle, with measurement range of up to 45mm at frequencies up to 100kHz. The model which we used had measurement range of 20 mm with offset distance of 80 mm. It has an analog voltage out from 0-10V corresponding to 0-2cm [10].

The first two plots show the displacement waveform got from the LUT and the string vibration waveform from the LDS sensor respectively. It was noted that the string vibration waveform in the second plot was without any scaling and offset correction and was in the form of direct output voltage from it. As we know from the datasheet of Laser Distance Sensor, that 1V corresponds to 20mm, we first convert the waveform from voltage scale to displacement in millimetre scale, then we account for the initial offset. Finally, we obtain the corrected displacement waveform from the string vibration which is shown in the third plot.

We can compare the first plot and the third plot and find that they are very similar.

4.2.1 FFT ANALYSIS OF THE TWO WAVEFORMS

Now we take the FFT of the waveform obtained from the LUT and the waveform obtained from the Laser Distance Sensor. We can conclude from the FFT spectrum below that the fundamental frequencies are same in both the waveforms.

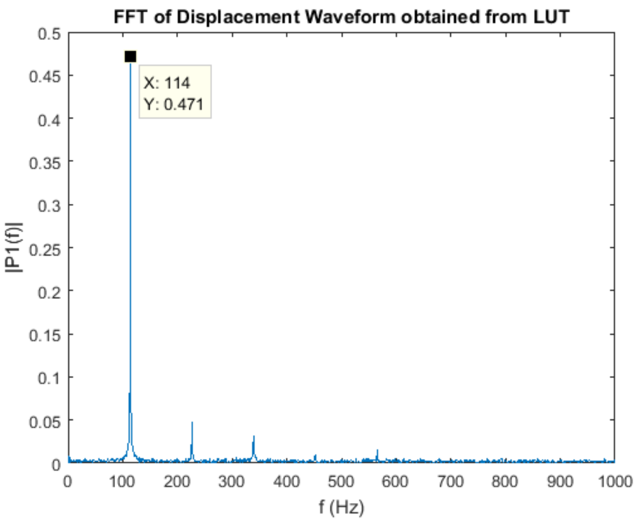


Figure 4.4: FFT spectrum of the displacement waveform

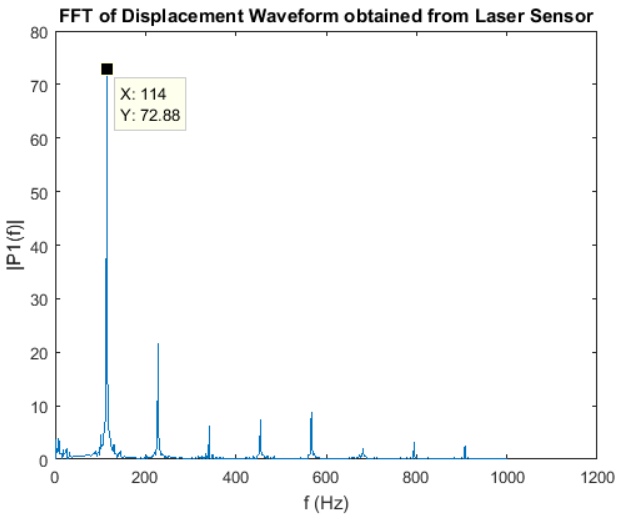


Figure 4.5: FFT spectrum of the displacement waveform obtained from Laser Distance Sensor

4.3 GENERATING LUT WITH LASER DISTANCE SENSOR

After verifying the results with the Laser Distance Sensor, it was decided that a more accurate Look up Table to be made as the displacement readings from the LDS is very reliable than the less reliable displacement data from the screw setup.

The output signal from the LDS was connected to the analog input port of the Elvis board and a Lab VIEW interface was made where the displacement data was captured along with the amplitude data from the TMR prototype circuit. We used the same screw setup for displacing the string, but in this case, we used displacement readings from the LDS.

The figure below shows the amplitude vs displacement plot for various values of current through the string. We chose our standard 100mA current through the string for generating the LUT. After obtaining the array of displacement and amplitude data, we use the same algorithm discussed earlier to generate the LUT.

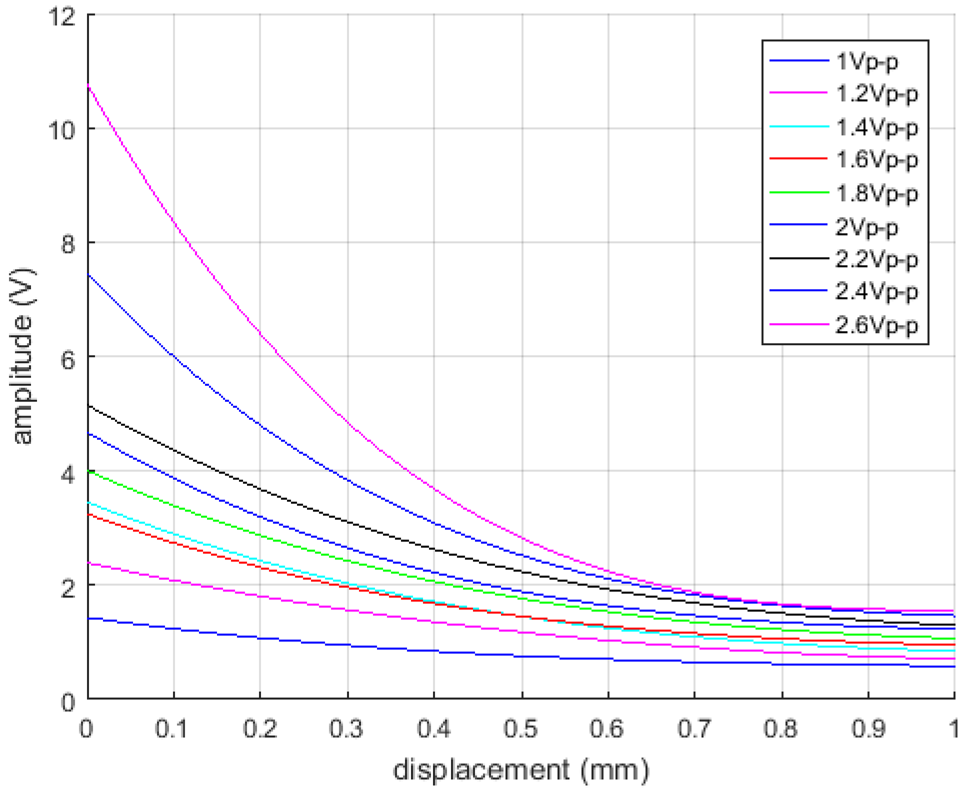


Figure 4.6: amplitude vs displacement data from the laser distance for various current values through the string

4.1 USING HILBERT TRANSFORM TO GENERATE AMPLITUDE WAVEFORM

The Hilbert Transform takes a time domain signal into another time domain signal. We are not changing the domains. This transform is very useful for representing certain kind of signals, representation of bandpass signals. Useful for representing amplitude modulation. It simply phase shifts all frequency components of the input by $-\frac{\pi}{2}$ radians [11].

The Hilbert transform is related to the actual data by a 90-degree phase shift; sines become cosines and vice versa. A real function f(t) and its Hilbert transform f'(t) are related to each other in such a way that they together create a so called strong analytic signal. The strong analytic signal can be written with an amplitude and a phase where the derivative of the phase can be identified as the instantaneous frequency. The analytic signal is useful in the area of communications, particularly in bandpass signal processing. The toolbox function hilbert in MATLAB computes the Hilbert transform for a real input sequence x and returns a complex result of the same length, y = hilbert(x), where the real part of y is the original real data and the imaginary part is the actual Hilbert transform. y is sometimes called the analytic signal, in reference to the continuous-time analytic signal [12].

Envelope Extraction Using the Analytic Signal:

As we have earlier approximated the input signal as an AM waveform, now it becomes very easy to extract out the source signal from the modulated signal by extracting out its envelope. We do this using Hilbert transform.

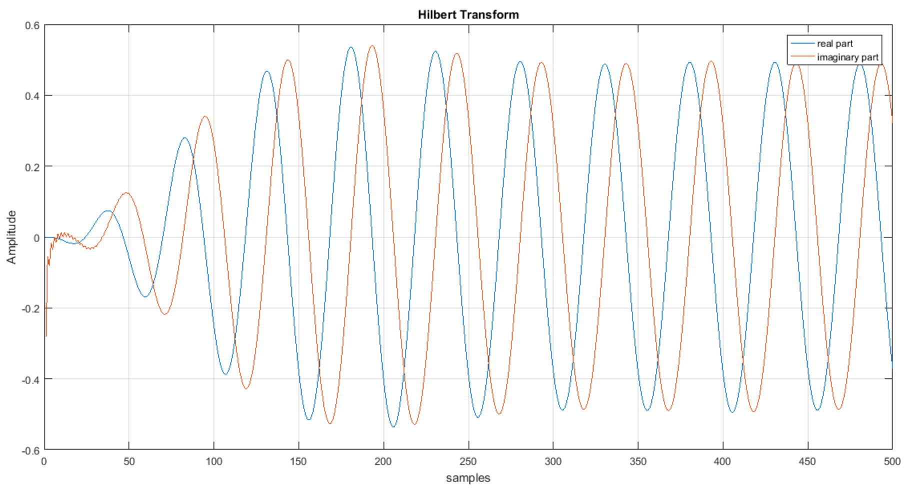


Figure 4.7 Red curve showing the 90-degree phase shifted waveform due to Hilbert transform and Blue curve showing the original waveform

From the above graph, we see that the Hilbert transform(red), which is the imaginary part of the analytic function, and the original signal (blue) is just phase shifted by -90 degrees. To extract out the envelope, we take the magnitude (modulus) of the analytic signal. Below shows the plot for 50,000 samples of the peaks obtained by taking modules of the result y = hilbert(x), where x is the original waveform.

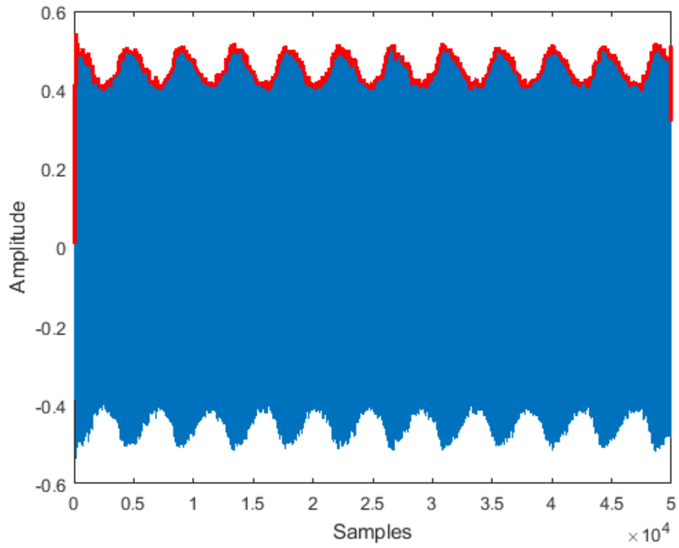


Figure 4.8: Envelope extraction by taking the modules of the analytic function generated by Hilbert Transform

The red curve is then extracted and plotted below as amplitude (V) vs samples which can be used to feed into displacement generation algorithm using LUTs. This method of envelope extraction/peak extraction requires more mathematical calculations than the previous method as discussed earlier, but the earlier peak extraction was based to searching the local maxima, and for that, we need to define a window size beforehand. We can use Hilbert Transform method for covering other strings.

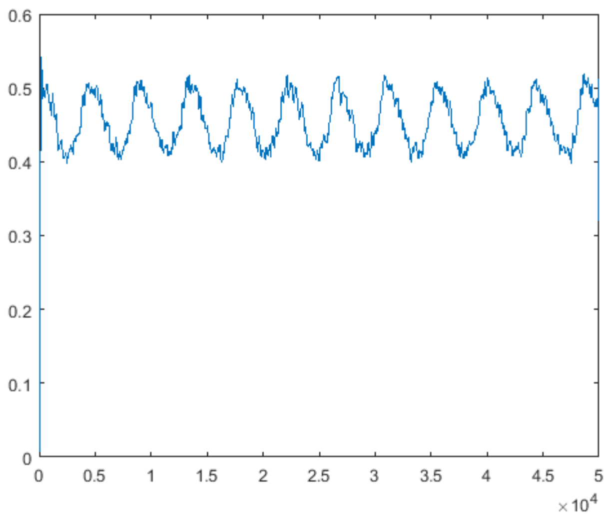


Figure 4.9: Plot showing the extracted amplitude waveform, ready to be sent to LUT algorithm

CHAPTER 5

5.1 EXPERIMENTAL ANALYSIS

A guitar fingerboard is fretted, having raised strips of hard material perpendicular to the strings, which the player presses the strings to produce sounds of different notes. In this experiment, starting from the open string till 9th fret, the string is vibrated and the frequency got from the experiment is then compared to the actual frequency [16] with which I should vibrate and the results obtained are very good.

Table 5.1: This table compares the frequency obtained from the experiment to the actual frequency with which the string should vibrate, for different frets.

Fret	Freq(Hz): Obtained from experiment	Freq (Hz): Actual	% Error
0 (Open String)	110.1	110	-0.090909091
1	117.9	116.5	-1.201716738
2	124.8	123.5	-1.052631579
3	132.05	130.8	-0.955657492
4	140.1	138.6	-1.082251082
5	148.3	146.8	-1.021798365
6	157.3	155.6	-1.092544987
7	166.5	164.8	-1.031553398
8	176.4	174.6	-1.030927835
9	186.3	185	-0.702702703

5.2 CONCLUSION

The main aim of this project was to test the application of TMR sensor in the area of electric guitar pickup and it proved to be successful. The TMR sensor properly detects the frequency of the current through the string and produces a distorted amplitude modulated waveform when the string is vibrated. With the help of signal processing algorithms, we are successfully able to extract out the string vibration waveform, which produces sound very similar to the actual guitar sound.

The results obtained from the project has proved that there is a huge scope and potential in the music industry by using the TMR sensor for electric guitar pickup. The prototype circuit which was designed around the TMR sensor can be easily built and the signal processing algorithms used in the project is also not computationally tough.

5.3 FUTURE WORK

The project has a lot of scope for future work which are described below:

i. The present work is done only for the 5th string or open ‘A’ string of the guitar. The current algorithm can be applied to the other strings, with slight tweaks.
ii. The algorithm can be ported into a low power high performance standalone microprocessor or microcontroller board which will be perfectly suitable for portability.
iii. The signal conditioning circuit may be revisited and can be improved further to increase the signal to noise ratio.
iv. By increasing the signal to noise ratio, the amplitude of the current through the string can be lowered, discarding the use of power amplifier which is currently used to supply current through the string. General purpose opamps can therefore be used to provide current through the string, thereby increasing the portability.

REFERENCES

[1] AAT001-10E TMR Angle Sensor, https://www.nve.com/Downloads/AAT001.pdf

[2] Concepts of Physics: Volume 1, H.C. Verma

[3] Concepts of Physics: Volume 2, H.C. Verma

[4] 3D Magnetic Field Computation of a Straight Wire of Finite Length using Biot-Savart's Law, https://in.mathworks.com/matlabcentral/fileexchange/47368-3d-magnetic-field-computation-of-a-straight-wire-of-finite-length-using-biot-savart-s-law

[5] Active Volume Control for Professional Audio-Texas Instruments, Ian Williams

[6] https://in.mathworks.com/help/curvefit/curvefitting-app.html

[7] Advantages to the Look-up Table Approach of Voltage-Velocity Conversion In Thermal Anemometry http://tsi.com/uploadedFiles/Product_Information/Literature/Technical_Notes/Adv_Look-up_Table_App_Volt-Vel_Conver_TA.pdf

[8] Find local maxima, https://in.mathworks.com/help/signal/ref/findpeaks.html

[9] Savitzky-Golay filtering, https://ai.berkeley.edu/~ee123/sp15/docs/SGFilter.pdf

[10] http://technodata.com/PDF/LMI/LDS_060200[1].pdf

[11] Discrete-time analytic signal using Hilbert transform https://in.mathworks.com/help/signal/ref/hilbert.html

[12] Envelope Extraction Using the Analytic Signal http://in.mathworks.com/help/signal/ug/envelope-extraction-using-the-analytic-signal.html

[13] http://www.ti.com/lit/ds/symlink/ina129.pdf

[14] www.ti.com/lit/gpn/opa4134

[15] www.ti.com/lit/ds/symlink/opa541.pdf

[16] Tensions of Guitar Strings, Daryl Achilles https://courses.physics.illinois.edu/phys406/Student_Projects/Fall00/DAchilles/Guitar_String_Tension_Experiment.pdf

APPENDIX-A

Matlab code for peak extraction from the input waveform:

```
Fs = 500000;              % Sampling frequency
T = 1/Fs;               % Sampling period
L = 10000;               % Length of signal
t = 1:length(VarName1);

VarName1_inv=VarName1*-1;
[~,locs_Rwave2] = findpeaks(VarName1,'MinPeakDistance',40);
[~,locs_Rwave_m2] = findpeaks(VarName1_inv,'MinPeakDistance',40);

n = length(locs_Rwave2)
p = length(locs_Rwave_m2)

if (n > p)
    locs_Rwave_m2(numel(locs_Rwave_m2):numel(locs_Rwave2)) =
locs_Rwave_m2(p);
elseif (p > n)
    locs_Rwave2(numel(locs_Rwave2):numel(locs_Rwave_m2)) = locs_Rwave2(n);
else
    %disp('Value is below minimum value.')
end


peaks2=(VarName1(locs_Rwave2) + VarName1_inv(locs_Rwave_m2));

tolerance = 0.0001;
len=length(locs_Rwave2);
disp_index2(1,len) = 0;
disp_waveform2(1,len)=0;
count = 0;
for i = [1:len];
  if(peaks2(i)>0.07)
    k2=find(amplitude_lut>(abs(peaks2(i))-tolerance) &
amplitude_lut<(abs(peaks2(i))+tolerance));
    if(k2)
        disp_index2(1,i)=k2(1);
        disp_waveform2(1,i) = displacement_lut(k2(1));
    else
       disp_waveform2(1,i) = 0;
       count=count+1;
    end
  else
      disp_waveform2(1,i) = 0;
  end
end

[~,locs_Rwave4] =
findpeaks(disp_waveform2,'MinPeakDistance',50,'MinPeakHeight',1.4);

Fs_disp = length(locs_Rwave2);
T_disp = 1/Fs_disp;              % Sampling period
L_disp = length(locs_Rwave2);
t_disp = (0:L_disp-1)*T_disp;         % Time vector

X=disp_waveform2;
```

```
Y = fft(X);

P2 = abs(Y/L_disp);
P1 = P2(1:L_disp/2+1);
P1(2:end-1) = 2*P1(2:end-1);

%f _disp= Fs_disp*(0:(L_disp/2))/L_disp;
P1(1)=0;
%plot(P1);
[max_value, index_number] = max(P1);
smtlb = sgolayfilt(disp_waveform2,1,11);
data_d = detrend(smtlb);
len_data=length(data_d);
data_d_mult = data_d * 50;
data_d_mult(1:9)=0;
plot(data_d_mult);
```

Matlab code for simulation of standing waves:

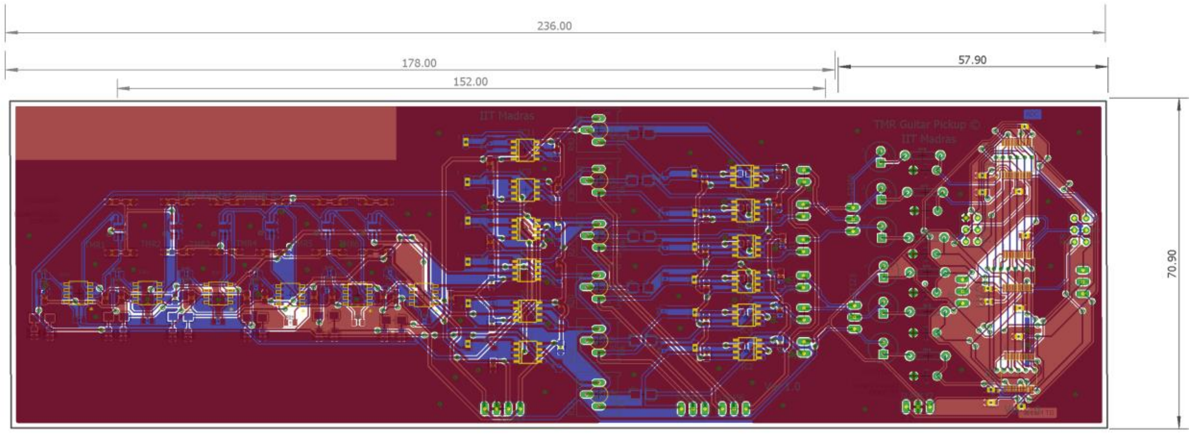
```
figure('color','white')
x=0:0.01:10;
A=1;
L=0.63;
for t=0:0.1:5
   y=2*A*sin((pi*x)/L)*cos(2*pi*110*t);
   plot(x,y,'b');
   hold all;
   ylim([-2 2]);
   title('Standing Wave')
   getframe;
end
```

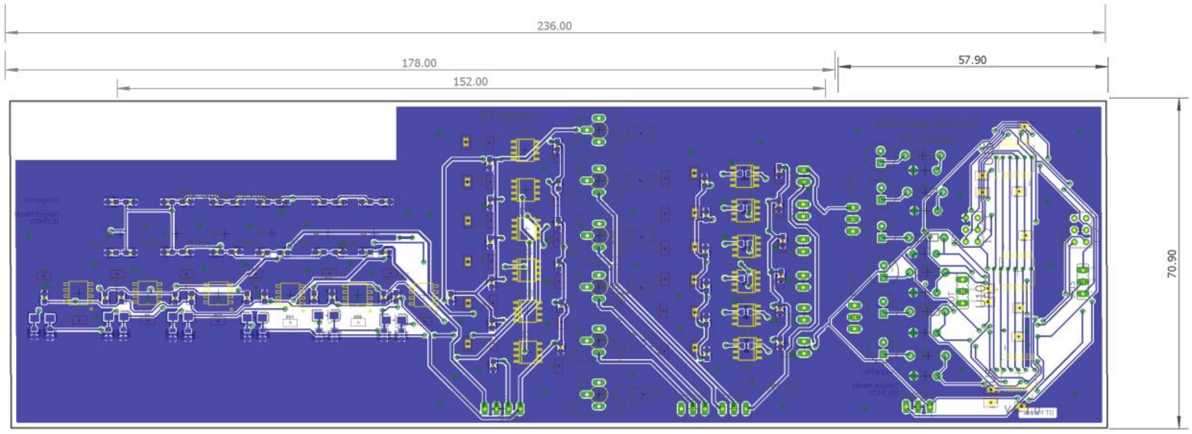
Matlab code for Hilbert transform:

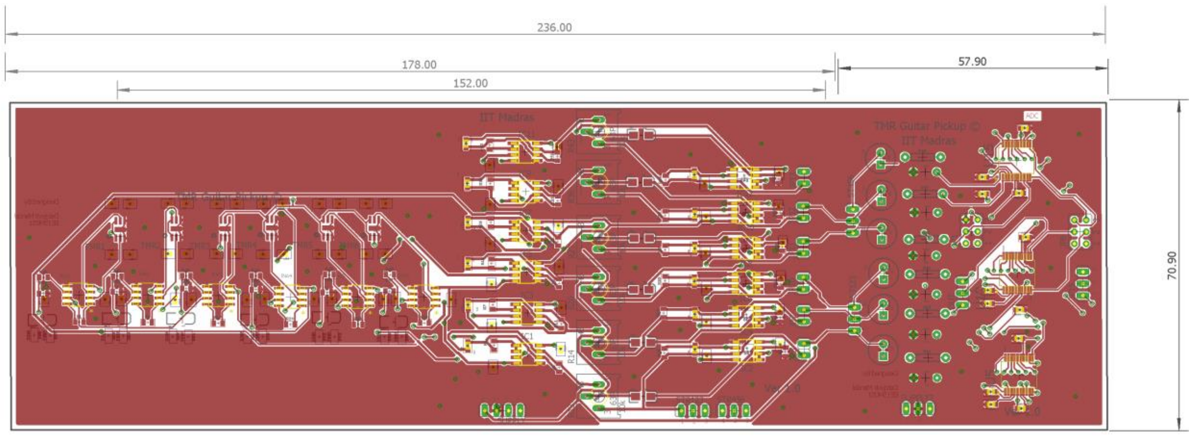
```
%t = 0:1e-4:1;
t = 1:50000;
%x = [1+cos(2*pi*50*t)].*cos(2*pi*1000*t);
x=VarName1(1:50000);

plot(t,x)
%xlim([0 0.1])
xlabel('Samples')
ylabel('Amplitude')

y = hilbert(x);
env = abs(y);

plot(t,x)
hold on
plot(t,env,'r','LineWidth',2)
%xlim([0 0.1])
xlabel('Samples')
ylabel('Amplitude')
```

APPENDIX-B

After the results from the prototype circuit, followed by signal processing was correct, PCB was designed in EAGLE. The figure below shows the complete two-layer PCB, the bottom layer in blue and the top layer in red.







CURRICULUM VITAE

Name :	Debjyoti Kumar Mandal
Date of Birth :	25th January, 1992
Education :	B.Tech in Electronics and Instrumentation Engineering from RCC Institute of Information Technology, Kolkata [2010-2014] M.Tech in Electrical Engineering, Control and Instrumentation from Indian Institute of Technology Madras, [2015-2017]
E-mail :	danieltombay@gmail.com
Permanent Address:	BH-17, 1st Floor, Sec-2, Salt Lake, Kolkata-700091, WB