

**DESIGN AND DEVELOPMENT OF  
A SIGNAL CONDITIONING CIRCUIT FOR  
INDUCTIVE SENSORS**

*A THESIS*

*submitted by*

**PIYUSH KUMAR  
(EE12M096)**

*for the award of the degree*

*of*

**MASTER OF TECHNOLOGY**

*in*

**ELECTRICAL ENGINEERING**



**DEPARTMENT OF ELECTRICAL ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY MADRAS**

**MAY 2014**

# **DESIGN AND DEVELOPMENT OF A SIGNAL CONDITIONING CIRCUIT FOR INDUCTIVE SENSORS**

*A THESIS*

*submitted by*

**PIYUSH KUMAR  
(EE12M096)**

*for the award of the degree*

*of*

**MASTER OF TECHNOLOGY**

*in*

**ELECTRICAL ENGINEERING**



**DEPARTMENT OF ELECTRICAL ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY MADRAS**

**MAY 2014**

## CERTIFICATE

This is to certify that the thesis titled **DESIGN AND DEVELOPMENT OF A SIGNAL CONDITIONING CIRCUIT FOR INDUCTIVE SENSORS**, submitted by **Piyush Kumar**, to the **Indian Institute of Technology Madras**, Chennai in partial fulfilment of the requirements for the award of the Degree of **Master of Technology** (Control and Instrumentation) in Electrical Engineering, is a bona fide record of work done by him, under my supervision and guidance. The contents of this project thesis, in full or in parts, have not been submitted to any other University/Institute for the award of any Diploma or Degree.

Dr. Bobby George

(Project Guide)

Assistant Professor

Dept. of Electrical Engineering,

IIT-Madras, 600 036

Place: Chennai

Date:

## ACKNOWLEDGEMENTS

Foremost, I would like to express my deep sense of gratitude to my guide Dr. Bobby George for his erudite guidance, support, encouragement and generous help throughout the period of the project work. I am highly indebted to him for devoting his valuable time to help me execute this work in good time. I must not forget to mention that he had instant solution to every single problem I faced during the project work.

I am also extremely grateful to Prof Jagadeesh Kumar, V., Head, Measurement and Instrumentation Laboratory, IIT Madras for allowing me to have access to all the lab facilities that I needed from time to time for successful completion of the project. Also, his brilliant comments and suggestions during project progress review sessions proved priceless to me.

I also thank all the teaching and non-teaching staff, Shri. Umaithanu Pillai, in particular, of the Electrical Department for extending all the support and cooperation to me throughout.

I also avail this opportunity to express my gratitude to “Indian Air Force” for giving me the opportunity to be part of the prestigious institute like IIT Madras.

Special thanks to my lab mates and friends Sachin, Danish, Gaurav, Ritesh, Anish and Arjun for the informative discussions and memorable times in the lab.

Also, words cannot express how grateful I am to my beloved parents, wife Mamta and sons Naman and Rishit for their unconditional support at home.

Last but not the least, I must not forget to thank Almighty for the wisdom and perseverance that He bestowed upon me during this project work, and indeed, throughout my life.

Piyush Kumar

## ABSTRACT

**KEYWORDS:** Inductive sensor, proximity sensor, front-end circuit, offset correction.

Signal conditioning of inductive sensors so as to obtain an output proportional to just the change in the inductance alone is fraught with problems. The large value of self inductance that is present in a sensor coil and the change in the inductance being a small fraction of this large inductance coupled with the winding resistance of the sensor coil make signal conditioning of such inductive sensors a challenge. A simple analog front-end circuit suitable for signal conditioning of inductive sensors is presented. The proposed signal conditioning circuit provides an output linearly related to the change in inductance due to the measurand alone, masking the large value of self (offset) inductance present in inductive sensors as well as the appreciable winding resistance. A prototype of the proposed signal conditioning circuit has been developed and tested in the laboratory. Test results validate the efficacy of the technique presented herein.

The circuit may find wide range of applications in sensor world. Apart from their expected use in Inductive Proximity Sensor (IPS) applications, the circuit is especially suited for linear displacement sensors due to highly linear and accurate characteristics. Microcontroller based automatic calibration introduced at the start-up of the circuit ensures that changes caused due to environmental factors like temperature, humidity etc do not contribute to any errors in measurement.

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS</b>	i
<b>ABSTRACT</b>	ii
<b>TABLE OF CONTENTS</b>	iii
<b>LIST OF TABLES</b>	v
<b>LIST OF FIGURES</b>	vi
<b>ABBREVIATIONS</b>	viii
<b>NOTATIONS</b>	ix
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Inductive Sensors .....	1
1.2 Objective and Scope of the Project .....	3
1.3 Organisation of the Thesis .....	4
<b>2. DESIGN AND DEVELOPMENT OF THE FRONT-END CIRCUIT</b>	<b>5</b>
2.1 Block Diagram of the Front-end Circuit .....	5
2.2 Proposed Circuit.....	6
2.3 Generalized Impedance Converter (GIC).....	7
2.4 Instrumentation Amplifier.....	9
2.5 Phase Sensitive Detector (PSD).....	10
2.5.1. Mixer.....	10
2.5.2. Low Pass Filter.....	11
2.6 Complete Circuit.....	13
<b>3. CIRCUIT CALIBRATION</b>	<b>15</b>
3.1 Digital Potentiometer IC MCP4261.....	16
3.1.1 Resistor Network.....	16
3.1.2 Volatile and Non-Volatile (EEPROM) Memory Registers.....	17
3.1.3 Serial Interface (SPI).....	18
3.1.4 SPI Modes.....	20
3.1.5 Device Commands.....	20
3.2 AVR Series Microcontroller ATmega8.....	21
3.3 Calibration Process.....	23

3.3.1	Adjustment of $R_f$ .....	23
3.3.2	Adjustment of $R_7$ .....	23
3.3.3	Storage of Wiper Positions.....	24
3.4	Using the ADC.....	24
3.5	Generation of Sine Wave Excitation.....	27
3.6	16x2 LCD.....	28
<b>4.</b>	<b>EXPERIMENTAL SETUP AND RESULTS</b>	<b>30</b>
4.1	Virtual Instrument.....	30
4.2	Prototype of the Circuit-Stage I.....	31
4.3	Prototype of the Circuit-Stage II.....	31
4.4	PCB Design and Fabrication.....	32
4.5	Annealed/Non-annealed Metal Detection.....	33
4.6	Proximity Sensing.....	34
4.7	Accuracy and Linearity.....	35
<b>5.</b>	<b>CONCLUSION</b>	<b>37</b>
5.2.1.	Summary of the Work.....	37
5.2.2.	Future Scope.....	37
	<b>REFERENCES</b>	<b>39</b>
	<b>APPENDIX-A</b>	<b>41</b>
	<b>APPENDIX-B</b>	<b>52</b>
	<b>CURRICULUM VITAE</b>	<b>53</b>

## LIST OF TABLES

1.1:	Traditional methods of sensor coil inductance measurements .....	2
3.1:	Memory Map of MCP4261 registers.....	18
3.2:	Description of the pins of 16x2 LCD module.....	29

## LIST OF FIGURES

1.1:	Industrial applications of inductive sensors.....	1
2.1	Block diagram of the front-end circuit.....	5
2.2	Basic Schematic of the circuit.....	6
2.3	Electronic impedance circuit used to realize a variable inductance $L_r$ .....	7
2.4	Bode plot obtained for the electronic inductance shown in Fig 2.3.....	8
2.5	Basic schematic of an instrumentation amplifier.....	9
2.6	Mixer stage of the Phase Synchronous Demodulator.....	10
2.7	Frequency spectrum of Mixer output.....	11
2.8	Low-pass filter stage of the Phase Synchronous Demodulator.....	12
2.9	Transfer characteristics of Butterworth filter as simulated in TINA-TI.....	12
2.10	Complete circuit after integration of GIC and PSD.....	14
3.1	Interconnections between ATmega8 and DigiPot IC MCP4261 .....	15
3.2	Resistor network structure of MCP4261.....	17
3.3	Typical SPI interface connections.....	19
3.4	Formats of MCP4261 supported SPI commands.....	21
3.5	ATmega8 pin diagram.....	22
3.6	Auto-calibration process.....	24
3.7	Mapping of -5V to +5V range onto 0 to +5V range.....	25
3.8	Flow chart of microcontroller program for calibration of the circuit.....	26
3.9	Generation of sinusoidal excitation signal using XR2206.....	27

3.10	Pin diagram of a 16x2 LCD module.....	28
4.1	Front panel of the VI developed in LabVIEW.....	31
4.2	Experimental setup.....	32
4.3	Components mounted on double layer PCB.....	33
4.4	Annealed/unannealed metal detection.....	34
4.5	Proximity sensing.....	35
4.6	Output and error characteristics of the circuit.....	36

## ABBREVIATIONS

ADC	Analog-to-Digital Converter
AVR	Alf (Egil Bogen) and Vegard (Wollan)'s RISC processor
CGIC	Current Generalized Impedance Converter
CMDERR	Command Error
CMRR	Common-Mode Rejection Ratio
CS	Chip Select
EEPROM	Electrically Erasable Programmable Read-Only Memory
ELVIS	Educational Laboratory Virtual Instrumentation Suite
GIC	Generalized Impedance Converter
GPIO	General Purpose Input/Output
INA	Instrumentation Amplifier
IPS	Inductive Proximity Sensor
LCD	Liquid Crystal Display
PSD	Phase Sensitive Detector
RAM	Random Access Memory
SCK	Serial Clock
SDI	Serial Data In
SDO	Serial Data Out
SPDT	Single Pole Dual Throw
SPICE	Simulated Program with Integrated Circuit Emphasis
TCON	Terminal Control
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
VGIC	Voltage Generalized Impedance Converter
VI	Virtual Instrument

## NOTATIONS

V	Voltage
I	Current
R	Resistance
C	Capacitance
L	Inductance
Z	Impedance
f	Frequency
$\mu$	Micro
m	milli
k	kilo
M	Mega
Hz	Hertz
$\Omega$	Ohm
$\omega$	Angular frequency

# CHAPTER 1

## INTRODUCTION

### 1.1 INDUCTIVE SENSORS

Inductive sensors based on change in the self-inductance of a sensing coil as a function of change in a measurand are widely used in industry. Inductive sensors provide non-contact measurement and hence require very low maintenance, and operate under extreme environmental conditions [1]. Inductive sensors have distinct advantage over optical [2] and capacitive sensors [3], [4] in industrial applications due to their ability to tolerate oil, water and dirt ingress.

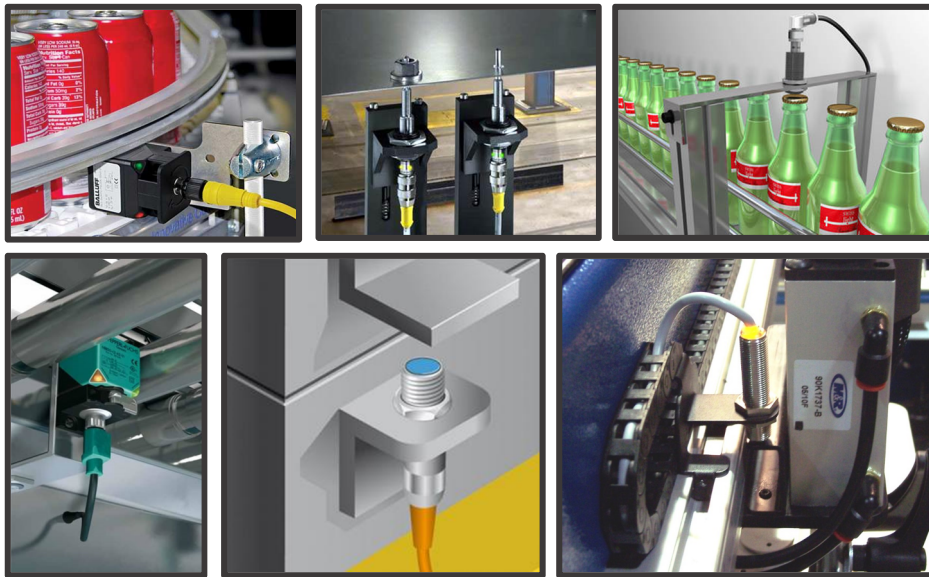


Figure 1.1: Industrial applications of inductive sensors [5][6]

The coil of an inductive sensor will possess fixed (offset) self-inductance and winding resistance. Moreover, the maximum change in inductance due to change in the measurand will be small compared to the offset inductance. In such a condition, bridge based measurement is an excellent choice. Traditional AC Bridges like Maxwell's Inductance Bridge, Maxwell-Wien Bridge, Anderson Bridge, Hay's Bridge, etc. can be used to determine the value of the sensor coil inductance by comparing it with a known impedance under balanced conditions. While balancing a bridge for each and every

measurement provides a linear output characteristic, balancing an ac bridge is tedious and time consuming. An alternate way is to balance the bridge once with the input to the sensor coil set at zero and take the output of the bridge (due to an unbalance created by the small change in the sensor's inductance proportional to a change in the measurand) to represent the input (measurand) to the sensor coil. Such an arrangement though solves the problem of balancing the bridge at each and every measurement instant but results in a nonlinear input-output characteristic. Thus, these types of bridges are not suited for linear inductive sensors [7]. Some bridges employ automatic balancing schemes [8], but either take a long time to settle or need complex arrangements. Inductive sensors that employ a change in the quality factor in a resonant circuit caused by eddy current losses in conductive materials are more problematic in this aspect [9]. The most common application of inductive sensors is found in Inductive Proximity Sensors (IPS). IPSs generate an output signal when metallic objects enter into their sensing area, from any direction.

Linear inductive displacement sensors have also been developed [10]. Such a sensor operates without any contact between the moving and fixed parts of the sensor. Normally a dedicated metallic part is attached to the target and the position of the metallic part (and hence the target) is ascertained by a coil that encompasses the metallic part. A patented signal conditioning method that describes the details of a ring oscillator based linearization mechanism has been reported for such a sensor [11]. As evident, in this method, an output is obtained from the inductive sensor by utilizing the change in the sensor coil inductance corresponding to change in the measurand. Table 1.1 below shows the main disadvantages in employing traditional methods of sensor coil inductance measurements.

Table 1.1: Traditional methods of sensor coil inductance measurements

Measurement Techniques	Limitations
Bridge Circuits	Tedious balancing process. Non-linear output for unbalanced conditions even for sensors with linear displacement vs inductance characteristics.
Inductance to freq converter	Output dependent on $(L_s + \Delta L_s)$
Ring Oscillator based	Output dependent on $(L_s + \Delta L_s)$

In order to obtain an inductive sensor, a coil made of a conductive wire must be employed. Such a sensor coil will have a nominal inductance when the measurand is zero. The inductance of a sensor coil when the input to the sensor is zero is considered as an offset inductance. Apart from the offset inductance, the sensor coil will also possess a winding resistance. Conventional signal conditioning circuits provide an output in which a large part of the output is reserved for the offset inductance of the sensor. The winding resistance will also have a contribution to it and special care is required to remove its effect in the measurement [12]. Finally, the most important portion, the output corresponding to change in inductance owing to change in the measurand will be very small. Typically the variation in coil inductance is in the range 5% to 10% of the offset inductance [9]. It will be very useful, if a signal conditioning circuit that can provide an output as a function of only the change in inductance of the sensor coil, eliminating the influence of the offset inductance and winding resistance can be developed. Such a front-end circuit suitable for capacitive sensors with large offset capacitance has been presented [13].

The developed circuit provides an output with respect to the change in inductance  $\Delta L_s$  of the sensor coil alone and not affected by the presence of a large valued offset inductance and appreciable winding resistance. The circuit presented has all the advantages of a bridge based measurement and provides an output that is directly proportional to  $\Delta L_s$ , unlike the output of an unbalanced bridge. The circuit uses a Generalized Impedance Convector (GIC) to realize a voltage controlled variable inductor and a resistor, which are then used to balance the circuit once, when measurand is zero. As the output has no fixed (offset) voltage due to offset inductance, suitable gain can be introduced and the full output-swing of the measurement system can be exploited to obtain a desired range.

## **1.2 OBJECTIVE AND SCOPE OF THE PROJECT**

The objective of the project is to design a front-end circuit that generates output proportional only to the change in inductance with no output factor owing to offset inductance and coil resistance.

The circuit can be interfaced to work with a wide range of industrial and non-industrial applications involving inductive sensors. The microcontroller used in the circuit can be

programmed to give the output in desired form suitable for both, proximity as well as displacement sensor applications.

### **1.3 ORGANISATION OF THE THESIS**

The Chapter-1 gives an introduction to inductive sensors, their types and applications in industrial sector. It also elaborates the importance of having a linear inductive displacement sensor with output dependent only on the measurand.

The Chapter-2 discusses, in detail, design and development of the front-end circuit. Step-by-step progress of the project has been elaborated in this chapter supported with the related figures and graphs, as required.

Two-stage circuit calibration process has been elaborated upon in Chapter-3. The chapter also discusses the sinusoidal excitation generation followed by the details of LCD display module. Flow chart of the program written for the auto calibration of the circuit is also given in the chapter.

Chapter-4 explains the Virtual Instrument developed for the circuit in LabVIEW environment. Details of the circuit prototypes developed in the lab along with the experimental results are also discussed.

In Chapter-5, conclusions have been drawn on the present work and scope of the future work has been presented.

## CHAPTER 2

### DESIGN AND DEVELOPMENT OF THE FRONT-END CIRCUIT

#### 2.1 BLOCK DIAGRAM OF THE FRONT-END CIRCUIT

The basic block diagram of the developed front-end circuit is as shown in Fig. 2.1. Output from the first stage depends on the difference between the sensor coil inductance and the reference inductor. This output is then amplified to a desired level by an amplification stage. Output from this stage is then passed through a Mixer and Low Pass Filter to remove the unwanted frequencies. The final output from Low Pass Filter is given to the controller for circuit calibration and display purpose.

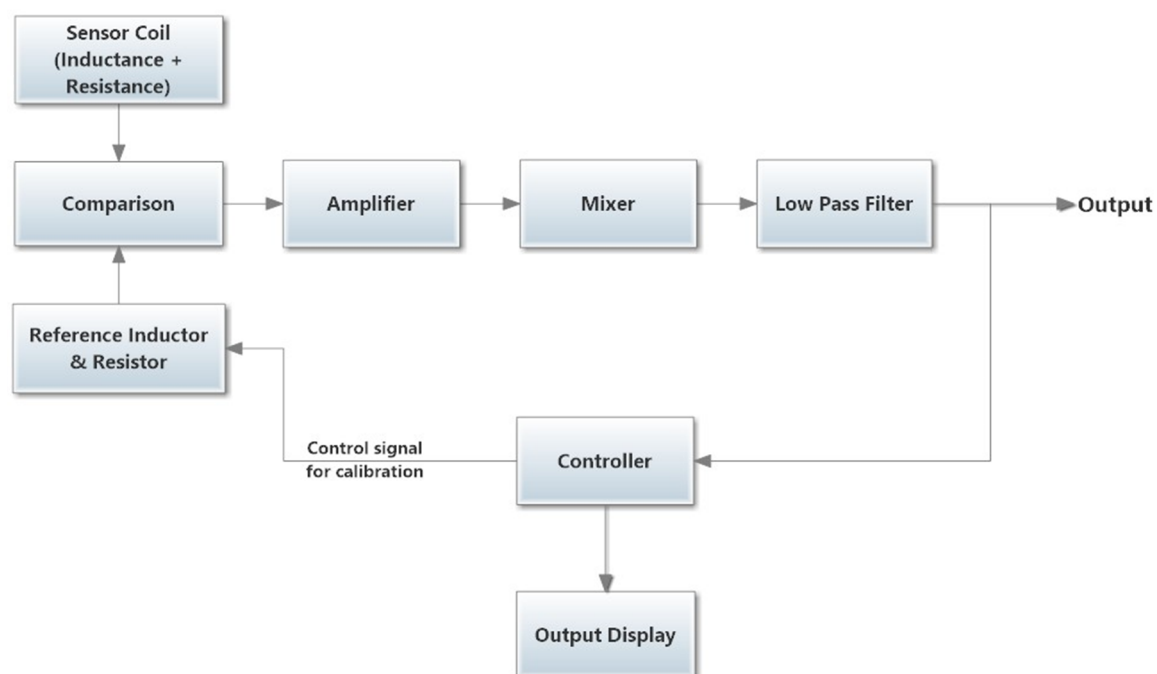


Figure 2.1: Block diagram of the front-end circuit

## 2.2 PROPOSED CIRCUIT

Fig. 2.2 shows the basic schematic of the developed signal conditioning scheme suitable for an inductive sensor using an op-amp  $OA_1$  and an Instrumentation Amplifier (INA). The sensor inductor ( $L_s \pm \Delta L_s$ ) represents the sensor inductance having an offset inductance  $L_s$  and a variable inductance  $\Delta L_s$ .  $L_s$  remains fixed for a selected sensor coil and  $\Delta L_s$  changes with the measured parameter such as displacement, proximity, etc.  $R_s$  represents the sensor coil/winding resistance that remains fixed for a given proximity sensor. The circuit uses, in the feedback path, a variable inductor  $L_r$  with adjustable resistance  $R_r$ .

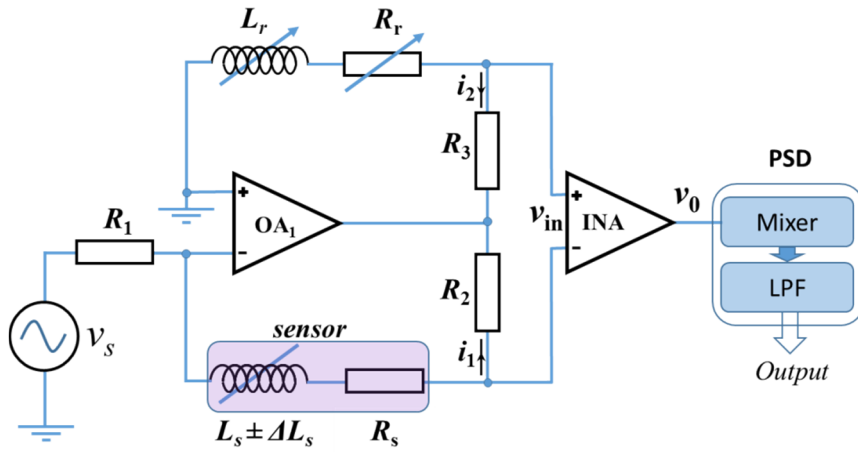


Figure 2.2: Basic schematic of the circuit

For a sinusoidal voltage source  $v_s = V_m \sin \omega t$ , with amplitude  $V_m$  and angular frequency  $\omega$ , constant current that flows through the sensor coil ( $L_s \pm \Delta L_s$ ) and  $R_s$  is given by (2.1) as follows.

$$\bar{I}_1 = \frac{\bar{V}_s}{R_1} \quad (2.1)$$

Current  $\bar{I}_2$ , that passes through  $L_r$ ,  $R_r$  and  $R_3$  can be obtained as in (2.2).

$$\bar{I}_2 = \frac{\bar{I}_1 [R_s + R_2 + j\omega(L_s \pm \Delta L_s)]}{(R_3 + R_r + j\omega L_r)} \quad (2.2)$$

Voltage signals fed to the INA input is given by  $\bar{V}_{in} = \bar{I}_2 R_3 - \bar{I}_1 R_2$ . Substituting the values of  $\bar{I}_1$  and  $\bar{I}_2$  from (2.1) and (2.2),  $\bar{V}_{in}$  can be written as in (2.3).

$$\vec{V}_{in} = \vec{I}_1 \left[ \frac{R_s + R_2 + j\omega(L_s \pm \Delta L_s)}{(R_3 + R_r + j\omega L_r)} R_3 - R_2 \right] \quad (2.3)$$

If we choose  $R_3 = R_2$ ,  $R_r = R_s$  and  $L_r = L_s$ , expression for INA output  $\vec{V}_0 = A\vec{V}_{in}$  can be obtained as follows.

$$\vec{V}_0 = (A\vec{V}_s k_z) \Delta L_s, \quad (2.4)$$

where,  $k_z = \pm j\omega R_2 / (R_1(R_2 + R_r + j\omega L_r))$ . It is a constant for given values of  $\omega$ ,  $R_1$ ,  $R_2$ ,  $R_r$  and  $L_r$ . As can be seen from (4), the output of INA is directly proportional to  $\Delta L_s$ . Once we set  $R_r = R_s$  and  $L_r = L_s$ , output will be minimum (ideally zero) when measurand is zero. No offset output voltage will be present owing to  $L_s$  and  $R_s$ . Output of INA is then given to a Mixer and Low Pass Filter (LPF) to perform synchronous demodulation [14]. INA gain can be suitably adjusted to achieve the desired output level for a given application.

### 2.3 GENERALIZED IMPEDANCE CONVERTER (GIC)

The Generalized Impedance Converter (GIC) is an active two port network in which the input impedance is equal to the load impedance times a conversion function of the complex frequency variable. There are two types of the GIC, the first is the Voltage Generalized Impedance Converter (VGIC) and the second is the Current Generalized Impedance Converter (CGIC) [15]. A GIC, which makes use of two op-amps  $OA_2$  and  $OA_3$ , a capacitor and resistors as shown in Fig. 2.3, is nothing but an inductor without any

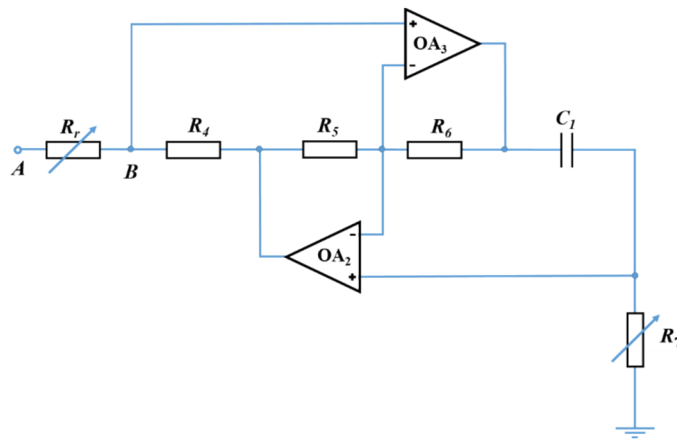


Figure 2.3: Electronic impedance circuit used to realize a variable inductance  $L_r$ . A variable resistance  $R_r$  is added in series to compensate for the sensor coil resistance  $R_s$ .

turns of wire. A GIC makes a capacitor behave like an inductor and thus makes it an obvious choice for the proposed signal conditioning circuit due to the following advantages it has over traditional inductors:-

- Smaller size
- Low cost
- Immunity to ambient magnetic fields
- Inductance can simply be changed by varying a resistor value

Similar GIC based circuit has been used to realize  $L_r$  and  $R_r$  in the circuit. Resistor  $R_7$  is chosen to be variable for adjusting the value of  $L_r$  equal to  $L_s$  and  $R_r$  can be varied to make it equal to sensor coil resistance  $R_s$ . The input impedance  $Z_{in}$  thus obtained can be expressed as in (2.5)

$$Z_{in} = R_r + j\omega L_r, \quad (2.5)$$

where,  $L_r$  is given by

$$L_r = R_4 R_6 R_7 C_1 / R_5. \quad (2.6)$$

Functionality of the prototype GIC based  $L_r$  was tested by taking a bode plot of the same, using ELVIS-II. This was compared with the Bode plot obtained by simulating the same circuit in LTspice, a SPICE based circuit simulation tool. The match was satisfactory. The bode plot obtained from the prototype electronic impedance realized is given in Fig. 2.4.

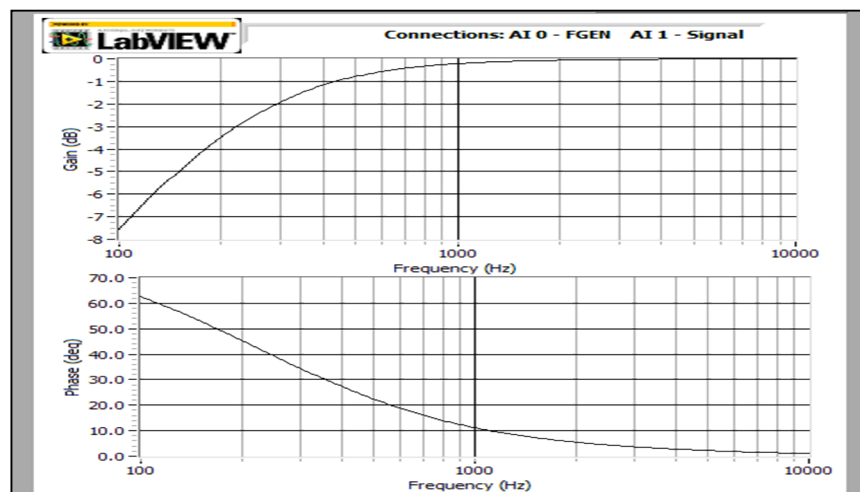


Fig. 2.4. Bode plot obtained for the electronic inductance shown in Fig 2.3.

## 2.4 INSTRUMENTATION AMPLIFIER

Amplification stage of the front-end circuit is implemented with an instrumentation amplifier. Basic schematic of an instrumentation amplifier is shown in Fig. 2.5. It is a type of differential amplifier that has two input buffer amplifiers to eliminate the need for input impedance matching and thus make the amplifier particularly suitable for use in measurement and test equipment. Other important characteristics of the instrumentation amplifier are as follows:-

- Very low DC offset
- Low drift
- Low noise
- Very high open-loop gain
- Very high Common-Mode Rejection Ratio (CMRR)
- Very high input impedances
- Single resistance gain adjustment

Instrumentation amplifiers are used where great accuracy and stability of the circuit are required. It consists of, ideally, infinite impedance buffers, followed by a differential subtractor.

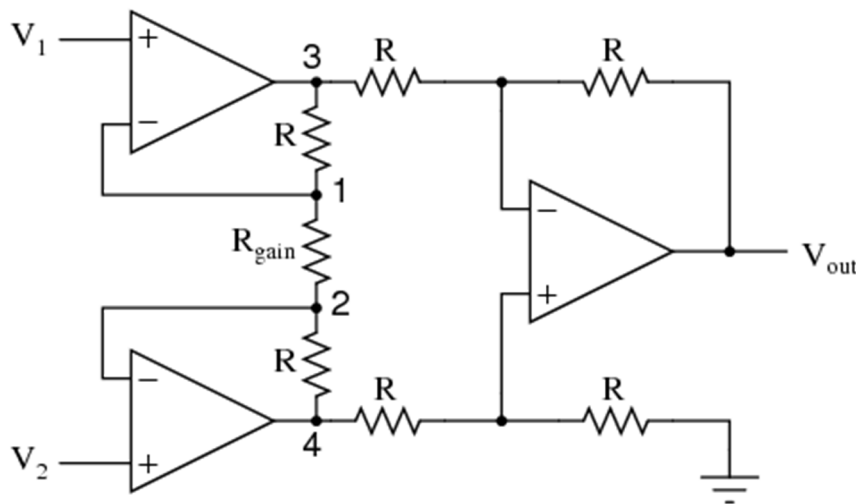


Figure 2.5: Basic schematic of an instrumentation amplifier showing two buffer and one differential amplifier stages.

Gain of the instrumentation amplifier circuit shown in the figure is given by (2.7)

$$V_{out} = \left(1 + \frac{2R_1}{R_{gain}}\right) \frac{R_3}{R_2} (V_2 - V_1) \quad (2.7)$$

## 2.5 PHASE SENSITIVE DETECTOR

Amplifier stage is followed by Phase Sensitive Detector (PSD) to filter out any noise from the output. PSD, also known as lock-in amplifier, comprises of following two stages:-

### 2.5.1 Mixer

### 2.5.2 Low Pass Filter

#### 2.5.1 Mixer

Schematic of the Mixer stage of Phase Sensitive Detector is as shown in Fig. 2.6. This part has been implemented using following components:-

- Op-amp OP07C: used as unity gain inverting amplifier
- LM311 Comparator: for square wave generation to operate the switch at excitation frequency
- MAX4053 switch: this is used to select between in-phase and 180° out-of-phase components of INA output

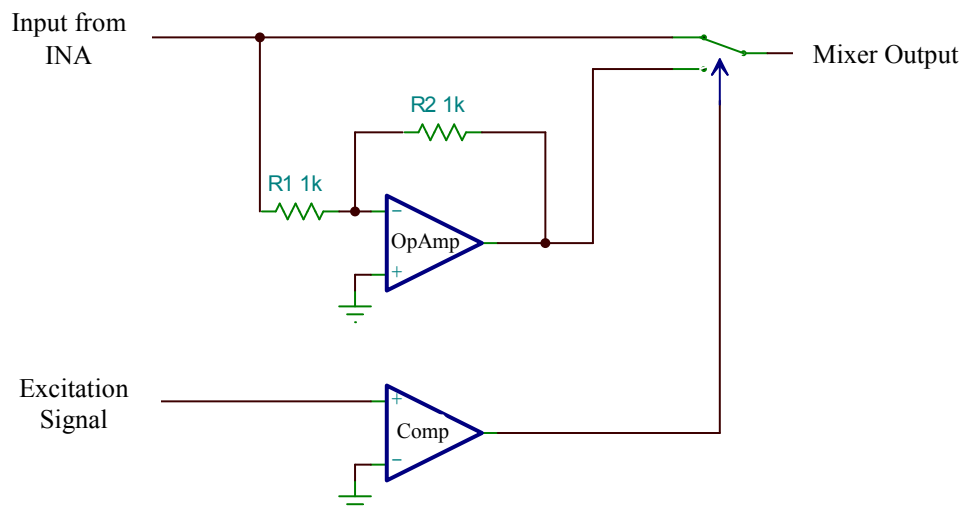


Figure 2.6: Mixer stage of the Phase Sensitive Detector

Excitation signal is fed to the comparator's positive terminal with the negative terminal connected to ground. Comparator output is then used to operate one of the channels of MAX4053 switch IC. This ensures that the switch is operated precisely at the excitation frequency.

INA output is fed to one of the inputs of the switch. Another input of the switch is connected to  $180^\circ$  phase-shifted INA output. Output of the switch is the mixer stage output that comprises of following frequency components:-

$$\text{Mixer Output Frequency} = n \times f_{ref} \pm \Delta f \quad (2.8)$$

As evident from above, the frequency component corresponding to the excitation frequency will be available as zero frequency (i.e. DC) and subsequent harmonics. Fig. 2.7 shows the frequency spectrum of mixer output as observed in LabVIEW environment. To filter out all the undesired frequencies, the mixer output must be passed through a low pass filter.

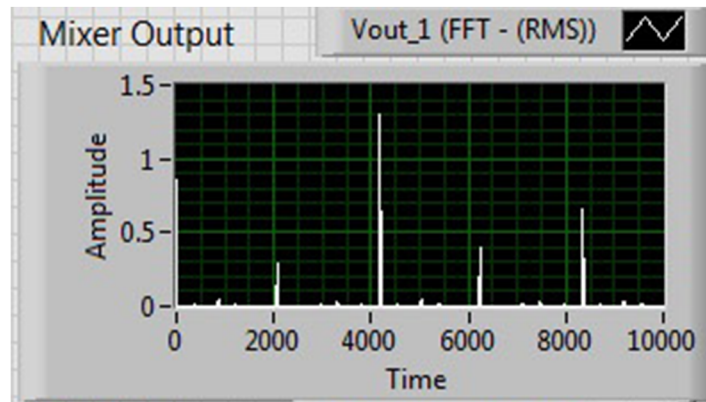


Figure 2.7: Frequency spectrum of Mixer output

### 2.5.2 Low Pass Filter

Fig. 2.8 shows one of the most common implementations of the low-pass filter circuit. This particular configuration is called a Butterworth filter and is characterized by a very flat response in the pass band portion of its response curve. Ideally, a low-pass filter will pass frequencies from DC up through a specified frequency, called the cut-off frequency, with no attenuation or loss. Beyond the cut-off frequency, the filter ideally offers infinite attenuation to the signal.

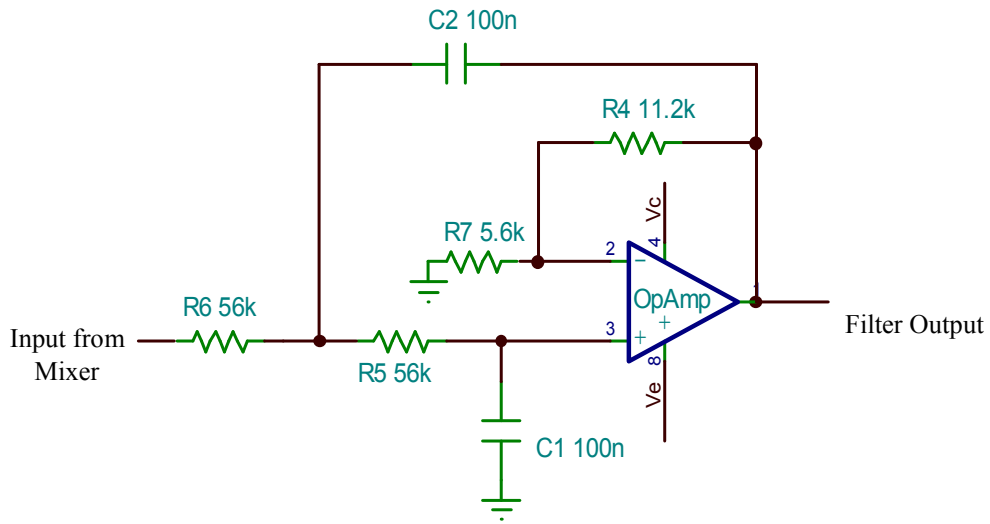


Figure 2.8: Low-pass filter stage of the Phase Sensitive Detector

Standard components including Op-amp IC OP07C have been used to implement a second order Butterworth filter. Cut-off frequency of the filter is given by (2.9)

$$f = \frac{1}{2\pi R \sqrt{C_1 C_2}} \quad (2.9)$$

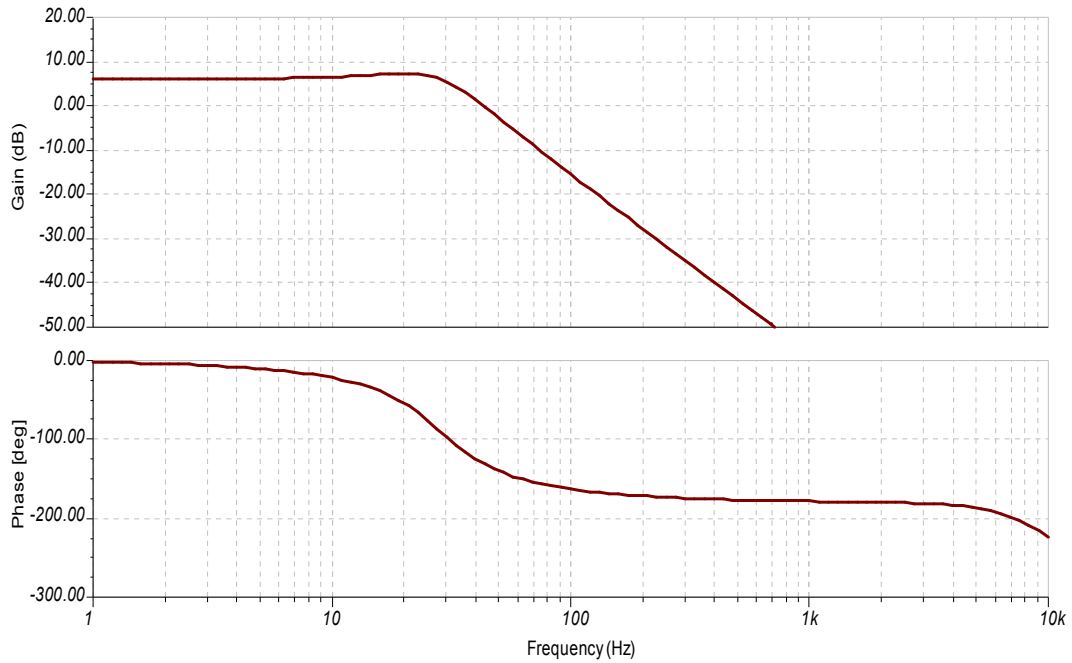


Figure 2.9: Transfer characteristics of Butterworth filter as simulated in TINA-TI

For  $C_1 = C_2 = 100 \text{ nF}$  and  $R_5=R_6=R=56 \text{ k}\Omega$ , cut-off frequency comes out to be 28 Hz. Transfer characteristics of the Butterworth filter were drawn using the simulation software TINA-TI. Transfer characteristics, obtained by simulating the filter are as shown in the Fig. 2.9. Simulation result was compared with the bode plot obtained from the actual implementation of the filter and the match was found to be satisfactory.

## **2.6 COMPLETE CIRCUIT**

The complete circuit after replacement of  $R_r$  and  $L_r$  with GIC and integration of mixer and LPF stages after instrumentation amplification stage is shown in Fig. 2.10. As discussed earlier, final output from the circuit is taken from Low Pass Filter stage. It's a DC signal and is a direct measure of the change in sensor coil inductance.

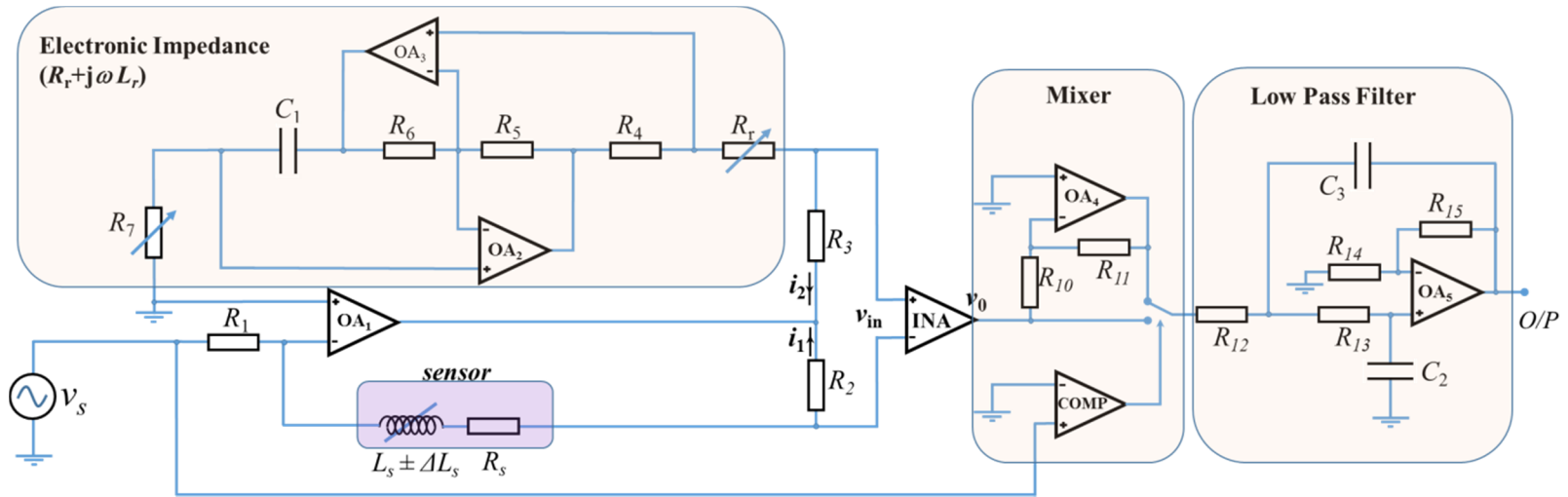


Figure 2.10: Complete circuit after integration of GIC and the Phase Sensitive Detector



### 3.1 DIGITAL POTENTIOMETER IC MCP4261

MCP4261 is an 8-bit dual SPI digital potentiometer IC from Microchip. Salient features of the IC are as follows:-

- 3.1.1 Dual resistor network
- 3.1.2 8-bit resolution (257 steps)
- 3.1.3 Non-volatile Memory with automatic recall of saved wiper setting
- 3.1.4 Communication using Serial Peripheral Interface (SPI)
  - Supports up to 10 MHz clock frequency
  - Supported modes: 0,0 and 1,1
  - High-Speed Read/Writes to wiper registers
  - Read/Write to Data EEPROM registers
- 3.1.5 Wide Operating Voltage:
  - 2.7V to 5.5V - Device Characteristics specified
  - 1.8V to 5.5V – Device operation
- 3.1.6 Extended temperature range (-40°C to +125°C)

MCP4261 has two resistor networks  $R_0$  and  $R_1$  and the same have been used in Rheostat mode.  $R_0$  has been used to replace  $R_r$  in GIC circuit and  $R_1$  has replaced  $R_7$ . As the  $R_0$  value varies from 0 to 5 k $\Omega$  in 257 steps, a fixed resistance of 45  $\Omega$  has been placed in parallel to it to restrict the upper limit and in turn, improve the resolution.

#### 3.1.1 Resistor Network

As mentioned earlier, MCP4261 has two resistor networks of 8-bit resolutions. Each resistor network allows zero scale to full scale connections. Fig. 3.2 shows representation of the resistive network of the device. The Resistor Network is made up of several parts. These include:

- Resistor Ladder
- Wiper
- Terminal Connections

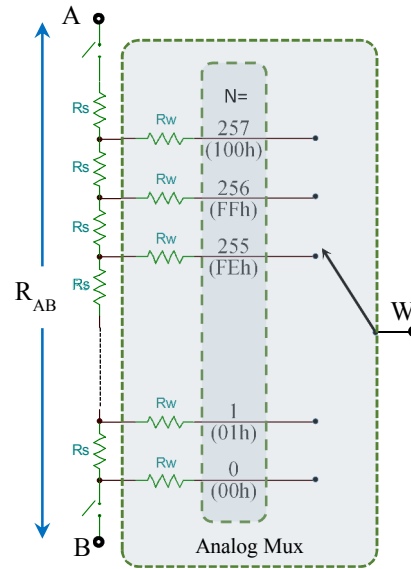


Figure 3.2: Resistor network structure of MCP4261 digital potentiometer IC

The resistor ladder is a series of equal value resistors ( $R_s$ ) with a connection point between the two resistors. The total number of resistors in the ladder determines the  $R_{AB}$  resistance. The end points of the resistor ladder are connected to analog switches which are connected to the device Terminal A and Terminal B pins. There are 256 resistors in a string between terminal A and terminal B. The wiper can be set to tap onto any of these 256 resistors thus providing 257 possible settings (including terminal A and terminal B). For the selected device i.e. MCP4261, total resistance  $R_{AB}$  between terminals A and B is 5 k $\Omega$ .

### 3.1.2 Volatile and Non-Volatile (EEPROM) Memory Registers

The device memory has 16 locations and each is 9-bits wide (16x9 bits). This memory space contains both volatile and non-volatile locations. Non-volatile memory is EEPROM type and is guaranteed to give life of 1,00,000 write cycles. The complete memory map of the device is as shown in the Table 3.1.

Table 3.1: Memory Map of MCP4261 registers

Address	Function	Memory Type
00h	Volatile Wiper 0	RAM
01h	Volatile Wiper 1	RAM
02h	Non-Volatile Wiper 0	EEPROM
03h	Non-Volatile Wiper 1	EEPROM
04h	Volatile TCON Register	RAM
05h	Status Register	RAM
06h	Data EEPROM	EEPROM
07h	Data EEPROM	EEPROM
08h	Data EEPROM	EEPROM
09h	Data EEPROM	EEPROM
0Ah	Data EEPROM	EEPROM
0Bh	Data EEPROM	EEPROM
0Ch	Data EEPROM	EEPROM
0Dh	Data EEPROM	EEPROM
0Eh	Data EEPROM	EEPROM
0Fh	Data EEPROM	EEPROM

Wiper positions of resistor networks  $R_0$  &  $R_1$  correspond to the volatile registers at 00h and 01h memory locations respectively. Memory addresses 02h and 03h are allocated to non-volatile memory registers corresponding to resistor networks  $R_0$  and  $R_1$  wipers. When the device powers up, volatile wiper registers are loaded with values in the corresponding non-volatile wiper registers.

### 3.1.3 Serial Interface (SPI)

The MCP4261 devices support the SPI serial protocol. This SPI operates in the slave mode (does not generate the serial clock). The SPI interface uses up to four pins. These are:

- SDI - Serial Data In
- SDO - Serial Data Out
- SCK - Serial Clock
- CS - Chip Select

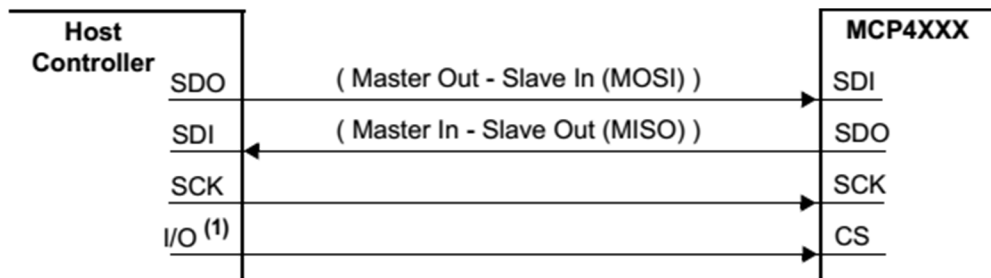


Figure 3.3: Typical SPI interface connections

Typical SPI Interface connections are shown in Figure 3.3. In SPI interface, the Master's output pin is connected to the Slave's input pin and the Master's input pin is connected to the Slave's output pin. The MCP4261's SPI module supports two (of the four) standard SPI modes. These are Mode (0,0) and (1,1).

The SPI mode is determined by the state of the SCK pin (High or Low) when the CS pin transitions from inactive to active.

The operation of the four SPI interface pins, SDI, SDO, SCK, and CS, are discussed in below:

#### **Serial Data In (SDI)**

The Serial Data In (SDI) signal is the data signal into the device. The value on this pin is latched on the rising edge of the SCK signal.

#### **Serial Data Out (SDO)**

The Serial Data Out (SDO) signal is the data signal out of the device. The value on this pin is driven on the falling edge of the SCK signal. Once the CS pin is forced to the active level, the SDO pin will be driven. The state of the SDO pin is determined by the serial bit's position in the command, the command selected, and if there is a command error state (CMDERR).

## **Serial Clock (SCK)**

SPI rate of operation is determined by this Serial Clock generated by the SPI Master. The SPI interface of MCP4261 is specified to operate up to 10 MHz. The actual clock rate depends on the configuration of the system and the serial command used.

## **Chip Select (CS)**

The Chip Select (CS) signal is used to select the device and frame a command sequence. To start a command, or sequence of commands, the CS signal must transition from the inactive state to an active state. After the CS signal has gone active, the SDO pin is driven and the clock bit counter is reset. After the command has ended, the CS signal is again driven to inactive state.

### **3.1.4 SPI Modes**

The SPI module supports two (of the four) standard SPI modes. These are Mode 0,0 and 1,1. The mode is determined by the state of the SDI pin on the rising edge of the 1st clock bit (of the 8-bit byte).

#### **Mode 0,0**

In Mode 0,0: SCK, in idle state, is at low level and data is clocked-in on the SDI pin on the rising edge of SCK and clocked out on the SDO pin on the falling edge of SCK.

#### **Mode 1,1**

In Mode 1,1, SCK, in idle state, is at high level and data is clocked-in on the SDI pin on the rising edge of SCK and clocked out on the SDO pin on the falling edge of SCK.

### **3.1.5 Device Commands**

The MCP4261's SPI command format supports 16 memory address locations and four commands:-

- Increment wiper
- Decrement wiper
- Read data
- Write data

Increment Wiper and Decrement Wiper commands are 8- bit commands and contain a Command Byte. On the other hand, Read Data and Write Data commands are 16-bit commands that contain a Command Byte and a Data Byte. Formats of both, the 8-bit as well as 16-bit commands are as shown in Fig 3.4. The Command Byte also contains two data Bits. Also, Increment/Decrement commands can be performed only on the volatile wiper registers. However, read/write data commands work for both, volatile as well as non-volatile memories.

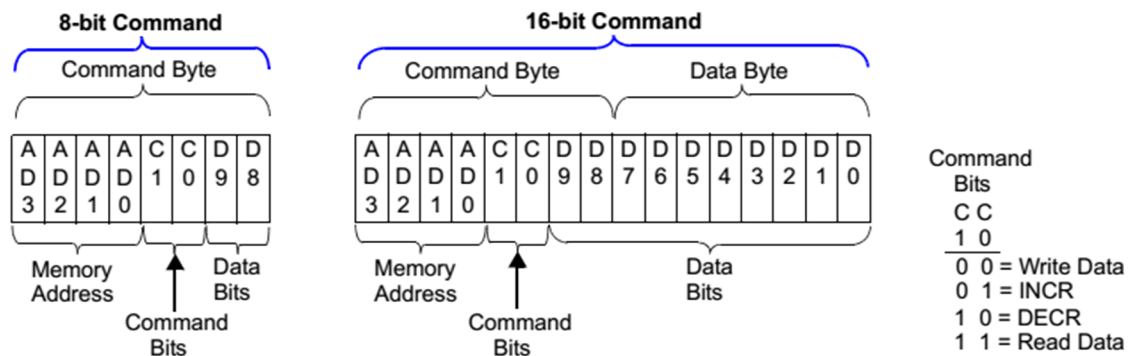


Figure 3.4: Formats of MCP4261 supported SPI commands [16]

### 3.2 AVR SERIES MICROCONTROLLER ATMEGA8

The Atmel<sup>®</sup> AVR<sup>®</sup> ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. Pin configuration of the microcontroller is as shown in Fig. 3.5.

Some of the salient features of this microcontroller are as follows:-

- Operating voltage 4.5V-5.5V
- Supports frequency up to 16 MHz
- 8 kb of in-system programmable flash with Read-While-Write capabilities
- 512 bytes of EEPROM
- 1 kb of SRAM
- 23 general purpose I/O lines

- 32 general purpose working registers
- Two 8-bit timer/counters with separate prescaler, one compare mode
- One 16-bit timer/counter with separate prescaler, compare mode, and capture mode
- Internal Calibrated RC Oscillator
- Internal and external interrupts
- A serial programmable USART
- A byte oriented Two wire Serial Interface
- A 6-channel ADC with 10-bit accuracy
- A programmable Watchdog Timer with Internal Oscillator
- Master/Slave SPI Serial Interface
- On-chip Analog Comparator
- Five software selectable power saving modes

Microcontroller assumes a key role in the calibration process by first acquiring the data from PSD output and then making necessary changes in MCP4261 resistor networks using SPI interface to achieve a perfect balance for offset inductance and coil resistance.

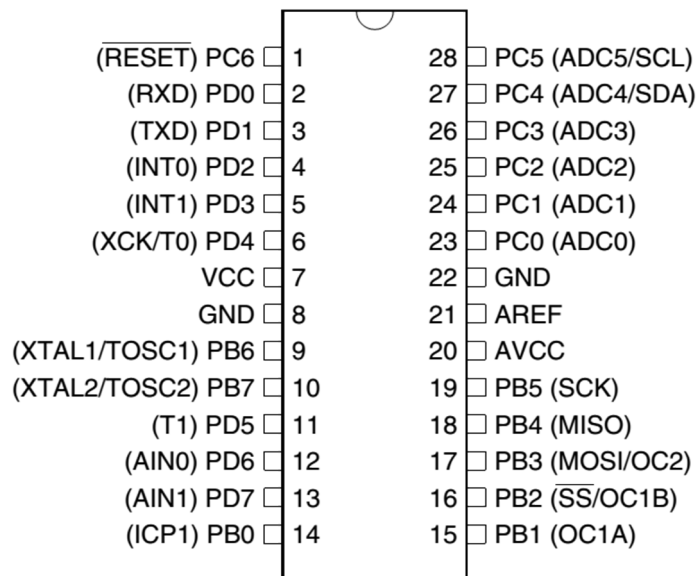


Figure 3.5: ATmega8 pin diagram [17]

The progress of the calibration process is displayed on LCD screen with details like output level, resistor network wiper position and Increment/Decrement cycle under progress. After the calibration process is over, the microcontroller also displays the final output of the circuit on the LCD screen.

### **3.3 CALIBRATION PROCESS**

Calibration process is achieved in three steps:-

3.3.1 Adjustment of  $R_r$  by giving DC excitation

3.3.2 Adjustment of  $R_7$  by giving 2 kHz excitation signal

3.3.3 Storage of Wiper Positions

#### **3.3.1 Adjustment of $R_r$**

This step is performed by giving a dc excitation of 0.2 volt to the circuit. As, the inductance offers zero impedance for dc excitation, this ensures that the output from the circuit corresponds only to the imbalance between  $R_s$  and  $R_r$ . This output is then monitored at ADC0 channel of the microcontroller. The microcontroller then interacts with DigiPot IC MCP4261 using SPI serial interface and resistor network  $R_0$  wiper is adjusted until the output reaches zero.

#### **3.3.2 Adjustment of $R_7$**

Once the coil resistance has been balanced, the coil offset inductance is required to be balanced by adjusting  $R_7$ . This is achieved by giving sinusoidal wave excitation to the circuit. Excitation selection is done by the microcontroller by controlling one of the channels of MAX4053. The circuit output is then monitored by the microcontroller and required controls are sent to the DigiPot to change the resistor network  $R_1$  wiper until zero output conditions are achieved.

#### **3.3.3 Storage of Wiper Positions**

At the end of each step, the corresponding wiper positions are written to the respective non-volatile memory registers by sending SPI commands to the DigiPot. This ensures that the calibration settings are recalled during the next power ON cycle.

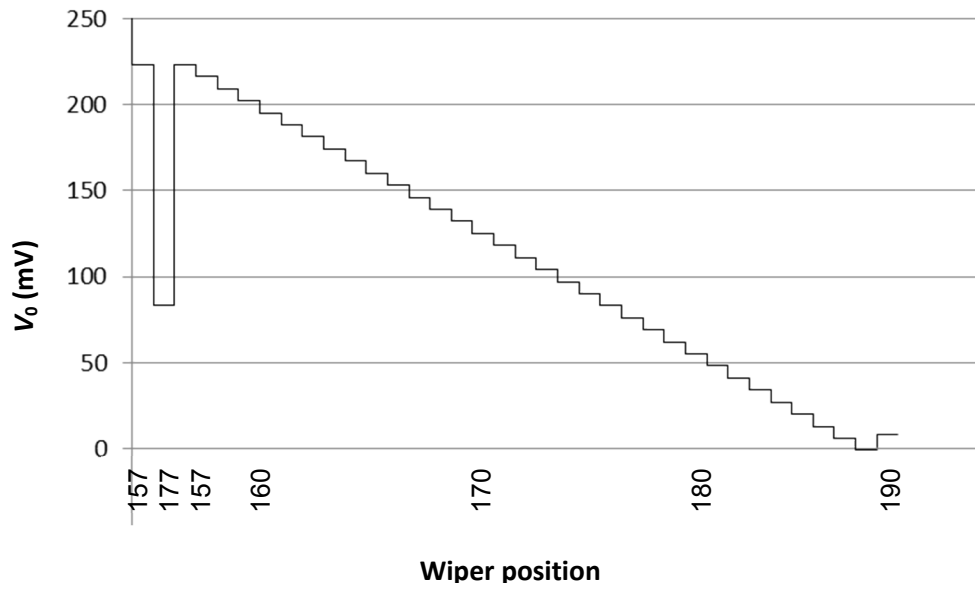


Figure 3.6: Auto-calibration process

Auto calibration, as achieved, by changing the wiper position using the microcontroller is shown in Fig. 3.6. Decision to increment or decrement the wiper is taken by observing the change in output after incrementing the wiper position by 20. Following this, the wiper position is returned back to original and then incremented/decremented in steps of one until the output becomes minimum.

### 3.4 USING THE ADC

10 bit ADC of the microcontroller is used to monitor the circuit output for calibration and output display purpose. The circuit output from PSD can vary from -5V to +5V. However, the input to the ADC port can only be a value from 0 to +5V. To overcome this problem, arrangement as shown in Fig. 3.7 was made to map the -5V to +5V range onto 0 to +5V range. This ensured that the output always remained within the desired limits before it was fed to ADC0 port of the microcontroller.

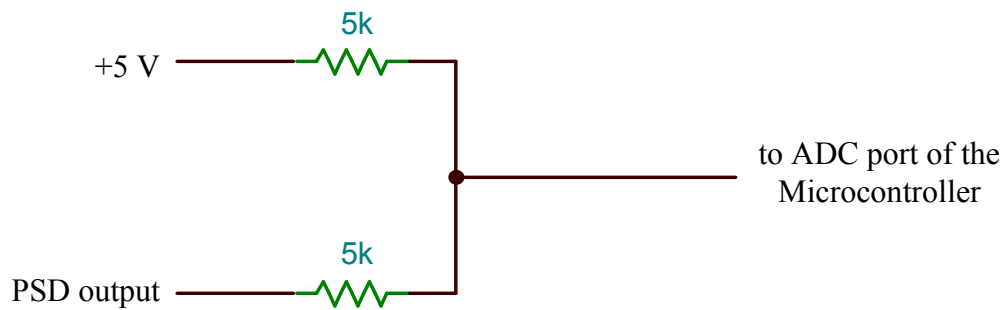


Figure 3.7: Mapping of -5V to +5V range onto 0 to +5V range

The microcontroller program has been written in C language using Atmel Studio 6.1. Flow chart of the same is placed in the Fig. 3.8 and the complete code has been placed as Appendix-A.

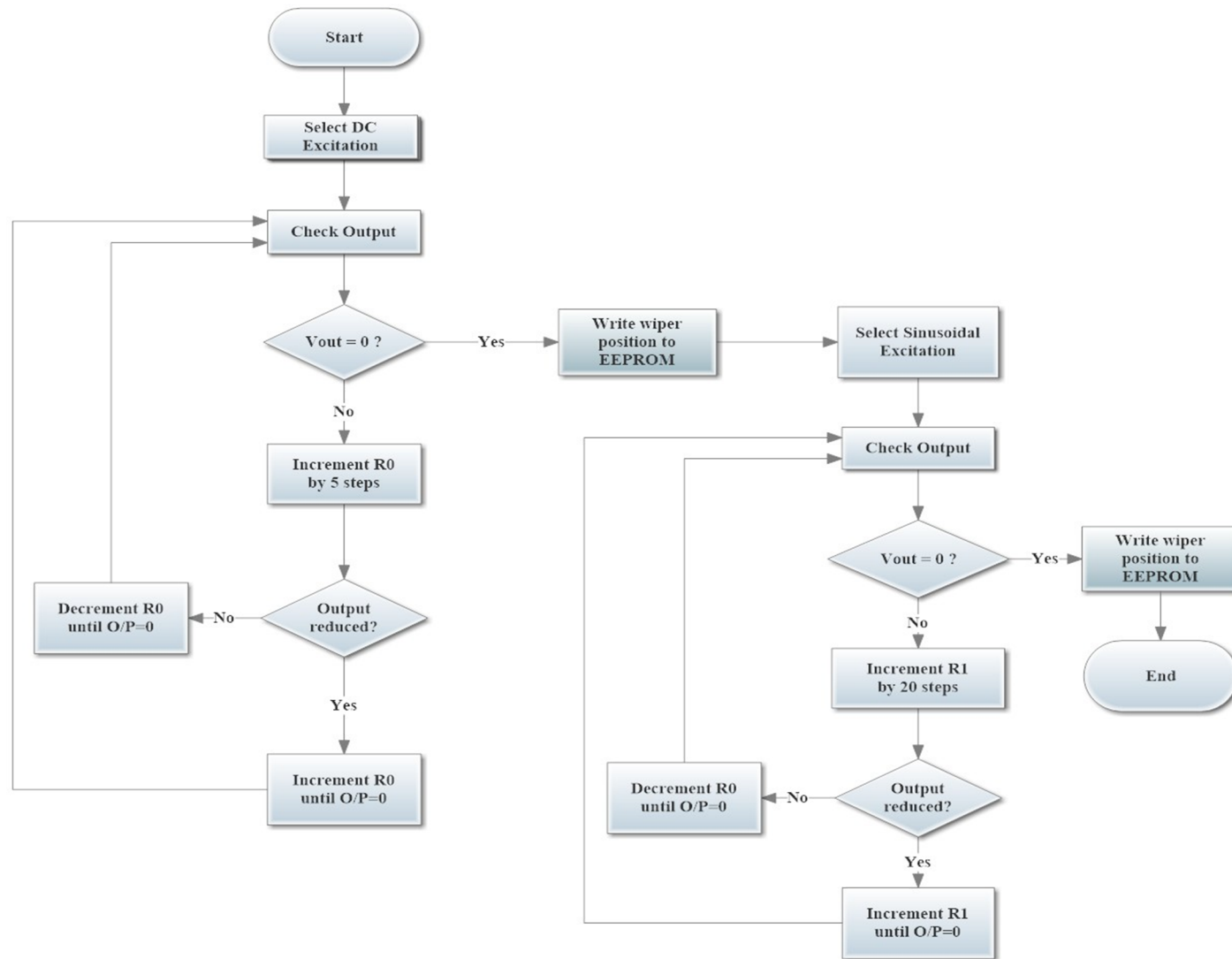


Figure 3.8: Flow chart of Microcontroller program for calibration of the circuit

### 3.5 GENERATION OF SINE WAVE EXCITATION

Sine wave excitation of 2 kHz has been generated using the XR2206 Function Generator IC. It is a monolithic function generator integrated circuit capable of producing high quality sine, square, triangle, ramp, and pulse waveforms of high-stability and accuracy. The output waveforms can also be both amplitude and frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01 Hz to more than 1 MHz. The circuit diagram for generation of the sine wave is as shown in Fig. 3.9. Frequency of the signal generated is given by  $f = \frac{1}{R_1 C}$ . Capacitor C is connected between pin no 5 and 6 of the IC and  $R_1$  between pin no 7 and ground. To obtain a frequency of 2 kHz, C was chosen to be 100 nF and  $R_1$  is a potentiometer that was tuned by monitoring the frequency of the signal on an oscilloscope.

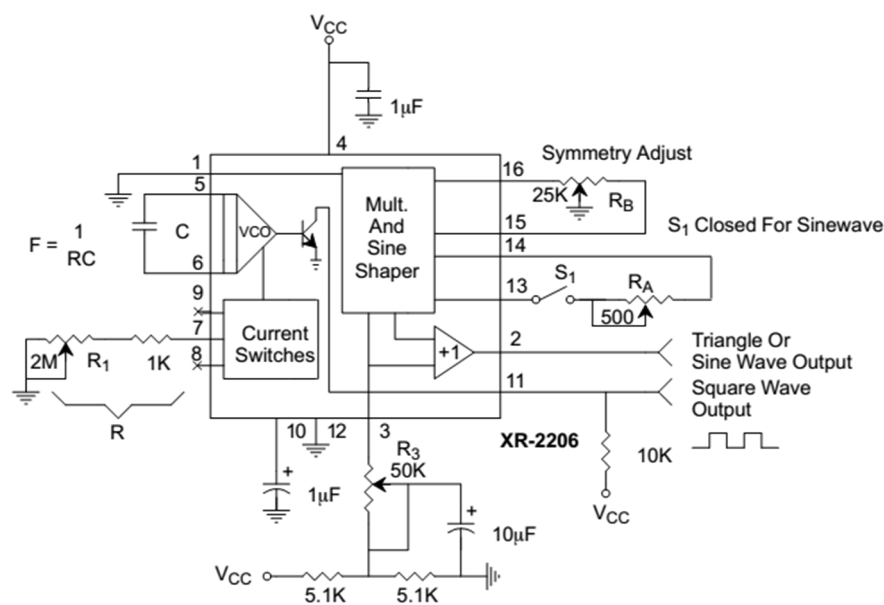


Figure 3.9: Generation of sinusoidal excitation signal using XR2206 [18]

The DC level at the output (Pin 2) is approximately the same as the DC bias at Pin 3. The potentiometer can be adjusted for setting the output dc level to zero. All the ground connections in the above circuit have been connected to -5V. The harmonic content of sinusoidal output can be reduced to -0.5% by adjusting the potentiometers  $R_a$  and  $R_b$ . The potentiometer,  $R_A$ , adjusts the sine-shaping resistor, and  $R_B$  provides the fine adjustment for the waveform symmetry.

The adjustment procedure is as follows:

- Set  $R_B$  at midpoint and adjust  $R_A$  for minimum distortion.
- With  $R_A$  set as above, adjust  $R_B$  to further reduce distortion.

### 3.6 16x2 LCD

A 16x2 LCD (Liquid Crystal Display) screen has been used to display the progress of the calibration process at start-up. After calibration process is completed, the final output of the circuit along with the wiper positions corresponding to the DigiPot resistor networks  $R_0$  and  $R_1$  is also displayed on the LCD.

LCD screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segment and other multi segment LEDs as they are economical, easily programmable, have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Pin diagram of the LCD module is as shown in Fig. 3.9 and the description of each pin has been given in Table 3.2.

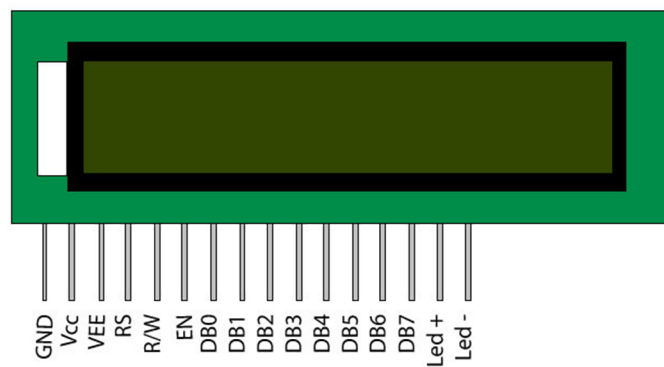


Figure 3.10: Pin diagram of a 16x2 LCD module [19]

Table 3.2: Description of the pins of 16x2 LCD module

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	V <sub>cc</sub>
3	Contrast adjustment; through a variable resistor	V <sub>ee</sub>
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V <sub>CC</sub> (5V)	Led+
16	Backlight Ground (0V)	Led-

The display has been used in advanced 4-bit mode so that only four data pins DB4 to DB7 of LCD are used for data transfer. Thus a total of seven GPIO (General Purpose Input/Output) pins of microcontroller are utilized for display purpose.

## CHAPTER 4

### EXPERIMENTAL SETUP AND RESULTS

#### 4.1 VIRTUAL INSTRUMENT

The software package used to build the virtual instrument for the project is LabVIEW-2011. LabVIEW stands for Laboratory Virtual Instrumentation Engineering Workbench. It is a system designing platform and development environment for a visual programming language from National Instruments. Virtual instrumentation is defined as the combination of measurement and control hardware and application software with industry standard technology to create user defined instrumentation system. LabVIEW programs are called Virtual Instrumentation (VI) where the operation resembles physical instruments and their functions. VI has the following two components:-

- Front panel - serves as user interface and control
- Block diagram - contains the graphical blocks and icons to define the functionality of the VI

LabVIEW has a set of blocks to perform function of an oscilloscope, function generator, bode analyser, impedance analyser, variable power supply etc., inbuilt in it for simulating the input signals and observe the output responses.

Front panel of the VI developed for the circuit is shown in Fig. 4.1. Signal waveforms at different stages are displayed. 1<sup>st</sup> graph displays the excitation signal waveform. INA output has been shown in 2<sup>nd</sup> graph in both, the time domain as well as frequency domain. Output from the mixer stage of PSD is displayed in 3<sup>rd</sup> graph. Harmonic components in accordance with (2.8) can be seen in the corresponding frequency domain representation of the mixer output. Final output as obtained from the Low Pass Filter stage is displayed in 4<sup>th</sup> graph. As seen in the frequency domain, all the harmonic components are filtered out at this stage and amplitude of the dc component is displayed by a numeric indicator on the panel.

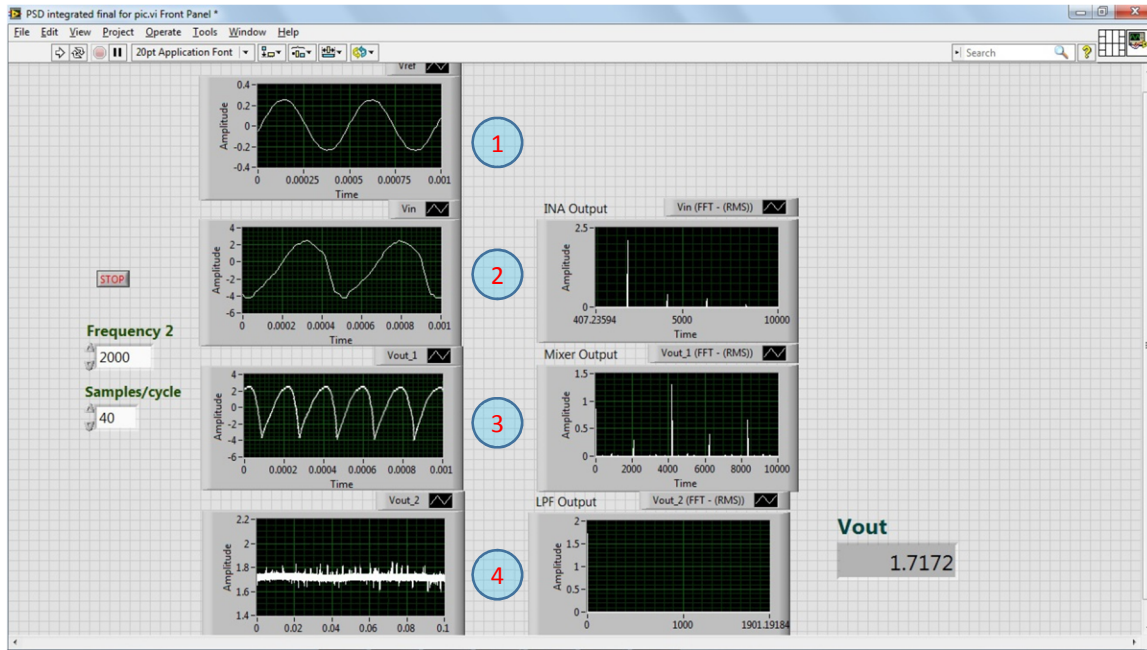


Figure 4.1: Front panel of the VI developed in LabVIEW

## 4.2 PROTOTYPE OF THE CIRCUIT-STAGE I

A prototype of the developed circuit has been made using the best and easily available components and ICs in the market. Op-amps  $OA_1$ ,  $OA_2$  and  $OA_3$  have been realized using the low offset IC OP07C. INA has been realized using INA129P. Excitation signal was generated using National Instruments ELVIS-II platform that was a sinusoidal wave with 2 kHz frequency. INA output was acquired using NI ELVIS-II and then was processed by Phase Sensitive Detector (PSD) implemented in a Virtual Instrument (VI) developed in a LabVIEW environment using a personal computer. Offset inductance and coil resistance balancing was achieved by manually adjusting the GIC potentiometers  $R_r$  and  $R_7$ .

## 4.3 PROTOTYPE OF THE CIRCUIT-STAGE II

The second stage of prototype development aimed at doing away with the requirement of NI ELVIS board and personal computer. To achieve this PSD implementation was moved from LabVIEW environment to the hardware based as discussed in 2.5. Excitation signal was also generated by configuring XR2206 function generator IC to give a 2 kHz sinusoidal waveform as discussed in 3.5.

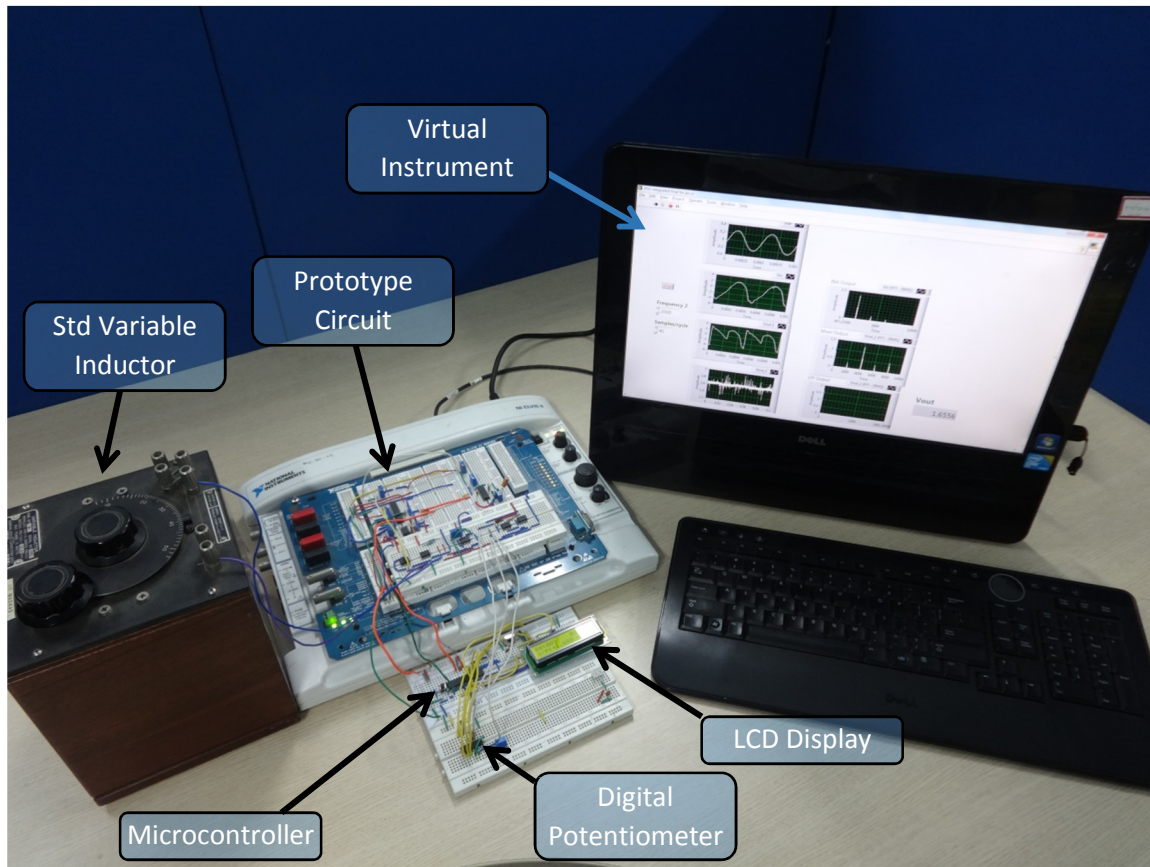


Figure 4.2: Experimental setup

Balancing of offset inductance and coil resistance was done by replacing the GIC resistors  $R_7$  and  $R_7$  with two resistor networks  $R_0$  and  $R_1$  of DigiPot IC MCP4261. AVR series ATmega8 microcontroller was used to control the auto calibration process by monitoring the output and displaying the same on 16x2 LCD module. The complete experimental setup is shown in Fig. 4.2.

#### 4.4 PCB DESIGN AND FABRICATION

A dual layer PCB has been designed and components mounted on that as shown in Fig 4.3. PCB layouts, both component side and the solder side, are attached as Appendix-B.

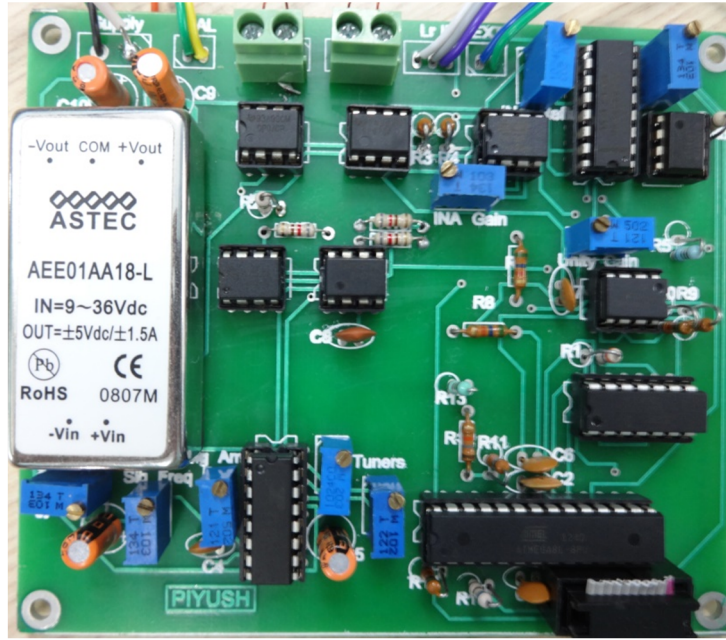


Figure 4.3: Components mounted on double layer PCB

Apart from the connector for connecting sensor coil, provision for connecting external reference coil, as discussed in 4.4 is also provided. An SPDT (Single Pole Dual Throw) switch is provided at the front panel to select the Internal/External reference coil.

#### 4.5 ANNEALED/NON-ANNEALED METAL DETECTION

Two identical planar coils as shown in Fig. 4.4 have been designed and fabricated on a single layer PCB. These coils are connected to the sensor coil and reference coil terminals. At this stage, the circuit is in balanced condition and hence produces zero output. Now the annealed metal piece is placed permanently over the reference planar coil. This would result in an increase in the reference coil inductance and hence a corresponding output from the circuit is generated. If another similar metal piece is placed over sensor coil then the circuit produces zero output if the piece under test is also annealed. If not, then the circuit gives the output corresponding to the difference between the two inductances. Experimental results may be used to set a threshold and final decision showing Annealed/Non-annealed can be displayed on LCD with the help of microcontroller.

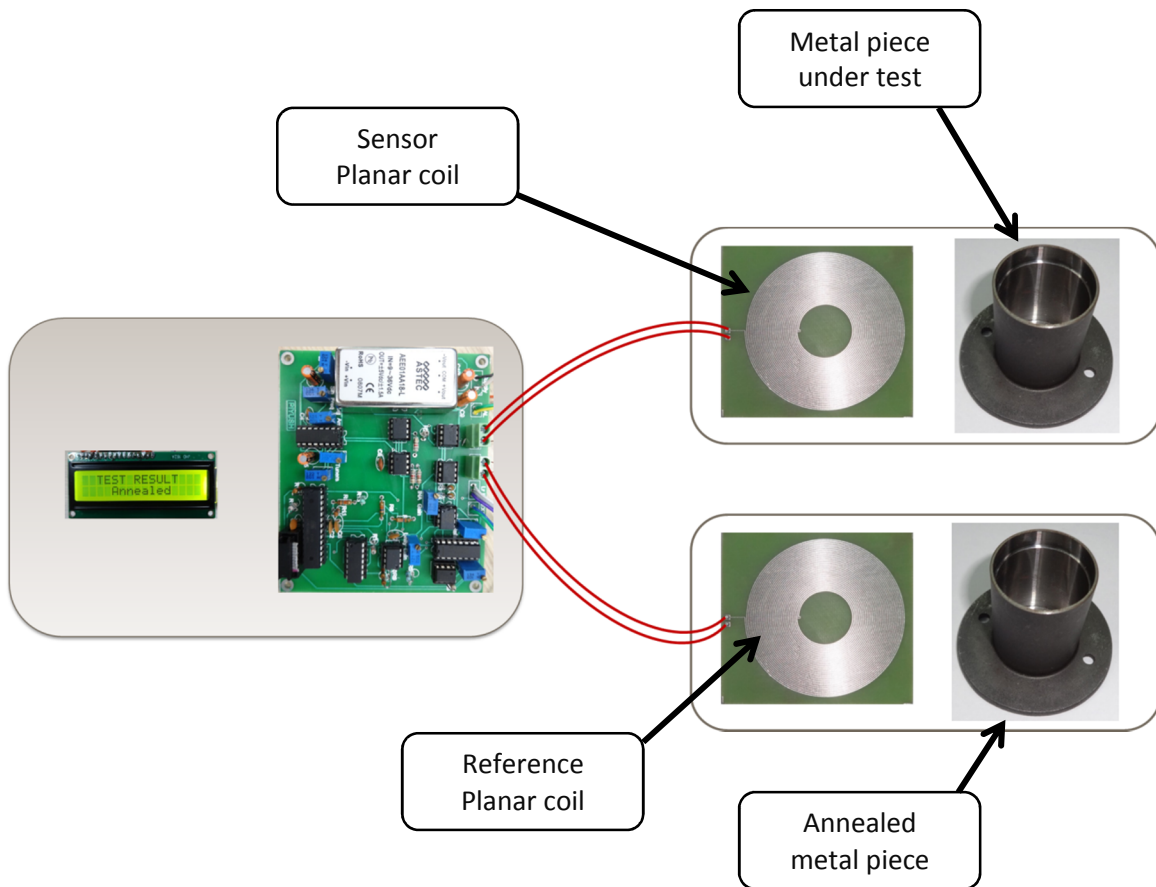


Figure 4.4: Annealed/unannealed metal detection

#### 4.6 PROXIMITY SENSING

The circuit has also been successfully tested for proximity detection applications. To achieve this, the SPDT switch was used to select GIC as reference coil and a sensor coil as shown in Fig. 4.5 was connected to the sensor coil terminal of the PCB.

After the power to the circuit is turned ON, microcontroller first calibrates the circuit by balancing the offset inductance and coil resistance as explained in 3.3. At this stage the circuit output is zero. Whenever any metal object is brought in proximity to the sensor coil, output is generated corresponding to the change in coil inductance. This is detected by the microcontroller and the result is displayed on 16x2 LCD module.

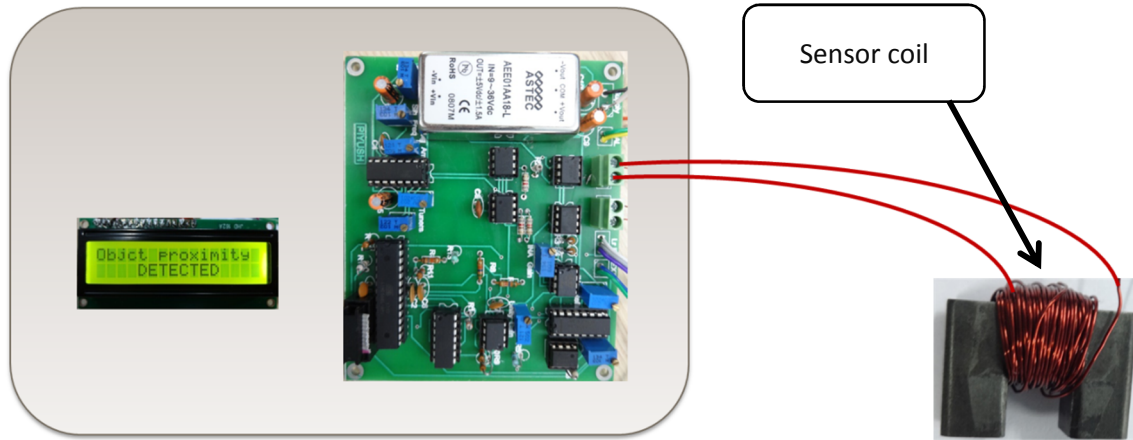


Figure 4.5: Proximity sensing

#### 4.7 ACCURACY AND LINEARITY

The circuit has been tested for its accuracy and linearity by using a standard variable inductor (make- General Radio Company, USA) capable of providing an inductance value from 8 to 52 mH. In this test,  $L_s$  was set at 25 mH. The coil resistance  $R_s$  was measured to be  $36.09 \Omega$ .  $R_1$  was selected as  $160 \Omega$  and  $R_2$  &  $R_3$  as  $300 \Omega$ . GIC was realized using  $C_1 = 10 \text{ nF}$  and  $R_4 = R_5 = R_6 = 820 \Omega$ . The value of  $R_r$  was set to  $36 \Omega$ . Before doing the test, calibration of circuit is done by the microcontroller. First DC excitation is given to the circuit and  $R_r$  is adjusted to obtain minimum output value. After  $R_r$  adjustment is completed, sinusoidal excitation is given to the circuit and  $R_7$  is adjusted till the condition  $L_s = R_4 R_6 R_7 C_1 / R_5 = L_r$  is achieved. In this condition, the final output was minimum (less than a few mV).

The standard variable inductor was then, varied from 23 mH to 40 mH in 1 mH steps and corresponding output readings were noted. As expected, the final output changed linearly with respect to  $\Delta L_s$  with good accuracy. The worst-case error noted was less than  $\pm 0.6 \%$ . A graph was plotted as shown in the Fig. 4.6 showing the output voltage  $V_0$  and percentage error.

The value of offset coil inductance  $L_s$  of typical sensors available, vary from as low as few tens of  $\mu\text{H}$  [8] to few tens of mH. The proposed circuit has been tested against such conditions and found to be satisfactory.

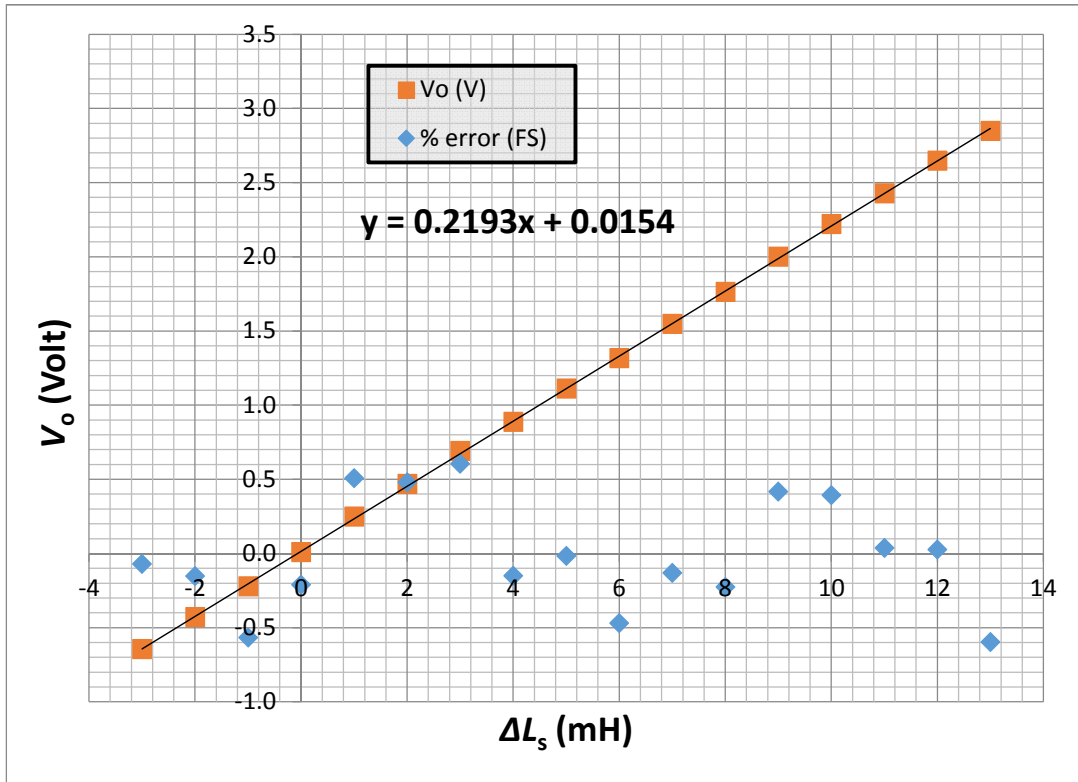


Figure 4.6: Output and error characteristics (% deviation from best linear fit) obtained from the prototype circuit by varying  $L_s \pm \Delta L_s$  from 23 mH to 40 mH (i. e.,  $L_s = 25$  mH) in steps of 1 mH.

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 SUMMARY OF THE WORK**

A simple analog front-end circuit suitable for inductive sensors has been developed. Existing measurement circuits for inductance or inductive sensors have large fixed output due to offset inductance and winding resistance. Thus, change in the final output due to measurand is relatively small. This limits the dynamic range of the sensor system. The developed circuit is capable of nullifying the effect of offset inductance and winding resistance of sensor coil, and providing a final output, which is due to the change in inductance, owing to the change in measurand alone. The output obtained from the circuit is directly proportional to the change in inductance, unlike the unbalanced output from an inductance bridge, which does not vary linearly with inductance. A prototype of the proposed circuit has been developed and tested in the laboratory.

The work has also been published in “Seventh International Conference on Sensing Technology” held at Wellington, New Zealand from Dec. 3 to Dec. 5, 2013.

#### **5.2 FUTURE SCOPE**

The circuit in its present form has been tested with the help of a standard variable inductor as replacement for the sensor coil. Circuit output has shown highly linear and accurate characteristics with respect to the inductance changed manually for the experiment purpose.

The circuit has also been tested for two applications i.e. annealed/unannealed metal detection and proximity sensing. The circuit will be very useful for inductive sensors (even with large offset inductance and coil/winding resistance) whose inductance vary linearly with the measurand. It can also be interfaced with existing inductive proximity sensors for better performance as the output does not contain any contribution from offset inductance and winding resistance.

Due to highly linear characteristics of the circuit, it is especially suited for linear inductive displacement sensors. The same may be tested by integrating it with any of the existing Linear Inductive Displacement sensors or by developing the same for use with the circuit.

## REFERENCES

- [1] E. O. Doebelin, "Measurement systems- application & design," 5th ed. New York: McGraw-Hill, 2004.
- [2] M. Kutila, Viitanen Jouko, and A. Vattulainen, "Scrap metal sorting with color vision and inductive sesor array," proc. of IEEE CIMCA-IAWTIC'05, vol. 2, pp.725,729, Nov. 2005.
- [3] M. Jagiella, K. Barth, and A. Topkaya, "Measuring device and method for its operational setting", US patent 5,218,311, 1991.
- [4] G. Brasseur, "Design rules for robust capacitive sensors," *IEEE Trans. Instr. Meas.*, vol. 52, no. 4, pp. 1261-1265, 2003.
- [5] <http://www.dpnccanada.com/Factory-Automation/News/Long-range-inductive-sensor-for-difficult-applications.html>
- [6] <http://www.sensorsincorporated.com>
- [7] M.R. Mousavi, R.P. Zangabad, S. Chamanian, and M.Bahrami, "Simulation of novel linear inductive displacement sensor," proc. of 21<sup>st</sup> International Conference on Systems Engineering (ICSEng), 2011, 16-18 Aug. 2011, pp. 383 - 385.
- [8] L. Callegaro, "On strategies for automatic bridge balancing," *IEEE Trans. Instr. Meas.*, vol. 54, no. 2, pp. 529-532, April 2005.
- [9] M. Jagiella, S. Fericean, and A. Dorneich, "Progress and recent realizations of miniaturized inductive proximity sensors for automation," *Sensors Journal, IEEE* , vol.6, no.6, pp.1734-1741, Dec. 2006.
- [10] S. Fericean, and R. Droxler, "New noncontacting inductive analog proximity and inductive linear displacement sensors for industrial automation," *Sensors Journal, IEEE* , vol.7, no. 11, pp. 1538-1545, Nov. 2007.
- [11] S. Fericean, M. Friedrich, and E. Gass, "Inductive sensor responsiveto the distance to a conductive or magnetizable object," US Patent 5,504,425, 1996.
- [12] M.A. Atmanand, V. J. Kumar, and V.G. K.Murti, "A novel method of measurement of L and C," *IEEE Trans. Instr. Meas.* vol. 44,no. 4, pp. 898-903, Aug.1995.
- [13] K. Cyril Baby, and B. George, "A simple analog front-end circuit for grounded capacitive sensors with offset capacitance," proc. of (*I2MTC*), 2013 *IEEE International*, pp. 1372-1375, 6-9 May 2013.
- [14] R. Pallas-Areny, and J. G. Webster, "Sensors and signal conditioning," J. Wiley, 2nd Edition, 2001.
- [15] Website: [http://link.springer.com/chapter/10.1007%2F978-3-642-36329-0\\_10](http://link.springer.com/chapter/10.1007%2F978-3-642-36329-0_10)
- [16] Datasheet MCP4261, Microchip Technology [Online]. <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en531252>
- [17] Datasheet ATmega8 microcontroller, ATMEL Corporation [Online]. [http://www.atmel.in/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8\\_L\\_summary.pdf](http://www.atmel.in/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_summary.pdf)

- [18] Datasheet XR2206, EXAR Corporation [Online]. [www.alldatasheet.com/datasheet-pdf/pdf/80496/EXAR/XR2206.html](http://www.alldatasheet.com/datasheet-pdf/pdf/80496/EXAR/XR2206.html)
- [19] <http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>

## **APPENDIX-A**

### **C-CODE FOR PROGRAMMING ATMEGA8** **MICROCONTROLLER**

The program is written for calibrating the circuit in two steps as discussed in 3.3. Pin PD0 is made high to select DC excitation for adjusting the  $R_r$ . Similarly Pin PD0 is kept at low to select sinusoidal excitation for  $R_7$  adjustment. On completion of the calibration process, output is continuously monitored and displayed on LCD module.

```
/*
 * ADCdigiPot.c
 *
 * Created: 06-03-2014 23:11:56
 * Author: Piyush
 */

#include <avr/io.h>
#include <math.h>
#include "MyLibs/ADC/ADC.h"
#include "MyLibs/LCD/lcd.h"
#include "MyLibs/Misc.h"
#include "MyLibs/SPI/spi.h"
#include "MyLibs/MCP4XX1/MCP4xx1.h"

#define READINGS 20
#define POT_STEPS 1
#define ZERO_REF 509
#define TOLERANCE 1
#define ADC_CH 0
#define R0_INIT_VAL 1
#define R0_INIT_TEST_STEPS 4
#define R1_INIT_VAL 150
#define R1_INIT_TEST_STEPS 20
#define EXCITATION_SELECT_DC() PORTD|=(1<<PD0)
#define EXCITATION_SELECT_SINE() PORTD&=~(1<<PD0))

float v_out1, v_out2;

float read_avg_voltage(uint8_t readings, uint8_t ch)
{
    float voltage=0;
    for (int i=0;i<readings;i++)
    {
        voltage+=ReadVoltage(ch);
        _delay_ms(30);
    }
    return(voltage/readings);
}

void minimise_by_decrement(uint8_t PotNo)
{
    uint8_t dec_count=0;
    LCDWriteStringXY(0,0,"DEC");
}
```

```

v_out1=read_avg_voltage(READINGS,ADC_CH);

for (uint8_t count=0; count<3;)
{
    for (int i=0;i<POT_STEPS;i++)
    {
        decrement(PotNo);
    }
    dec_count++;

    LCDWriteIntXY(13,0,read_wiper(PotNo),3);

    v_out2=read_avg_voltage(READINGS,ADC_CH);
    display_output(v_out2);
    LCDWriteIntXY(13,1,dec_count,3);

    if (fabs(v_out2-ZERO_REF)>fabs(v_out1-ZERO_REF))
    {
        count++;
    }
    else count=0;
    v_out1=v_out2;
    if (read_wiper(PotNo)==0)
    {
        return;
    }
}

write_wiper(PotNo, (read_wiper(PotNo)+3*POT_STEPS));
LCDWriteStringXY(0,0,"EXT");
LCDWriteIntXY(13,0,read_wiper(PotNo),3);

v_out2=read_avg_voltage(READINGS,ADC_CH);
display_output(v_out2);
_delay_ms(1000);
}

void minimise_by_increment(uint8_t PotNo)
{
    uint8_t dec_count=0;
    LCDWriteStringXY(0,0,"INC");

    v_out1=read_avg_voltage(READINGS,ADC_CH);

    for (uint8_t count=0; count<3;)
    {
        for (int i=0;i<POT_STEPS;i++)
        {
            increment(PotNo);
        }
        dec_count++;

        LCDWriteIntXY(13,0,read_wiper(PotNo),3);

        v_out2=read_avg_voltage(READINGS,ADC_CH);
        display_output(v_out2);
        LCDWriteIntXY(13,1,dec_count,3);

        if (fabs(v_out2-ZERO_REF)>fabs(v_out1-ZERO_REF))

```

```

        {
            count++;
        }
        else count=0;
        v_out1=v_out2;
        if (read_wiper(PotNo)==511)
        {
            return;
        }
    }

    write_wiper(PotNo, (read_wiper(PotNo)-3*POT_STEPS));
    LCDWriteStringXY(0,0,"EXT");
    LCDWriteIntXY(13,0,read_wiper(PotNo),3);

    v_out2=read_avg_voltage(READINGS,ADC_CH);
    display_output(v_out2);
    _delay_ms(1000);
}

void display_output(float v_out)
{
    if(v_out1>ZERO_REF)
    {
        LCDWriteStringXY(4,1,"+");
    }
    else LCDWriteStringXY(4,1,"-");

    LCDWriteFloatXY(5,1,((fabs(v_out1-ZERO_REF)*10.000)/1023),1,3);
}

int main()
{
    INIT_CHIP();
    DE_SELECT_CHIP();

    Init_SPI(0,1,0,0,0);
    Init_ADC();
    Init_LCD(LS_BLINK|LS_ULINE);

    DDRD |= (1<<PD0); //pin out for excitation selection

    //write_wiper(R1_VOLATILE,read_wiper(R1_N_VOLATILE));
    //happens automatically
    //write_wiper(R0_VOLATILE,read_wiper(R0_N_VOLATILE));
    //happens automatically

    /*****
        Balance Rs first
    *****/

    LCDClear();
    LCDWriteString("Rr auto balance");
    LCDWriteStringXY(0,1,"Loading.....");
    EXCITATION_SELECT_DC();
    _delay_ms(3000);

    v_out1=read_avg_voltage(50,ADC_CH); //read the Vo level 1st
    LCDClear();
    LCDWriteStringXY(0,1,"O/P:");

```

```

display_output(v_out1);
LCDWriteIntXY(13,0,read_wiper(R0_VOLATILE),3);

while ((v_out1<(ZERO_REF-3*TOLERANCE))||((v_out1>(ZERO_REF+3*TOLERANCE)))
{
    write_wiper(R0_VOLATILE,R0_INIT_VAL);
    v_out1=read_avg_voltage(50,ADC_CH);

    //display on LCD
    display_output(v_out1);
    LCDWriteIntXY(13,0,read_wiper(R0_VOLATILE),3);
    _delay_ms(250);

    //increase R0 1st and compare the results
    write_wiper(R0_VOLATILE,(R0_INIT_VAL+R0_INIT_TEST_STEPS));
    v_out2=read_avg_voltage(50,ADC_CH);

    display_output(v_out2);
    LCDWriteIntXY(13,0,read_wiper(R0_VOLATILE),3);
    _delay_ms(250);

    write_wiper(R0_VOLATILE,R0_INIT_VAL);

    if (v_out1>(ZERO_REF+TOLERANCE))
    {
        if (v_out2>v_out1)
        {
            minimise_by_decrement(R0_VOLATILE);
        }
        else
        {
            minimise_by_increment(R0_VOLATILE);
        }
    }
    else
    {
        if (v_out2>v_out1)
        {
            minimise_by_increment(R0_VOLATILE);
        }
        else
        {
            minimise_by_decrement(R0_VOLATILE);
        }
    }
    v_out1=v_out2;
    if (read_wiper(R0_VOLATILE)==0)
    {
        break;
    }
}

LCDWriteStringXY(0,0,"R0 adjust done..");
LCDWriteStringXY(0,1," Writing EEPROM ");
_delay_ms(100);

if (read_wiper(R0_N_VOLATILE)!=read_wiper(R0_VOLATILE))
{
    write_wiper(R0_N_VOLATILE,read_wiper(R0_VOLATILE));
}

```

```

LCDWriteStringXY(0,1," SUCCESSFUL.. ");
_delay_ms(100);

/*****
Balance Ls Now
*****/

LCDClear();
LCDWriteString("Lr auto balance");
LCDWriteStringXY(0,1,"Loading.....");
EXCITATION_SELECT_SINE();
_delay_ms(3000);

v_out1=read_avg_voltage(50,ADC_CH); //read the Vo level 1st
LCDClear();
LCDWriteStringXY(0,1,"O/P:");
display_output(v_out1);
LCDWriteIntXY(13,0,read_wiper(R1_VOLATILE),3);

while ((v_out1<(ZERO_REF-TOLERANCE))||(v_out1>(ZERO_REF+TOLERANCE)))
{
    write_wiper(R1_VOLATILE,R1_INIT_VAL);
    v_out1=read_avg_voltage(50,ADC_CH);

    //display on LCD
    display_output(v_out1);
    LCDWriteIntXY(13,0,read_wiper(R1_VOLATILE),3);
    _delay_ms(250);

    //increase R0 1st and compare the results
    write_wiper(R1_VOLATILE,(R1_INIT_VAL+R1_INIT_TEST_STEPS));
    v_out2=read_avg_voltage(50,ADC_CH);

    display_output(v_out2);
    LCDWriteIntXY(13,0,read_wiper(R1_VOLATILE),3);
    _delay_ms(250);

    write_wiper(R1_VOLATILE,R1_INIT_VAL);

    if (v_out1>(ZERO_REF+TOLERANCE))
    {
        if (v_out2>v_out1)
        {
            minimise_by_decrement(R1_VOLATILE);
        }
        else
        {
            minimise_by_increment(R1_VOLATILE);
        }
    }
    else
    {
        if (v_out2>v_out1)
        {
            minimise_by_increment(R1_VOLATILE);
        }
        else
        {

```

```

        minimise_by_decrement(R1_VOLATILE);
    }
}
v_out1=v_out2;

}

LCDWriteStringXY(0,0,"R1 adjust done..");
LCDWriteStringXY(0,1," Writing EEPROM ");
_delay_ms(100);

if (read_wiper(R1_N_VOLATILE)!=read_wiper(R1_VOLATILE))
{
    write_wiper(R1_N_VOLATILE,read_wiper(R1_VOLATILE));
}

LCDWriteStringXY(0,1," SUCCESSFUL.. ");
_delay_ms(500);

LCDClear();

LCDWriteString("R0:");
LCDWriteIntXY(3,0,read_wiper(R0_VOLATILE),3);
LCDWriteStringXY(10,0,"R1:");
LCDWriteIntXY(13,0,read_wiper(R1_VOLATILE),3);
LCDWriteStringXY(0,1,"O/P:    volt")

while (1)
{
    v_out1=read_avg_voltage(READINGS,ADC_CH);
    display_output(v_out1);
    LCDGotoXY(17,1);
}
}

*****

/*
 * MCP4xx1.h
 *
 * Created: 10-03-2014 23:05:40
 * Author: Piyush
 */

#ifndef _MCP4xx1_H_
#define _MCP4xx1_H_

#define POT_ADDR 4
#define SUCCESS 1
#define ERROR 0
#define R0_VOLATILE 0
#define R1_VOLATILE 1
#define R0_N_VOLATILE 2
#define R1_N_VOLATILE 3
#define VOL_TCON 4
#define STATUS_REG 5

#define INIT_CHIP() DDRB|=(1<<PB2)
#define SELECT_CHIP() PORTB&=~(1<<PB2))
#define DE_SELECT_CHIP() PORTB|=(1<<PB2)

```

```

/*
*****
FUNCTIONS
*****
*/
void Init_chip(void);
uint8_t increment(uint8_t PotNo);
uint8_t decrement(uint8_t PotNo);
uint16_t read_wiper(uint8_t PotNo);
void write_wiper(uint8_t PotNo, uint16_t value);

#endif /* MCP4xx1_H_ */

*****

/*
 * MCP4xx1.c
 *
 * Created: 10-03-2014 21:13:32
 * Author: Piyush
 */

#include <avr/io.h>
#include "MCP4xx1.h"
#include "../MyLibs/LCD/lcd.h"

uint16_t read_wiper(uint8_t PotNo)
{
    uint16_t result;
    SELECT_CHIP();
    SPDR = ((PotNo << POT_ADDR)|(3<<2));

    //wait for the SPI bus to finish
    while(!(SPSR & (1<<SPIF)));

    //check validity of result
    if ((SPDR&0x02)!=0)
    {
        result=SPDR & 0x01;           //read only the LSB
        result <<= 8;                 //shift lower 8 bits to the higher byte
    }
    else
    {
        LCDClear();
        LCDWriteString("Err reading data");
        LCDWriteStringXY(0,1,"from DigiPot : ( ");
        while(1);                     //halt
    }

    SPDR = 0;
    //wait for the SPI bus to finish
    while(!(SPSR & (1<<SPIF)));
    DE_SELECT_CHIP();

    result|=SPDR;
    return result;
}

void write_wiper(uint8_t PotNo, uint16_t value)
{
    SELECT_CHIP();

```

```

SPDR = ((PotNo << POT_ADDR))|((value>>8)&0x0002);
//wait for the SPI bus to finish
while(!(SPSR & (1<<SPIF)));

//check validity of result
if ((SPDR&0x02)!=0)
{
    SPDR=(value&0x00FF);
    while(!(SPSR & (1<<SPIF)));           //wait for the SPI bus to finish
    DE_SELECT_CHIP();                     //de-select chip
    _delay_ms(100);
    if (SPDR==0xFF)
    {
        //LCDClear();
        //LCDWriteString("Write successful");
        //_delay_ms(500);
    }
}
else
{
    LCDClear();
    LCDWriteString("Err writing data");
    LCDWriteStringXY(0,1,"to DigiPot.. : ( ");
    while(1);
}
}

uint8_t increment(uint8_t PotNo)
{
    SELECT_CHIP();
    SPDR = ((PotNo << POT_ADDR)|(1<<2));

    //wait for the SPI bus to finish
    while(!(SPSR & (1<<SPIF)));
    DE_SELECT_CHIP();
    _delay_ms(50);

    //get the received data at SDO
    if (SPDR == 0xFF)
    {
        return SUCCESS;
    }
    else
    {
        LCDClear();
        LCDWriteString("Wrong Command");
        while(1);
        return ERROR;
    }
}

uint8_t decrement(uint8_t PotNo)
{
    SELECT_CHIP();
    SPDR = (PotNo<<POT_ADDR)|(2<<2);

    //wait for the SPI bus to finish
    while(!(SPSR & (1<<SPIF)));
    DE_SELECT_CHIP();

```

```

        //get the received data at SDO
        if (SPDR == 0xFF)
        {
            return SUCCESS;
        }
        else
        {
            return ERROR;
        }
    }
}

*****
/*
 * spi.h
 *
 */

#ifndef _SPI_H
#define _SPI_H
#include <inttypes.h>

#define SPI_PORT PORTB
#define SPI_DDR DDRB
#define SPI_MOSI PB3
#define SPI_MISO PB4
#define SPI_SCK PB5
#define SPI_LSBFIRST_MASK 0b00000001
#define SPI_MASTER_MASK 0b00000001
#define SPI_MODE_MASK 0b00000011
#define SPI_SPEED_MASK 0b00000011
#define SPI_DBLCLK_MASK 0b00000001

//initialize the SPI bus
// uint8_t lsbfirst - if 0: most significant bit is transmitted first
// uint8_t master - if 1: use master mode, if 0: slave mode is used
// uint8_t mode - sets the transfer mode:
//          mode  leading clock edge  trailing clock edge
//          -----
//          0   sample (rising)   setup (falling)
//          1   setup (rising)   sample (falling)
//          2   sample (falling)  setup (rising)
//          3   setup (falling)  sample (rising)
// uint8_t clkrate - spi bus clock rate, valid speeds are 0-3
//          rate  speed
//          -----
//          0   CPUCLK/4
//          1   CPUCLK/16
//          2   CPUCLK/64
//          3   CPUCLK/128
// uint8_t dblclk - if 1: doubles the SPI clock rate in master mode
// EXAMPLE: spi_init(0, 1, 0, 3, 0)
void Init_SPI(uint8_t lsbfirst, uint8_t master, uint8_t mode, uint8_t clkrate, uint8_t dblclk);
uint8_t spi_send(uint8_t value);

#endif

*****
/*
 * spi.c

```

```

*
*/

#include <avr/io.h> //for I/O pin operations
#include "spi.h"

void Init_SPI(uint8_t lsbfirst, uint8_t master, uint8_t mode, uint8_t clkrate, uint8_t dblclk)
{
    SPI_DDR |= ((1<<SPI_MOSI) | (1<<SPI_SCK));
    SPI_DDR &= ~(1<<SPI_MISO);
    SPI_PORT |= (1<<SPI_MISO);
    SPCR = ((1<<SPĒ) | ((lsbfirst & SPI_LSBFIRST_MASK)<<DORD) | ((master &
SPI_MASTER_MASK)<<MSTR) | ((mode & SPI_MODE_MASK)<<CPHA) | (clkrate &
SPI_SPEED_MASK<<SPR0));
    SPSR = ((dblclk & SPI_DBLCLK_MASK)<<SPI2X);
}

uint8_t spi_send(uint8_t value)
{
    uint8_t result;

    SPDR = value; //shift the first byte of the value
    while(!(SPSR & (1<<SPIF))); //wait for the SPI bus to finish
    result = SPDR; //get the received data
    return(result);
}

*****

/*
 * ADC.h
 *
 * Created: 10-03-2014 21:13:32
 * Author: Piyush
 */
#ifndef _ADC_H
#define _ADC_H

void Init_ADC(void);
uint16_t ReadVoltage(uint8_t Ch);

#endif

*****

/*
 * ADC.c
 *
 * Created: 10-03-2014 21:13:32
 * Author: Piyush
 */

#include <avr/io.h>

/*
ADC Initialization
*/

void Init_ADC()
{

```

```

        ADMUX = (1<<REFS0);                // set internal AVCC as ref for
AREF
        ADCSRA = (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1); // Set ADC enable and division
factor 64
    }

/*
ADC Conversion
*/

uint16_t ReadVoltage(uint8_t Ch)
{
    ADMUX |= (Ch & 0x07);                // channel selection
    ADCSRA |= (1<<ADSC);                 // start conversion
    while (!(ADCSRA & (1<<ADIF)));        //wait for conv to complete
    ADCSRA |= (1<<ADIF);                 //clear ADIF by writing "1" to it
    return(ADC);                         //returns    all    10    bits
(ADCH+ADCL)
}

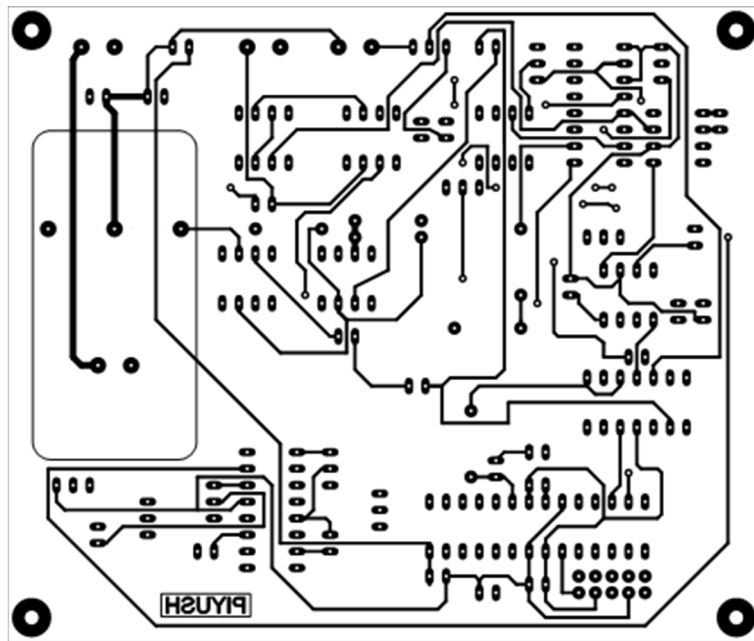
```

## APPENDIX-B

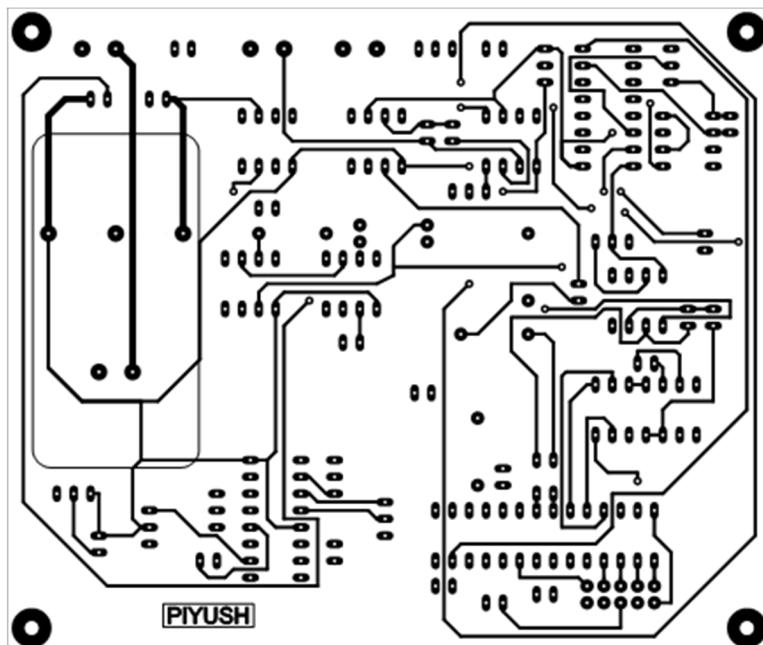
### PCB LAYOUT

Solder and component sides of the PCB layout are as shown below:-

#### SOLDER SIDE



#### COMPONENT SIDE



## **CURRICULUM VITAE**

**Name** : Piyush Kumar

**Date of Birth** : 09 Jul 1981

**Place of Birth** : Sitapur, U.P.

**Education** : B.Tech from Institute of Engineering and Technology,  
Lucknow University. 2002 Batch.

**Work experience** : 10 years working experience in radar maintenance and  
Electronic Warfare

**E-mail** : piyush.iaf@gmail.com

**Publication** : Kumar, P., George, B. and Kumar, V.J., “A simple signal  
conditioning scheme for inductive sensors”, proc. of IEEE  
Seventh International Conference on Sensing Technology  
(ICST), Dec. 3 – Dec. 5, 2013, Wellington, New Zealand,  
pp: 512-515.