COMPENSATION OF MEMS SENSORS USING NEURAL NETWORK

A PROJECT REPORT

submitted by

DARSANA U D.

(Reg. No. TKM21EEII05)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology
in

Electrical and Electronics Engineering

with specialisation in

Industrial Instrumentation and Control

Department of Electrical and Electronics Engineering

TKM College of Engineering Kollam
Kollam - 691 005

JUNE 2023

DECLARATION

I undersigned hereby declare that the project report entitled "**Compensation of MEMS Sensors using Neural Network**", submitted for partial fulfillment of the requirements for the award of degree of Master of Technology in Electrical and Electronics Engineering with specialisation in Industrial Instrumentation and Control, of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under the supervision of *Dr. Mathew P Abraham*, Internal Project Supervisor, Assistant Professor, Department of Electrical and Electronics Engineering, *Mr. Rajesh George P*, External Project Supervisor, GH, LVIS/NSEG, ISRO Inertial Systems Unit, Thiruvananthapuram, *Prof. Amal A*, Project Co-ordinator, Assistant Professor, Department of Electrical and Electronics Engineering. This submission represents my ideas in my own words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam
June 26, 2023

DARSANA U D.

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

TKM COLLEGE OF ENGINEERING KOLLAM-691 005

CERTIFICATE

This is to certify that the project report entitled " **Compensation of MEMS Sensors using Neural Network** " submitted by **DARSANA U D.** , **(Reg. No. TKM21EEII05)** of fourth semester to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Electrical and Electronics Engineering with specialisation in Industrial Instrumentation and Control, is a bonafide record of the project done by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. Mathew P. Abraham
Internal Project Supervisor
Assistant Professor
Department of Electrical & Electronics Engg
TKM College of Engineering, Kollam

Mr. Rajesh George P
External Project Supervisor
GH
LVIS/NSEG
ISRO Inertial Systems Unit, Thiruvananthapuram

Prof. Shanavas T. N
Associate Professor and PG Co-ordinator
Department of Electrical & Electronics Engg
TKM College of Engineering, Kollam

Dr. Sabeena Beevi. K
Associate Professor and Head
Department of Electrical & Electronics Engg
TKM College of Engineering, Kollam

ACKNOWLEDGEMENT

A lot of effort and hard work has been put into this project in course of its presentation. However, it would not have been possible without the kind support and help of many individuals and other sources. I would like to extend my sincere thanks to all of them. I take this opportunity to express my deep sense of gratitude and sincere thanks to all who helped me to complete this project report successfully.

I thank God Almighty for guiding me throughout the project. I express my sincere thanks to *Dr. T A Shahul Hameed*, Principal, TKM College of Engineering, for his encouragement and support in the successful completion of this project work. I thank *Dr. Sabeena Beevi. K*, Associate Professor and Head, Department of Electrical and Electronics Engineering and *Prof. Shanavas T. N*, PG Co-ordinator, Department of Electrical and Electronics Engineering for their support and help. I express my sincere thanks to *Dr. Imthias Ahamed T P*, Head of Centre for Artificial Intelligence, for his valuable help and guidance.

I am greatly thankful to my internal project supervisor *Dr. Mathew P. Abraham*, Assistant Professor, Department of Electrical and Electronics Engineering, my external project supervisor *Mr. Rajesh George P*, GH, LVIS/NSEG, ISRO Inertial Systems Unit, Thiruvananthapuram and *Ms. Mary Jermila M*, SF, NSEG, ISRO Inertial Systems Unit, Thiruvananthapuram for their supervision, assistance and helpful suggestions. I am much thankful to *Prof. Amal. A*, Project Co-ordinator, Assistant Professor, Department of Electrical and Electronics Engineering, and *Prof. Sumayya Jaleel*, Assistant Professor, Department of Electrical and Electronics Engineering, for their guidance, positive criticism and valuable comments.

Finally I thank my parents and friends who directly and indirectly contributed to the successful completion of my project presentation.

DARSANA U D.

ABSTRACT

Microelectromechanical system (MEMS) is a technology that in its most general form can be defined as miniaturized mechanical and electro-mechanical devices and structures that are made using the techniques of microfabrication. MEMS are low-cost, low to medium accuracy inertial sensors that are extensively used in the industrial as well as in the real time applications. Also, they are widely used in the satellite and the navigation field. The real time systems are greatly affected by the conditions of the external surrounding, thus the results or the responses of these systems or applications will be highly noisy signals and readings.

The temperature compensation of the sensor data is very important, and can be done with any of the techniques. The classic way is to use the filters. In this work two methods are used for the purpose one is using a filter and Neural Network(NN). The main aim of the work is the temperature compensation. The filter used, the Kalman filter(KF), is an algorithm that provides estimates of some unknown variables given the measurements observed over time and works on updation and prediction method. The other method is neural network technique in which the fitting of the neural network for the noise reduction has been done. By a proper temperature compensation, the noise reduction will also be resulted.

Contents

List of Figures

ABBREVIATIONS

ADC	Analog to Digital Converter
AI	Artificial Intelligence
ANN	Artificial Neural Network
CMOS	Complementary Metal-Oxide-Semiconductor
FPGA	Field Programmable Gate Array
IC	Integrated Circuit
IMU	Inertial Measurement Unit
KF	Kalman Filter
MATLAB	Matrix Laboratory
MEMS	Micro-electromechanical systems
NN	Neural Network
ReLU	Rectified Linear Unit
SPLL	Software Phase Locked Loop

NOTATIONS

x, y, z	Target Positions
a	Acceleration
$\mathbf{x_n}$	p-dimensional State Vector
$\Phi_{n,k}$	State Transition Matrix
I	Identity Matrix
X_i	Input
W_i	Weight
b	Bias
θ	Threshold Value
λ	Steepness Parameter
α	Learning Rate
P	Covariance Matrix
Q	Process Noise Covariance
H	Observation Matrix
R	Sensor Noise Covariance
$\mathbf{H}$	Hessian Matrix
$\mathbf{J}$	Jacobian Matrix
$\mathbf{g}$	Gradient
$\mathbf{e}$	Network Error

Chapter 1

INTRODUCTION

1.1 GENERAL BACKGROUND

Gyroscopes are the devices used for maintaining or measuring the angular velocity. These devices works on the principle of conservation of angular momentum. A mechanical gyroscope shown in Figure 1.1 works in way such that, it has a spinning rotor or a disc which is made to rotates in the space by maintaining the orientation. It consists of a wheel mounted on two or three gimbals which provides a pivoted support. This support will allow the wheels, disc or the rotor to rotates about a single axis. The gyroscopes are mainly rate sensing gyroscopes and rate integrating gyroscope. The rate sensing gyroscopes are used to find out the velocity and the rate integrating gyroscopes are used to measure the orientation. MEMS gyroscopes are rate sensing or measuring gyroscopes.

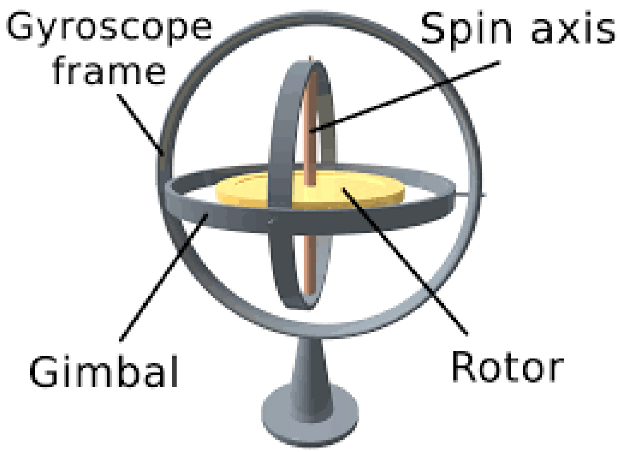

Figure 1.1: A Mechanical Gyroscope

The vibrating gyroscopes are the another type of gyroscopes used. These gyroscope usually works on the Coriolis force. The Coriolis force is a fictitious force that acts on the object in a rotating reference frame. This force arises due to the combination of inertia of the object and the rotation of the frame. This force in the case of the vibrating gyroscopes are used for finding out the velocity. These are called as the Coriolis vibrating gyroscope. The principle of operation of the MEMS gyroscope is described below using the Figure 1.2. The vibrating gyroscope has a proof mass, inertial element (sensitive element) which will make the primary oscillations or the primary mode. The primary oscillations are assumed to be oscillatory, thus the secondary oscillations will also be in an oscillatory mode. The secondary oscillation or the secondary motion is formed by the movement of the inertial element in orthogonal direction, this movement of the inertial element is caused by the Coriolis force. The vibrating gyroscopes using the MEMS technology is the MEMS gyroscopes.

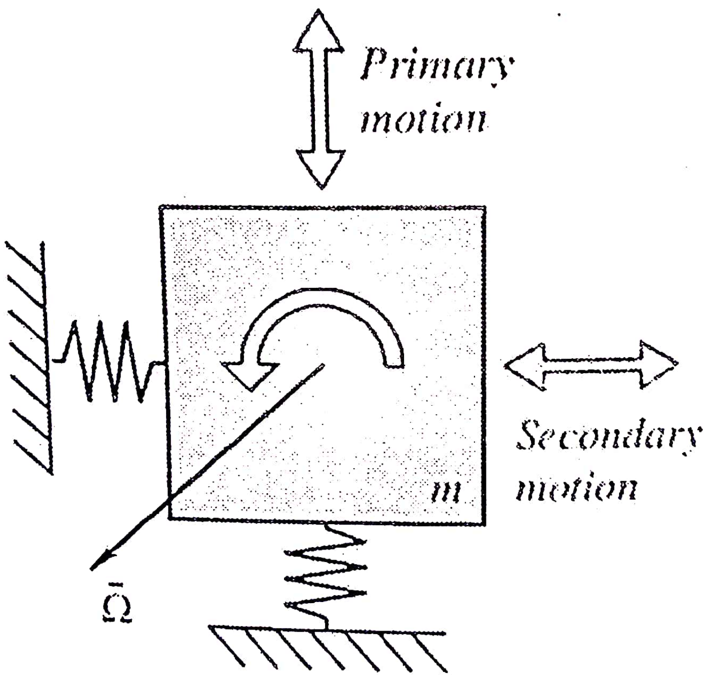

Figure 1.2: Principle of Operation of vibrating gyroscope

MEMS sensors are the angular rate sensors, used for measuring the velocity. Accelerometers and the gyroscopes contribute to the majority of the MEMS sensors. Microelectromechanical systems (MEMS) is a technology that in its most general form can be defined as miniaturized mechanical and electro-mechanical elements (i.e., devices and structures) created through the process of micro-fabrication. MEMS are inertial sensors with low to medium accuracy and low cost that are employed in a wide variety of industrial applications. These sensors are used to both detect and measure external stimuli like pressure. They then react to the recorded pressure by performing a few mechanical operations. Micro-Electro-Mechanical Systems (MEMS) combine electrical circuitry with mechanical transducers. They have been widely used in many different industries, including the automotive, biomedical, and military systems. More than 20% of the MEMS market is taken up by inertial MEMS sensors, such as MEMS accelerometers and gyroscopes. These sensors are extensively used in the field of satellite and navigation due its compatible size.

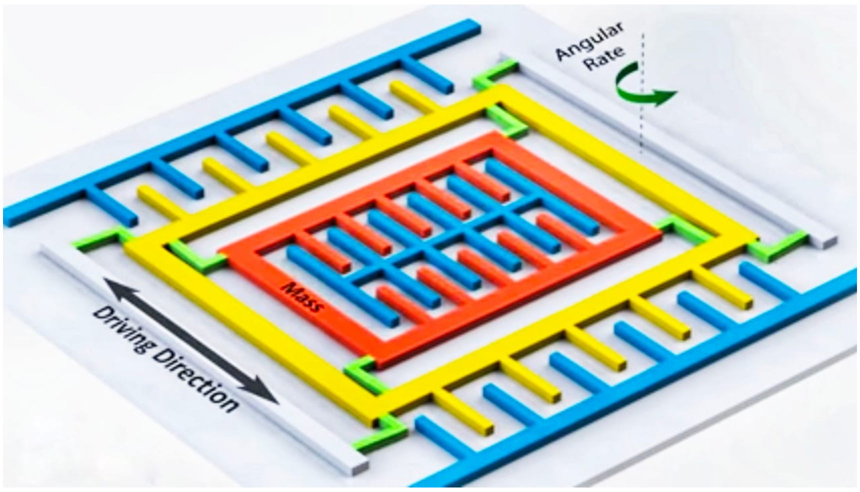

Figure 1.3: Working of MEMS gyroscope

The accelerometer and the gyroscopes works on the same method, that the capacitive changes in the suspended mass will make the velocity difference which will be measured. The Figure 1.3 shows the working of a MEMS gyroscope, when a mass is moving in a direction with a velocity and an external angular rate is applied, a force is occurred will cause a perpendicular displacement, this will cause the capacitance changes. This will be measured. The Figure 1.4 shows the force acting on the gyroscope. The mass have a velocity as shown in red arrow, if an angular rate shown in green colour is applied, a force ie., the fictitious Coriolis force [1] will be acted in a perpendicular direction as shown with the blue arrow. The MEMS gyroscopes are commonly used in the navigation systems to find out the orientation and the angular velocity, the motion sensing such as in smartphones, gaming controllers, the next application of the gyroscope is in the field of robotics, where the autonomous vehicles and the unmanned aerial vehicles use this device to be in the orientation and the stabilization.

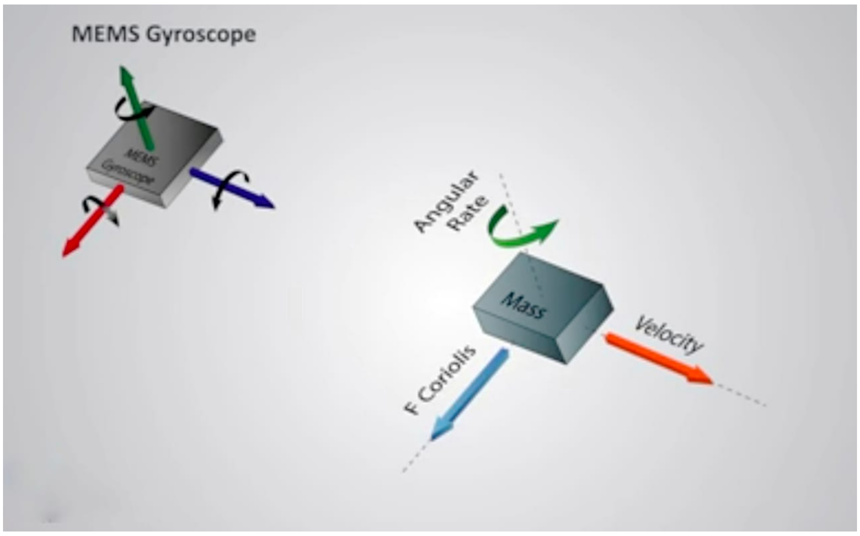

Figure 1.4: Principle of MEMS gyroscope

The sensors are mostly employed in the real time systems or in the real time application thus the noise level in the system will be high. The noise will reduce the performance. These can be due to the measurement, or the external environment and should be reduced. Along with the noise reduction the compensation is also an essential task. The gyroscopes' performance can be affected by the temperature. The temperature changes will cause the performance degradation and results in low accuracy. The temperature variations will cause the changes in the resonant frequency, this will cause the changes in the damping and the noise level. Thus, to obtain a noise reduction a proper compensation should be given.

1.2 OBJECTIVE

Since the MEMS sensors are implemented in the real time systems, the environment and the surrounding in which the sensor implemented will affect the performance of the sensors. The temperature variations in the sensor data will results in the wide changes in resonant frequency and the mechanical property changes of the sensing element. And also this will causes an increased level of the noise in the results. Thus, the temperature compensation of the data is important. So, the main objective of this work is to compensate the temperature of the sensor data, thus, a filtering can be achieved. For that, techniques like Kalman filter and neural network are used here.

1.3 SCOPE

Gyroscopes and accelerometers, which make up the majority of MEMS sensors, are becoming more and more advantageous in engineering applications; examination of these sensors is crucial. The MEMS sensors have taken the place of the earlier sensors due of their compact design. The requirement has grown as the world transitions to a nano environment.
In the navigation, satellite and launch vehicle field the size of the components used is a major concern. The use of the Inertial Measurement Units (IMU) is less advised due to the increased size of the components, also the parts or the components used in this field need to be worked on a high risk environment. The earlier available sensors are not up to the mark when coming to the working conditions. The MEMS sensors can work on the environment with high risks like high temperature, high vibrations, tensions etc. Thus, as a solution MEMS sensors can be used as these sensors offers a medium class of performance in harsh environment.

1.4 SCHEME OF PROJECT WORK

The report is organized in 5 chapters. Chapter 1 titled by Introduction includes general background, objective, scope and scheme of project work. Chapter 2 is the detailed analysis of the previous papers related with the filtering, compensation, MEMS sensors and neural network, the literature that I have used in this work are cited with adequate references. Chapter 3 titled as Kalman filter, where the theory of the filter, the workdone and the simulation results are discussed. Also the algorithm and the derivation of the Kalman filter equations are described. Chapter 4 has the detailed description of the neural network, its training, history of the artificial neural network, the workdone with the neural network. Along with these details the training results and the training algorithm are described deeply in this chapter. Chapter 5 is the Conclusion and Future Scope, in the conclusion section the summary of the work is given. The future scope includes the possibilities of advanced works in this topic.

Chapter 2

LITERATURE REVIEW

2.1 OVERVIEW

This chapter gives the previous works and studies related with MEMS, MEMS sensors, Kalman filter, noise filtering, temperature compensation and neural networks. The below listed literature gives the idea of using the Kalman filter and neural network for the temperature compensation and filtering.

2.2 PREVIOUS WORKS ON MEMS AND NOISE FILTERING

MEMS is the technology of microscopic devices, especially those with moving elements, includes microelectromechanical systems (MEMS) and the associated micro mechatronics and micro systems. It mainly sizes in the range of 20 micrometers to millimeters and the MEMS are made up of components in a size range of 1 to 100 micrometers. They usually consists of a central unit to processes data and several other components to interact with surroundings. MEMS has very large surface area to volume ratio, and this force the ambient electromagnetism and fluid dynamics are important factors to be considered in the design procedure. The MEMS will be an interface between the IC world and the microtechnology.
The literature [2] gives the different type of sensors and the recent advances in the field of MEMS. By this work the mechanical application of the sensors are given, the mechanical sen-

sors commonly used are described. These can be used for measuring velocity, displacement, acceleration, force, deformation, stress, and strain. Sensing techniques are typically based on piezoelectric, capacitive, electromagnetic and piezoresistive principles.[3] MEMS its fabrication, design and the applications are discussed in this work. The micromachining technology is also discussed here. The history of the micromachining technology and MEMS technology are given. The basic introduction, application and functions of the MEMS technology can be understood from this literature.
In [4] design, challenges and applications of the micromachined gyroscopes are given. By the literature, automotive field, medical field, navigation, satellite and GPS systems,robotic field. These gyroscope normally works with the Coriolis force, it is actually a fictitious force acting perpendicular in an object on a rotating frame.The basic details of a MEMS gyroscope with the principle of operation can be understood from the paper. The MEMS gyroscopes are used in the field of industrial application and consumer applications like smartphones, gaming controllers, navigation and satellite field [5]. Commercially successful devices and systems that use microfabrication and MEMS technologies include many microsensors (e.g., inertial sensors, pressure sensors, magnetometers, chemical sensors etc), microactuators (e.g., micromirrors, microrelays, microvalves, micropumps etc) and microsystems (e.g., chemical analysis, sensor-feedback controlled actuators etc). Most successful MEMS product exploit any one of the below properties:

1. Advantageous scaling properties
2. Batch fabrication
3. Circuit integration

[6] gives the designing and the analysis of the MEMS gyroscope, particularly the tuning fork gyroscope is designed and analysed. A brief introduction to the modelling equations and the derivations can be taken and understood from this literature. The literature [7] gives a control loop strategy for the MEMS sensor. Two different algorithm for Software Phase Locked Loop (SPLL) to implement the frequency tracking is compared here.
The article sensing and control of accelerometers [8] give the basic idea on the noises in the MEMS accelerometers and the control strategies. Here a Kalman filter is implemented for a MEMS capacitive accelerometer in order to filter out two major noise sources in the accelerometer, which are electronic noise and thermal mechanical noise (Brown noise). Kalman filter as an observer to estimate the states of accelerometer through measured outputs, and then employ

the estimated states which have much lower noise influence to close the system. This method not only gives a noiseless estimated output, but also improves the control performance. Brownian noise caused by damping effect and the electronic noise from CMOS readout circuits are two major noise sources in capacitive accelerometers. Besides noise, the sensing accuracy of a micro-machined accelerometer is also limited by the nonlinearities and system uncertainties due to fabrication imperfections. Therefore, a feedback controller is essential for an accelerometer to compensate for the fabrication imperfections and to improve its performance. the Kalman filter modelling is based on the force-to-rebalance based method. The application of a Kalman filter to a system can be studied from this article. The literature [9] gives another ides of optimal filtering with Kalman filter, from this piece of work the advanced idea is collected to the programming. Here an integrated Kalman filter with six states is designed for the accuracy improvement. The technique is that the correlation between the elements in the array is the basis of the improving the accuracy. MEMS integrated array design is followed with the optimal filtering.

[10] gives the application of a Kalman filter to a MEMS gyroscope to filter out the noise in the system. The true angular rate signal was directly modeled to obtain an optimal estimate and make a self-compensation for the gyroscope without needing other sensor's information, whether in static or dynamic condition. The approach of direct modeling for the angular rate signal could reduce the measurement noise and improve the accuracy of the single MEMS gyroscope considerably. The greatest advantage of such an approach is that an estimation of the bias drift can be achieved to compensate for the outputs of gyroscope.Both the noise and the drift are compensated and reduced in this method. And a discrete time Kalman filter is designed on this article.

The article [11] describes the noises in a MEMS gyroscope. Here basically mechanical and thermal noises are discussed. Vibrating gyroscopes are used to determine rotation rate by sensing Coriolis force. The effects of the noise and the way to eliminate it are given in this article.

The application of MEMS in the aerospace navigation is briefly described this paper, [12]. The MEMS performances should remain stable for its life duration in order to avoid costly periodic ratings. The reliability needs to be very high for safety reasons. High analog electronic sensitivity and a very efficient modeling of the sensors is required to improve performance forecast. In some cases optical detection is preferred. Navigation systems use mainly pressure sensors, accelerometers, and rate gyros. High accuracy and precision in output can be obtained by

introducing the MEMS in the navigation field. The article [13] High sensitivity sensors are vulnerable to noise when reading data from environment. External disturbances can be vibration, pressure; even though the noise affected these sensors give high system performances. Here also a Kalman filter is designed to filter out the noise. This design is implemented on Aurdino to reduce the noise of Inertial Measurement Unit(IMU).
[14] also gives a brief idea on the MEMS gyroscopes' noise reduction. Basically, two methods are there for noise reduction in MEMS gyroscope : first one is to improve the structure of the gyroscopes to reduce the structural hardware errors and the other method is using any of the filtering algorithm. Mostly the models are established by time series analysis of existing data, which requires that the data to be steady, normal and zero mean. Thus, here a model is established which can be applied to most of the dynamic data of the gyroscopes. The data collected were the gyroscope at stationary, to this data Kalman filter is applied, static and dynamic data test are carried out.

2.3 PREVIOUS WORKS ON KALMAN FILTER

For a better understanding of Kalman filter [15] [16] were used as a reference. The two mentioned literature; one is a book which will give an idea on the Kalman filter, its applications. The book has explained everything from its base, from the history. [17] is a recently published book in which the Kalman filter from the scratch can be studied. This book contains numerical examples for each cases by using the Kalman filter. The derivations of each equation with fundamentals were discussed and explained in this book.
The renowned paper by R.E. Kalman outlining a recursive solution to the discrete-data linear filtering issue was published in 1960. Since then, the Kalman filter has been extensively researched and used, particularly in the field of autonomous or assisted navigation, thanks in large part to advancements in digital computing. The least-squares approach has an effective computational (recursive) solution provided by the Kalman filter, a collection of mathematical equations. The filter is extremely potent in a number of ways, including the ability to enable estimations of past, present, and even future states even when it is unknown exactly what the modelled system will actually be like. A practical introduction to Kalman filter is given in the literature [16].

The Kalman filter is a minimum variance estimation in which the unknown quantities or the variables can be estimated. This type of filter is available in the form of extended, unscented based on the applications. The extended Kalman filter is applied to the nonlinear systems. Unscented Kalman filter was proposed by Julier and Uhlmann based on the unscented transform and the sampling strategy.The sampling points in the unscented KF is small.The article [18] gives the comparison between these types. Based on the system complexity ans structure the use of the filter varies.

2.4 PREVIOUS WORKS ON TEMPERATURE COMPENSATION

Temperature compensation is a technique for modifying a system's performance to account for effects caused on by temperature variations. The article [19] gives a method of temperature compensation within a range of $40°C$ to $60°C$. By this work the temperature compensation is done in two ways: Firstly, a temperature test is done to describe the scale factor's temperature curve which is the foundation of the compensation. Then, the parameter values are configured in the circuit with different compensation methods. Next, the temperature tests are arranged to verify the compensation results. Finally, the bias compensation method is implemented based on the most effective scale factor compensation circuit. The article [20] describes the temperature compensation of a low temperature bias drift MEMS based gyroscopes. A frequency synthesizer is used for obtaining the absolute temperature, which is implemented by the FPGA. From the experimental results the temperature compensation of the system about $0.2°C$.
MEMS gyroscopes are a type of angular rate sensor which is widely used, but have errors in practical applications. The temperature and the temperature gradient is compensated and controlled in the article [21]. Here a hardware structure with a data acquisition system is provided for the transmission and collection of the data signals. From the work a conclusion on the zero bias is obtained as it is dependent on the temperature and the temperature gradients, the repeatability error can be reduced and a theoretical basis for the accurate compensation of the gyroscope in the practical application is also provided.
A drive mode vibration characteristics based temperature compensation method is described in [22]. Here a dual mass and tuning fork structure of the MEMS gyroscope is investigated and

a the scale factor temperature compensation circuit is designed. The paper concludes that the variation of the scale factor improved from 3.680% to 1.577% with a temperature range from $-40°C$ to $60°C$ and the bias improved from 3.880% to 1.930%.

In the literature mentioned above, a separate hardware is to be implemented with the system for the temperature compensation. This hardware implementation will make the system uneconomical and bulky, thus, another methods should be used, ie., by using the fitting in software. The below literature, are meant for these purpose. The polynomial fitting for achieving the compensation is one of the method, which is commonly used in the labs. [23] gives such an idea. Here along with the quadratic polynomial fitting, the cubic spline interpolation is also used. The pressure sensor is analysed here. The readings of the pressure sensor and the temperatures detected were the ADC used will convert the the data to digital one which is used with the microprocessors for the compensation. Despite this it has some disadvantages like the limited accuracy as the single polynomial function will not be able to accurately capture and model the temperature dependent effects, other disadvantage is overfitting, this method will capture the noise than the trend, thus an overfitting will be resulted.

The literature [24] gives the idea of using the neural network for the temperature compensation. From this work this ides is developed to the results that is obtained in this work. By this paper, a radial basis function neural network is used for the purpose. Both the sensors were compensated using this method, the improvement factor is also given. The use of neural network for the compensation is not a new one, as the paper [25] gives the method for a fibre optic gyroscope. In our compensation method, a simple feedforward network is used for the training.

2.5 PREVIOUS WORKS ON NEURAL NETWORK

As the world is moving to an AI scenario, the importance of the artificial neural network and the machine learning increased. Any filtering can be done by using these machine learning techniques. An artificial neural network is an informative processing model which is inspired from the biological nervous system.

The book [26] gives a clear cut idea about the neural network basics and its simple programmings. The book tells us everything related with the neural network from the history. In this literature the hybrid technology of the neural network and the genetic algorithm are explained. From the book the neural network, its training, parameters in the training like learning rate,

momentum were studied.
The literature [27] gives an idea on the noise filtering of the data using the neural network. This paper combines the back-propagation network with a relaxation mechanism for training. A learning algorithm is also proposed to update the probability based on the current value.
[28] the compensation strategy for the compensation of the temperature drift is established. From this work, the idea of using the back-propagation for the compensation is used and it is studied. The paper gives the temperature characters of the MEMS gyroscope, from the analysis of the paper, the method of using the back-propagation is superior to the method least square method. A developed MEMS IMU is used for the experiment. A three gyroscope is placed orthogonally. The scale factor is not considered in this literature, it is a limitation. Dynamic cases the scale factor should be considered, as the normalization of the results is important.
The network's utility in predicting noise from urban traffic is presented in the research [29]. A back-propagation network is considered for the prediction of the noise. A collective data of every location mainly commercial, residential and industrial areas with a four subsequent time schedule is used for this process. Sound pressure level is also associated with it. The paper concluded that the method used is helpful in the prediction of the noise in the traffic data signal. The training function used in the work is Levenberg-Marquardt training algorithm. [30] is having the algorithm, the least square technique which has the variation from the class least square technique. usually, the least square method uses steepest gradient descent or the Taylor expansion, but in this method, an interpolation between the two method is caused, this method is called as the neighborhood method. This interpolation will give the adequate representation of the nonlinear system.

2.6 SUMMARY

This chapter describes the previous work related with the MEMS, MEMS sensors, Kalman filter (KF),neural network. The work focusing on the noise filtering by using the Kalman filter and the neural network are also studied. The temperature compensation is the ultimate aim of the work, thus the literature related with the basics of temperature compensation, temperature compensation of gyroscope with the implementation of certain hardware is also studied. The implementation of a separate hardware for the compensation is uneconomical, thus software related works are needed, the papers on the neural network and polynomial fitting were also studied.

Chapter 3

KALMAN FILTER

3.1 OVERVIEW

This chapter discuss about the Kalman filter, its steps, algorithm and the work done with the filter. The Kalman filter is a recursive predictive technique and a filter, which uses the recursive algorithm. The dynamic system can be mainly disturbed by white noise, it (KF) estimates the system dynamics and improve the noise reduction capabilities. The steps included in the Kalman filter for the filtering are

1. Prediction

2. Correction

The correction is made with predicted results.

3.2 NEED FOR PREDICTION

As mentioned above the Kalman filter is a prediction method, it will predict the next state based on the measured parameters. So, why do we need the prediction or the requirement of the prediction is an important question, and it should be answered. This can be explained with an example of radar, consider a track cycle of 5 seconds, ie., in every 5 second the radar sends a beam in the direction of the target. After sending this beam the radar determines the current state or the current position and velocity of the target. The radar also predicts or estimate the target position at the next beam. This can be written by using Newton's motion equation,

$$x = x_0 + v_0 \Delta t + \frac{1}{2} a \Delta t^2 \tag{3.1}$$

where x is the target position, x_0, v_0 is the initial position and initial velocity respectively and a is the target acceleration.

The current state is the input to the prediction algorithm and the following states are the algorithm's output. In three dimension, 3.1 will be written as,

$$\begin{aligned} x &= x_0 + V_{x_0} \Delta t + \frac{1}{2} a_x \Delta t^2 \\ y &= y_0 + v_{y_0} \Delta t + \frac{1}{2} a_y \Delta t^2 \\ z &= z_0 + v_{z_0} \Delta t + \frac{1}{2} a_z \Delta t^2 \end{aligned} \tag{3.2}$$

The above equation 3.2 is the dynamic model which gives the relationship between the input and the output. If the current state and the dynamic state are known then the prediction will be easier. Well the radar readings cannot be accurate at all the time and has errors in the readings and measurement, these errors are random errors. This error depends upon many parameters such as radar calibrations, beam width, signal to noise reduction ratio of the returned echo. These random errors are called as the measurement noise. Several external factors such as wind, air turbulence, pilot maneuvers also affect the target position, this will cause another class of error, dynamic model error called as the process noise.

Due to the measurement and the process noise the estimated target position can be far away from the actual target position. In this case, the beam will be send to a wrong direction and the target will be missed. Thus the prediction is important, an algorithm should be used which consider both process as well as the measurement noise or the uncertainties to improve the tracking performance should be used.

3.3 STATE SPACE MODEL

The Kalman filter is based on a state/measurement model of the form;

$$\mathbf{x}_{n+1} = A_n \mathbf{x}_n + B_n \mathbf{u}_n + G_n \mathbf{w}_n \tag{3.3}$$

$$\mathbf{y}_n = C_n \mathbf{x}_n + \mathbf{v}_n \tag{3.4}$$

where $\mathbf{x}_n$ is a p-dimensional state vector and $\mathbf{y}_n$, an r-dimensional vector of observations. The equation 3.3 is the state model and 3.4 is the measurement model. The p×p state-transition matrix A_n and r×p measurement matrix C_n may depend on time n. The signals $\mathbf{w}_n$, $\mathbf{v}_n$ are assumed to be mutually-independent, zero-mean, white noise signals with a known covariance matrices of Q_n and R_n.

$$\begin{aligned} E\left[\mathbf{w}_n\mathbf{w}_i^T\right] &= Q_n\delta_{ni} \\ E\left[\mathbf{v}_n\mathbf{v}_i^T\right] &= R_n\delta_{ni} \\ F\left[\mathbf{w}_n\mathbf{v}_i^T\right] &= 0 \end{aligned} \tag{3.5}$$

where $\mathbf{u}_n$ is the deterministic input. The iteration will start at time n=0 and the initial state vector will be independent of w_n, v_n and assumed to be random. The iterated solution of the state equation may be obtained with the help of the corresponding p×p state-transition matrix $\Phi_{n,k}$ defined as follows:

$$\begin{aligned} &\Phi_{n,k} = A_{n-1}\cdots A_k,\ \text{for } n > k \\ &\Phi_{n,n} = I \\ &\Phi_{n,k} = \Phi_{k,n}^{-1}, \quad \text{for } n < k \end{aligned}$$

(3.6)

where I is the p×p identity matrix and the third equation is valid only if the inverse exists.

We need to estimate the state vector x_n from the observation signal y_n and the Kalman filter estimated will be the optimum linear combination of the observations that minimizes the mean square estimation error. Estimates, estimation errors, and mean-square error covariances can be expressed as,

$$\begin{aligned} &\hat{\mathbf{x}}_{n/n-1} = \mathrm{Proj}\left[\mathbf{x}_n \mid Y_{n-1}\right] \\ &\mathbf{e}_{n/n-1} = \mathbf{x}_n - \hat{\mathbf{x}}_{n/n-1} \\ &P_{n/n-1} = E\left[\mathbf{e}_{n/n-1}\mathbf{e}_{n/n-1}^T\right] \end{aligned} \tag{3.7}$$

$$\begin{aligned} &\hat{\mathbf{x}}_{n/n} = \mathrm{Proj}\left[\mathbf{x}_n \mid Y_n\right] \\ &\mathbf{e}_{n/n} = \mathbf{x}_n - \hat{\mathbf{x}}_{n/n} \\ &P_{n/n} = E\left[\mathbf{e}_{n/n}\mathbf{e}_{n/n}^T\right] \end{aligned} \tag{3.8}$$

3.3.1 Estimation Algorithm

Initialize in time by: $\hat{\mathbf{x}}_{0/-1} = \overline{\mathbf{x}}_0, P_{0/-1} = \Sigma_0$

At time n, $\hat{\mathbf{x}}_{n/n-1\times} P_{n/n-1}$, $\mathbf{y}_n$ are available,

$D_n = C_n P_{n/n-1} C_n^T + R_n$ innovations covariance

$G_n = P_{n/n-1} C_n^T D_n^{-1}$ Kalman gain for filtering

$K_n = A_n G_n = A_n P_{n/n-1} C_n^T D_n^{-1}$ Kalman gain for prediction

$\dot{\mathbf{y}}_{n/n-1} = C_n \dot{\mathbf{x}}_{n/n-1}$ predicted measurement

$\varepsilon_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1} = \mathbf{y}_n - C_n \hat{\mathbf{x}}_{n/n-1}$ innovations sequence

Measurement update / correction:

$\hat{\mathbf{x}}_{n/n} = \hat{\mathbf{x}}_{n/n-1} + G_n \boldsymbol{\varepsilon}_n$ filtered estimate

$P_{n<n} = P_n(n-1) - G_n D_n G_n^T$ estimation error

Time update / prediction:

$\hat{\mathbf{x}}_{n+1//n} = A_n \hat{\mathbf{x}}_{n/n} = A_n \hat{\mathbf{x}}_{n/n-1} + K_n \boldsymbol{\varepsilon}_n$ predicted estimate

$P_{n+1/n} = A_n P_{in/n} A_n^T + Q_{in}$ prediction error

The quantity D_n represents the innovations covariance matrix. Figure 3.1 shows the block

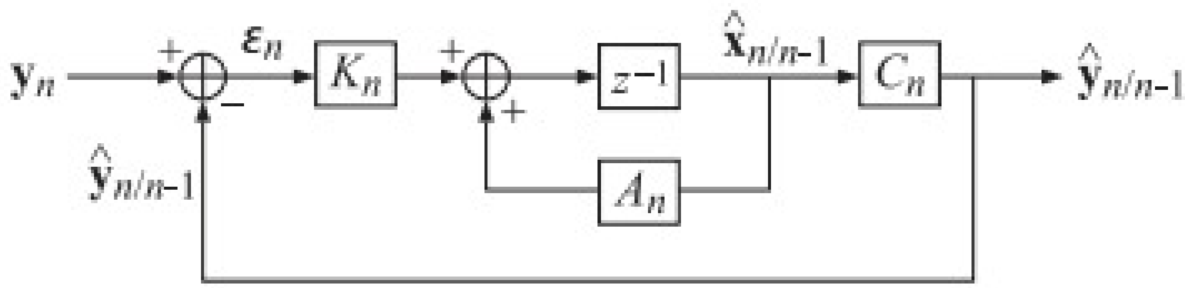

Figure 3.1: Kalman Filter Realization

diagram of the Kalman filter realization of the equations 3.7, 3.8. The innovations sequence $\varepsilon_0, \varepsilon_1, ..., \varepsilon_n$, constructed recursively by the algorithm, represents the Gram-Schmidt orthogonalization of the observations $\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_n$in the sense that the innovations form an orthogonal basis for the observation subspace $\mathbf{Y}_n$.

The information form of the updating equation,

$$P_{n/n}^{-1} \hat{\mathbf{x}}_{n/n} = P_{n/n-1}^{-1} \hat{\mathbf{x}}_{n/n-1} + C_n^T R_n^{-1} \mathbf{y}_n \tag{3.9}$$

The difference equation for the prediction estimation can be written as.

$$\begin{aligned}\hat{\mathbf{x}}_{n+1/n} &= A_n\hat{\mathbf{x}}_{n/n-1} + K_n\varepsilon_n = A_n\hat{\mathbf{x}}_{n/n-1} + K_n\left(\mathbf{y}_n - C_n\hat{\mathbf{x}}_{n/n-1}\right) \\ &= \left(A_n - K_nC_n\right)\hat{\mathbf{x}}_{n/n-1} + K_n\mathbf{y}_n\end{aligned} \tag{3.10}$$

The error covariance update equations may be combined into a single equation known as the discrete-time Riccati difference equation, to be initialized at $P_{0/-1} = \Sigma_0$,

$$\begin{aligned}P_{n/n} &= P_{n/n-1} - P_{n/n-1]}C_n^TD_n^{-1}C_nP_{w/n-1} \\ P_{n+1/n} &= A_nP_{n/n}A_n^T + Q_n\end{aligned} \tag{3.11}$$

$$P_{n+1/n} = A_n\left[P_{n/n-1} - P_{n/n-1}C_n^T\left(C_nP_{n/n-1}C_n^T + R_n\right)^{-1}C_nP_{n/n-1}\right]A_n^T + Q_n \tag{3.12}$$

The closed-loop transition matrix can be written as,

$$F_n = A_n - K_nC_n = A_n\left[I + P_{n/n-1}C_n^TR_n^{-1}C_n\right]^{-1} \tag{3.13}$$

3.4 WORK DONE WITH KALMAN FILTER

The Kalman filter is used for the noise reduction and the compensation in our data. The filter produces the estimates of the hidden variables based on the inaccurate and uncertain measurements. It will predict the future system based on the past estimation.

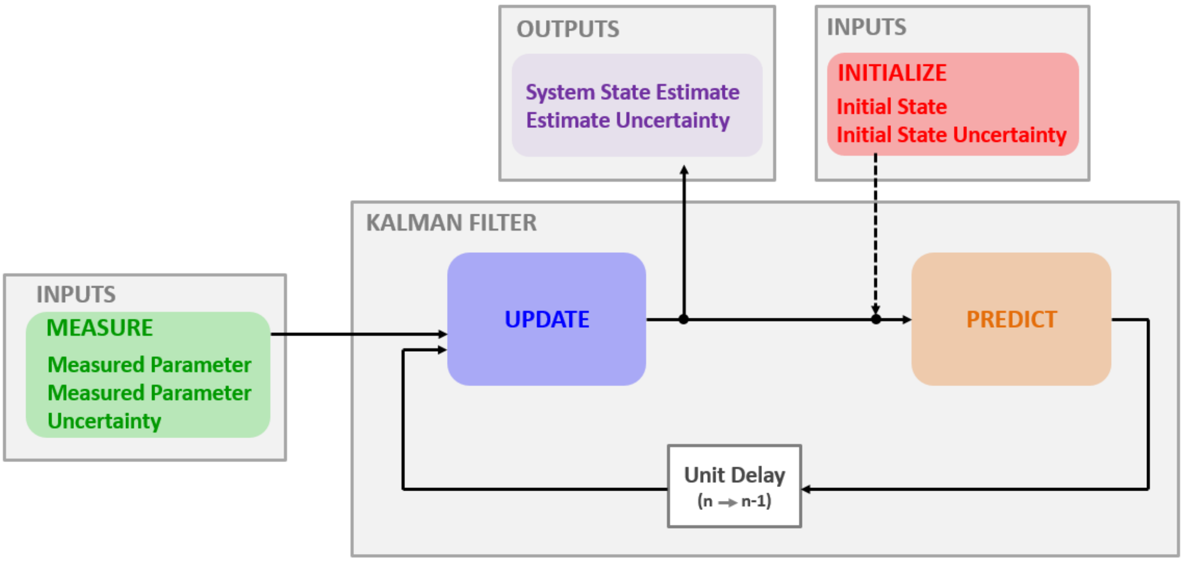

Figure 3.2: Kalman Filter Algorithm

Figure 3.2 shows the block diagram of the Kalman filter. The block input (initialize) is the first

step in which the initial guess of the uncertainty, system state is done. The next step is measuring, where the parameters are measured, the measured values and parameters are meant for the updation in next stage. In this updation process a Kalman gain is calculated. The updated parameters and the output estimates are made for the prediction. The filter calculates the predicted state and extrapolate the uncertainty.
The steps or the process in the Kalman filter is given below,

1. Initialization is the first step in which the initial state and the uncertainty is initialized.

2. The measurement step which provides the parameters measured system state and the measurement uncertainty.

3. State update the step in which the estimate process starts.

4. Prediction is the process which extrapolates the current estimate and the uncertainty to the next state.

The equations in a general form can be written as follows;

- Predicted state = state transition * previous state

- Predicted Covariance = State transition * Previous Covariance + Transpose of State Transition + Process Covariance

- Kalman gain = Predicted Covariance * Transpose of Observation Matrix * inv(Observation Matrix * Predicted Covariance * Transpose of Observation Matrix + Measured Variance)

- State estimated = State Predicted + Kalman gain * Residual

- Covariance estimated = (Identity matrix - Kalman gain * Observation Matrix) * Predicted Covariance

The program flow along with the results are discussed in the below.

3.5 SIMULATION RESULTS

The Kalman filter is used here to filter out the noise present in the sensor data and for compensation. It works on the prediction algorithm. A data is fed to the filter for the compensation, here

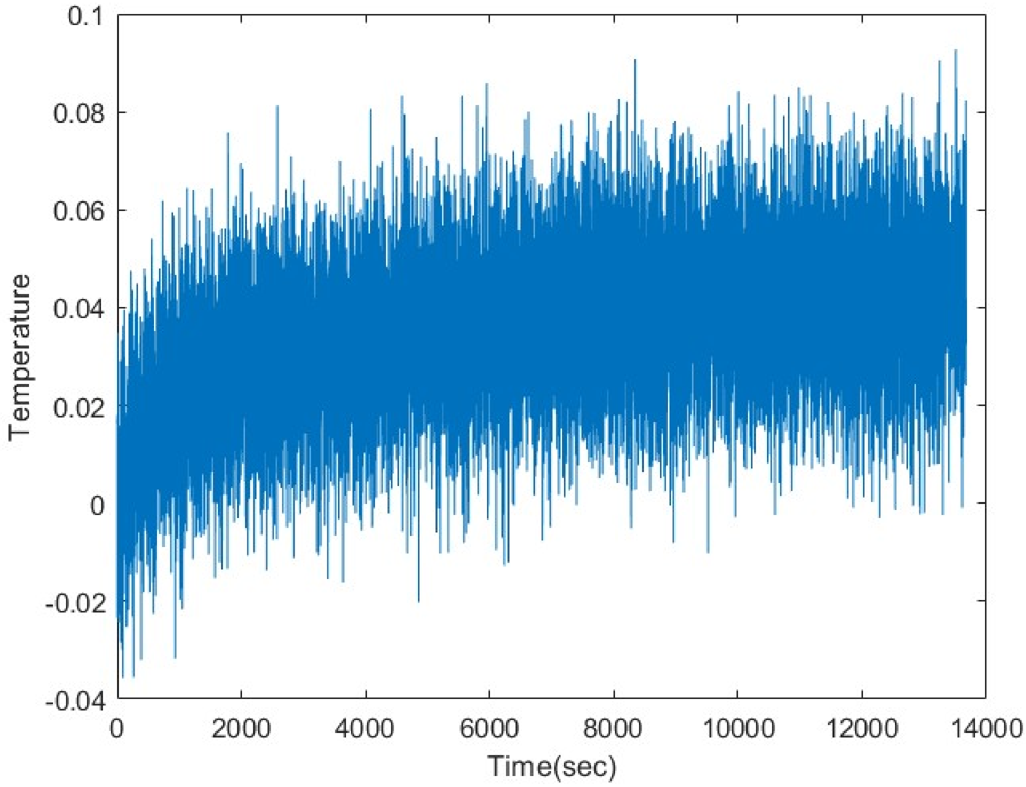

Figure 3.3: Temperature Plot

a sensor data, temperature is used. As given above, the first step is the initialization in which the initial guess of the uncertainty, system state is done. The next step is measuring, where the parameters are measured, the measured values and parameters are meant for the updation in next stage, in a program flow the matrices like the covariance matrices, observation matrix are also needed to be given as the input. The updation process is the next step in the flow, there Kalman gain is calculated. Kalman gain is a mathematical formula that used in the Kalman filter for estimation. The gain value is computed using the covariance matrices, prior estimates and the system. The updated parameters and the output estimates are made for the prediction, a new predicted state is estimated there in this step.

Figure 3.3 shows the temperature curve of the sensor, which is unfilteref and uncompensated, this is the exact depiction of the data from the experimental analysis, this curve is a temperature (in volt) vs time (in seconds) graph. It is clear from the curve that, the values are changing in a range from -0.04 to 0.1 volts. The shattered lines on the curve indicates the noise present in the data and it is uncompensated. For the applications the curve should have minimised noise, and should be compensated. This can be done with a Kalman filter.

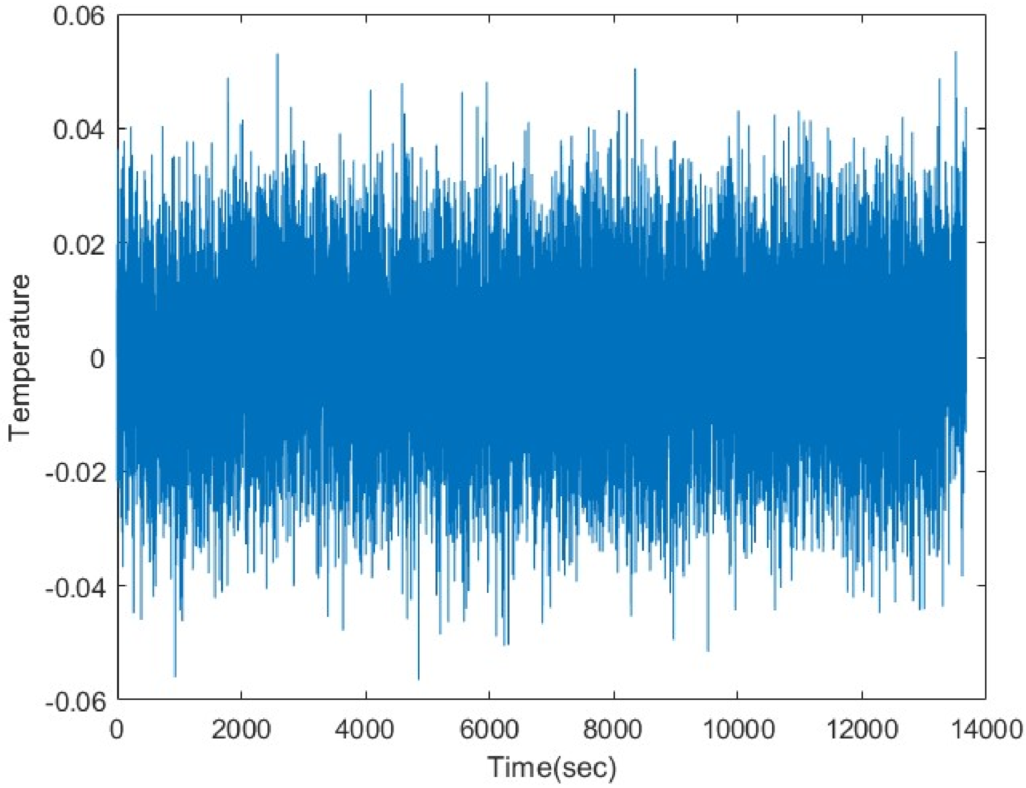

Figure 3.4: Compensated and filtered with Kalman Filter

As the filtering technique a Kalman filter is used. The Kalman filter is a recursive algorithm based on the prediction. The filter updates the parameters by using the data provided and a final prediction is obtained. This is shown in the Figure 3.4. The uncompensated temperature plot ranges from -0.04 to 0.1, this range is made to a reasonable value as approximately from -0.06 to 0.06. The variations in the curve is compensated to an extend. This two plots 3.3, 3.4 are made to a subplot or the comparison plot for the easy observation of the changes occurred, the Figure 3.5. From the comparison plot, the filtere output has almost a constant level bandwidth which is not in the uncompensated one. The system estimate is shown in the Figure 3.6 the rated voltage is plotted against the time.

The results shown are with the temperature compensation. The terms in the system model will be having a temperature term. The covariance matrices and the observation matrix is given below,

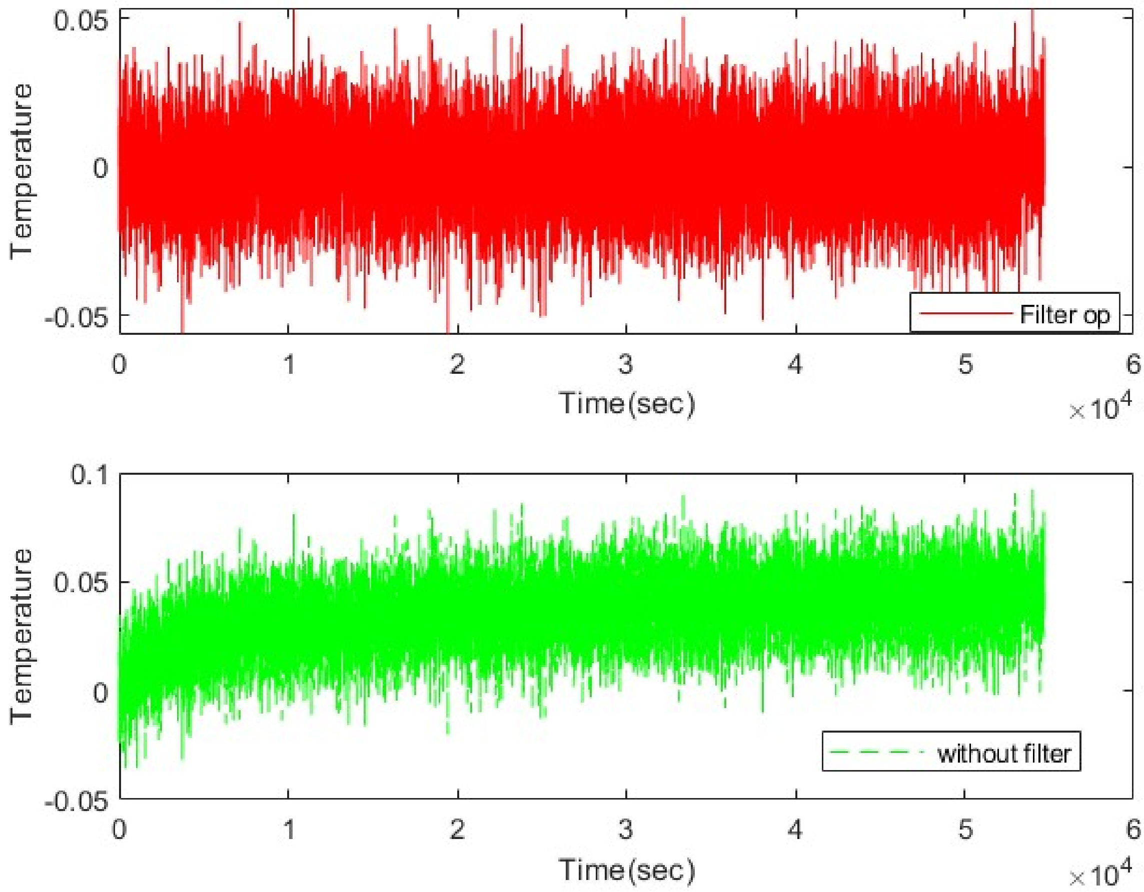

Figure 3.5: Comparison Plot

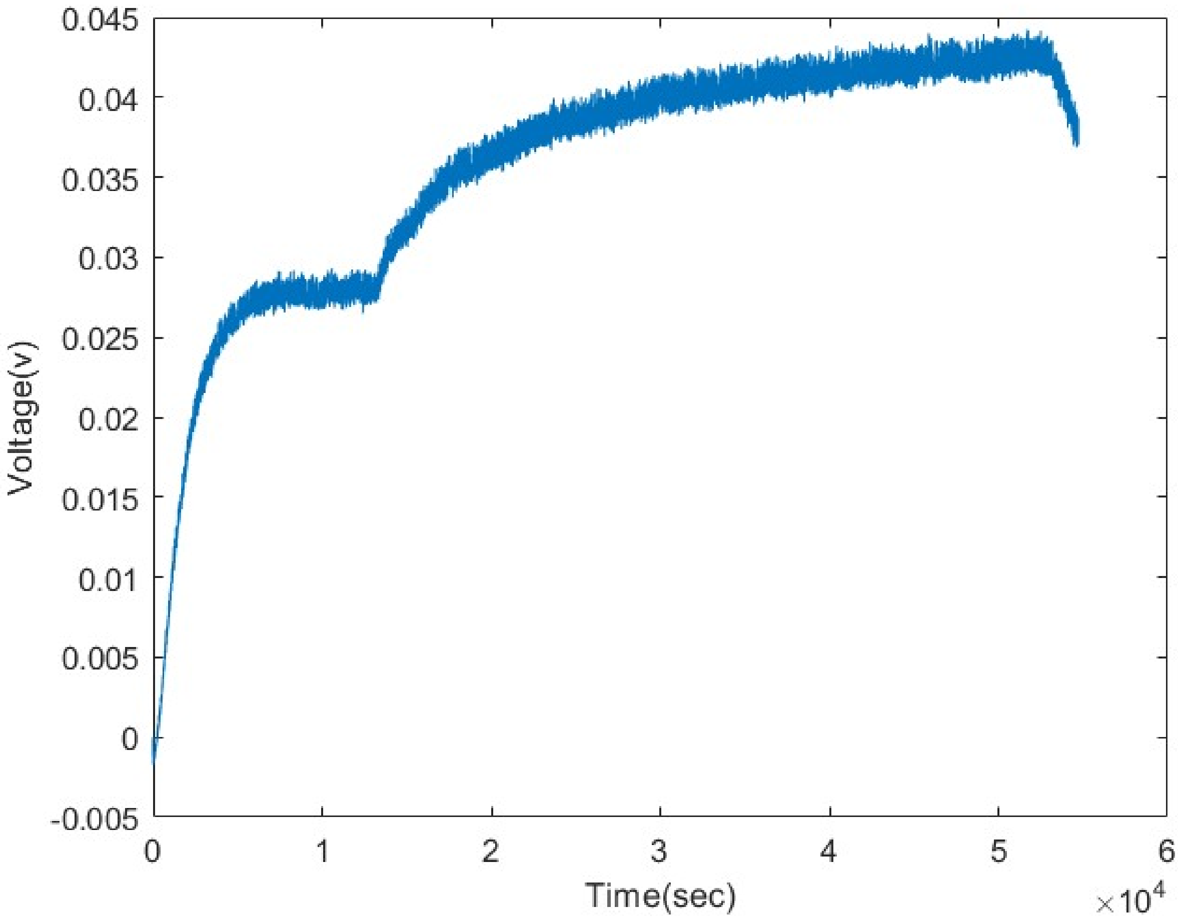

Figure 3.6: State Estimate

Covariance Matrix, $P = \begin{bmatrix} 0.0048 & 0 \\ 0 & 0.048 \end{bmatrix}$

Process Noise Covariance, $Q = \begin{bmatrix} 2 & 0 \\ 0 & 2e-5 \end{bmatrix}$

Observation Matrix, $H = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

Sensor Noise Covariance, $R = \begin{bmatrix} 2e-5 \end{bmatrix}$

3.6 SUMMARY

From the chapter the Kalman filter can be summarized as a recursive prediction algorithm, which is used for the prediction of the future state. Here, in this chapter, Kalman filter is explained with its equations and derivation. The compensation of the sensor data is achieved using the Kalman filter, for that a program flow is also mentioned. The simulations were done on the MATLAB software, in R2022a version. The simulation results are discussed and analysed in this chapter.

Chapter 4

NEURAL NETWORK

4.1 OVERVIEW

This chapter discuss about the neural networks. The neural network is a processing device, inspired from the design and the functioning of the animal brain and its related parts. The tasks in the neural network can be done without a deep knowledge in the area. The idea of neural network is more powerful and flexible.

4.2 BIOLOGICAL NEURON

Figure 4.1 shows a basic biological neuron, these neurons are the fundamental part of the nervous system which transmits the information to the body parts. Nerve cells are the building blocks. As shown in the figure below, the main parts of the neurons are;

1. Dendrites- They are the tree like branches responsible for receiving or collecting the information from the other neuron. These branches will transmit the information to the cell body or soma.

2. Cell Body/Soma- The cell body of the neuron is responsible for the processing of the information that collected or received by the dendrites.

3. Axon- The cable like part through which the information being sent.

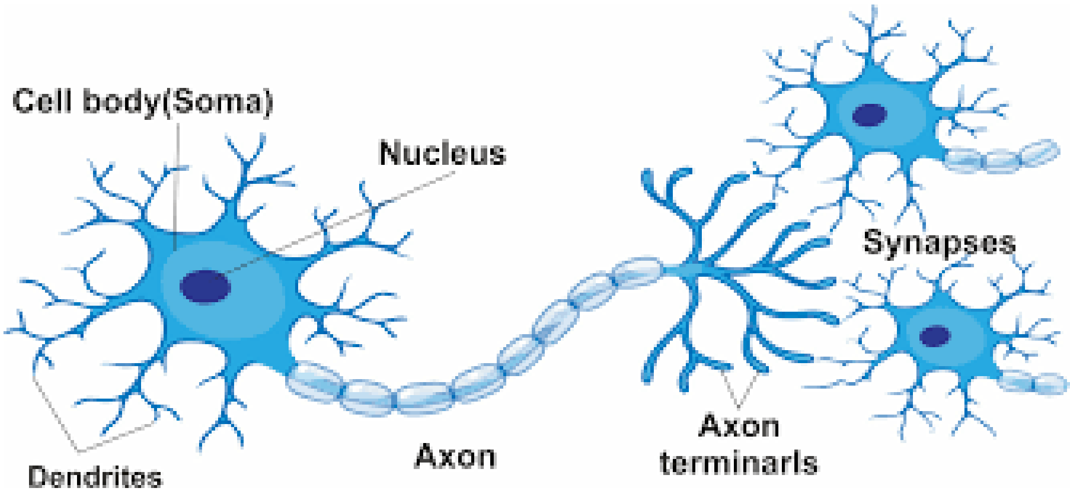

Figure 4.1: Structure of a Biological Neuron

4. Synapses- The synapses is the connection between the axon of one neuron and the dendrites of the other neuron.

The main parts of a biological neuron is explained above. Each part has a specific importance in the transmitting of an information. The biological neurons are the specialized cells that transfer the information throughout the nervous system. The working of a biological neuron is, when an electrical signal (or action potential) reaches the end of an axon, it triggers the release of neurotransmitters, which are chemical messengers that transmit the signal across the synapse to the dendrites of another neuron. The process of the transmitting data by the biological neuron is complex. This biological neuron are the basic idea behind the artificial neural networks used in the machine learning.

4.2.1 Terminology Relationship

- Dendrites - Input
- Cell Body/ Soma - Node
- Synapse - Weight
- Axon - Output

4.3 HISTORY OF ARTIFICIAL NEURAL NETWORK

The evolution of the neural network started in late 1800s.

- In 1890, William James published the first work about brain activity patterns.
- In 1943 McCulloth and Pitts created the model of the neuron that is being used till date.
- In 1949 Donald Hebb published 'The Organization of Behavior', which has the law of synaptic neuron learning, later it is called as the Hebbian learning.
- Frank Rosenblatt designed the perceptron in 1958 at Cornell University.
- Werbos developed the backpropagation algorithm in 1974; Rumelhart, Hinton, and Williams published Learning Internal Representation by Error Propagation in 1986.
- In 1988 counter propagation network was found out by Grossberg.
- In 1990, Nco cognition network was created. This network is essential in the character recognition.

4.4 ARTIFICIAL NEURAL NETWORK

Artificial Neural Network (ANN) also called as neural network or neural net is a collection of nodes or units. Like the biological synapses the ANN will also transmits the information signal from neuron to neuron. Similar to a human brain which has neurons interconnected, the ANN also have interconnection to various layers, these are called as nodes.
The Figure 4.2 shows the structure of a simple neural network. The structure has input, hidden layer and output. These neural structures shown in figure are called as multilayer feedforward network. The input layer receives the input, the output layer generates the output. All the other layers in-between the input and the output layer is called as the hidden layer, which are internal. As the number of the hidden layer increase the complexity and the efficiency in the output generated also increases.
The ANN takes the input and computes the output as the product of the input and weight added with the bias value.

$$y_i = \sum_{i=1}^{n} X_i * W_i + b \tag{4.1}$$

where X_i is the input, W_i is the weight, b is the bias and i represents the ith processing elements. The activation function is applied to the above equation (4.1) to calculate the output.

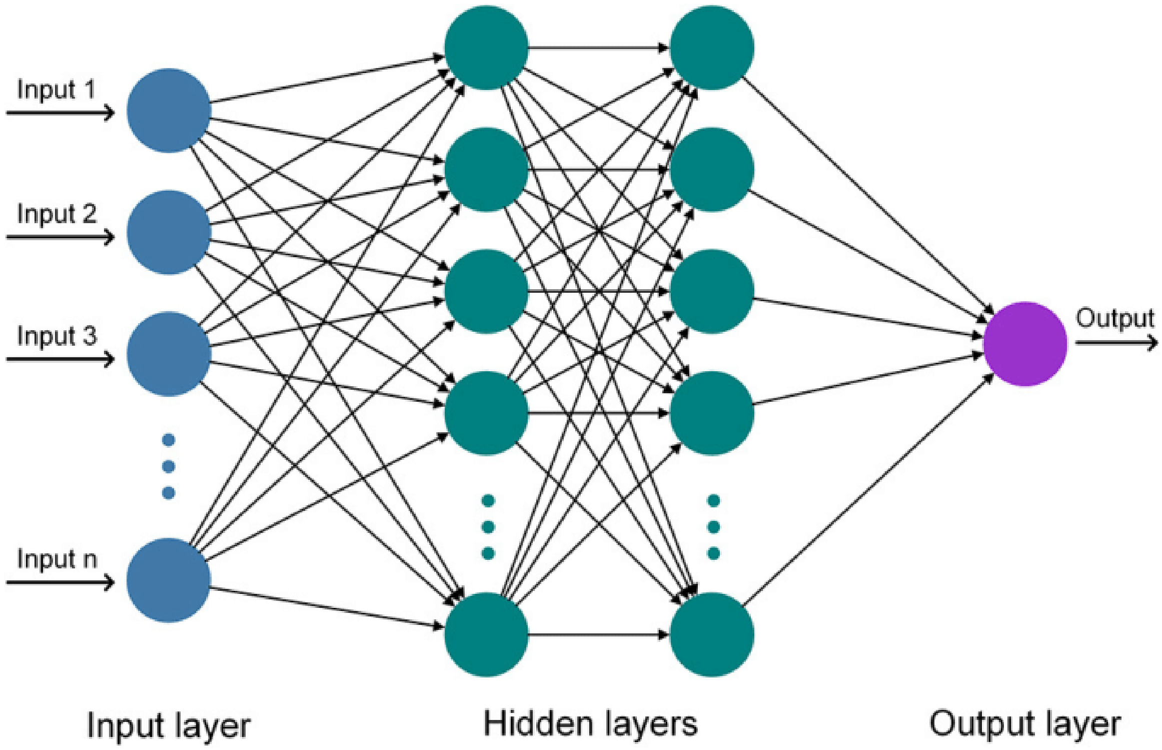

Figure 4.2: Structure of a Simple Artificial Neural Network

4.4.1 Activation Functions

Activation functions are some nonlinear functions used to found the output. The common activation functions are;

1. Binary Step Function

$$f(x) = \begin{cases} 1 & \text{if } x \geqslant \theta \\ 0 & \text{if } x < \theta \end{cases} \tag{4.2}$$

2. Bipolar Step Function

$$f(x) = \begin{cases} 1 & \text{if } x \geqslant \theta \\ -1 & \text{if } x < \theta \end{cases} \tag{4.3}$$

   where θ is the threshold value , rage fror binary function is from 0 to 1 and for bipolar -1 to 1.

3. Binary Sigmoid Function

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \tag{4.4}$$

4. Bipolar Sigmoid Function

$$f(x) = \frac{2}{1+e^{-\lambda x}} - 1 = \frac{1-e^{-\lambda x}}{1+e^{-\lambda x}} \tag{4.5}$$

where λ is the steepness parameter, in binary sigmoid function it ranges from 0 to 1 and for bipolar its is from -1 to 1.

5. Ramp function

$$f(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x < 0 \end{cases} \tag{4.6}$$

6. ReLU- Rectified Linear Unit

$$f(x) = \max(0, x) \tag{4.7}$$

7. Identity Function

$$f(x) = x, \text{for all} x \tag{4.8}$$

8. tanh- Hyperbolic Tangent, the activation function similar to the sigmoid activation function. The values ranges from -1 to 1.

4.4.2 Weights

The weights has the information about the input given , which should be used to solve the problem. The weight matrix is called as the connection matrix. Weights can be initialized by the following techniques;

1. Zero Initialization- All the weights are assigned to be zero.The main advantage of this method is, the computation is made easy and easy to implement. This method is an ineffective way of weight initialization as the neuron learn same feature during the iterations, this will leads to the symmetry breaking problem, where the network fails.

2. Random Initialization- The initialization technique in which the weights are assumed to be any of the random numbers. If the weights are assumed from the normal distribution

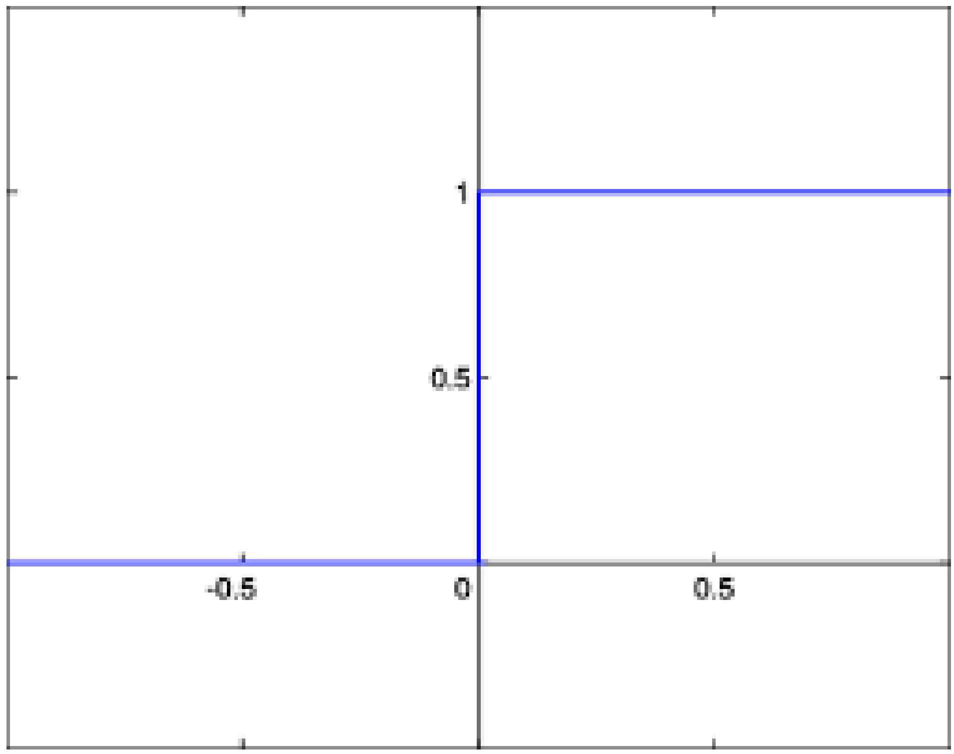

Figure 4.3: Binary Step Activation Function

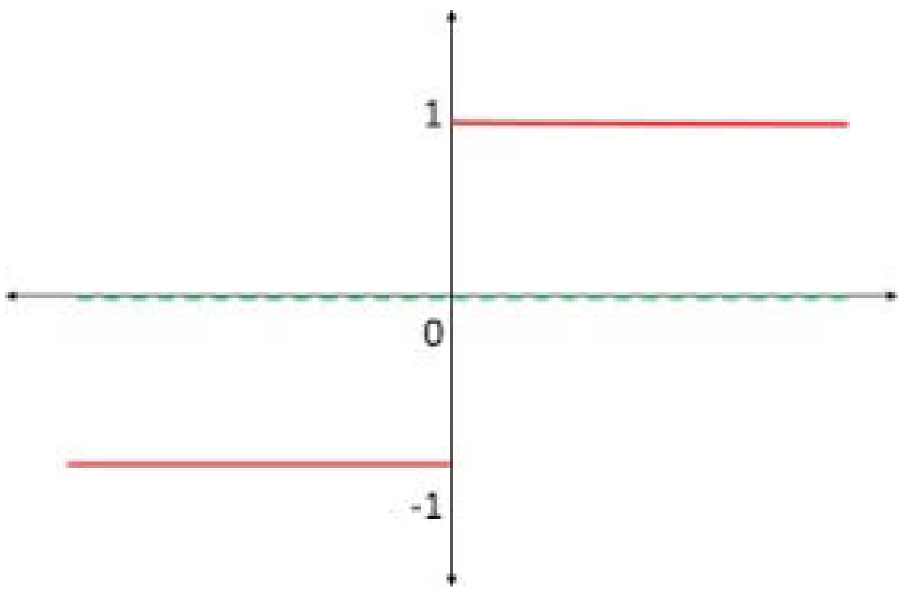

Figure 4.4: Bipolar Step Activation Function

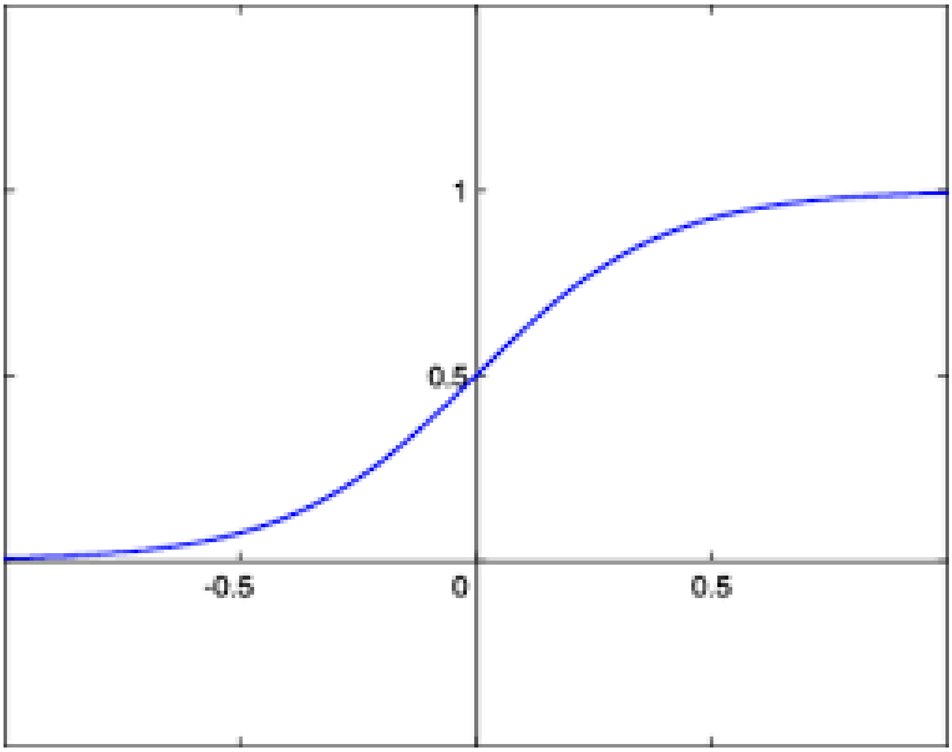

Figure 4.5: Binary Sigmoid Activation Function

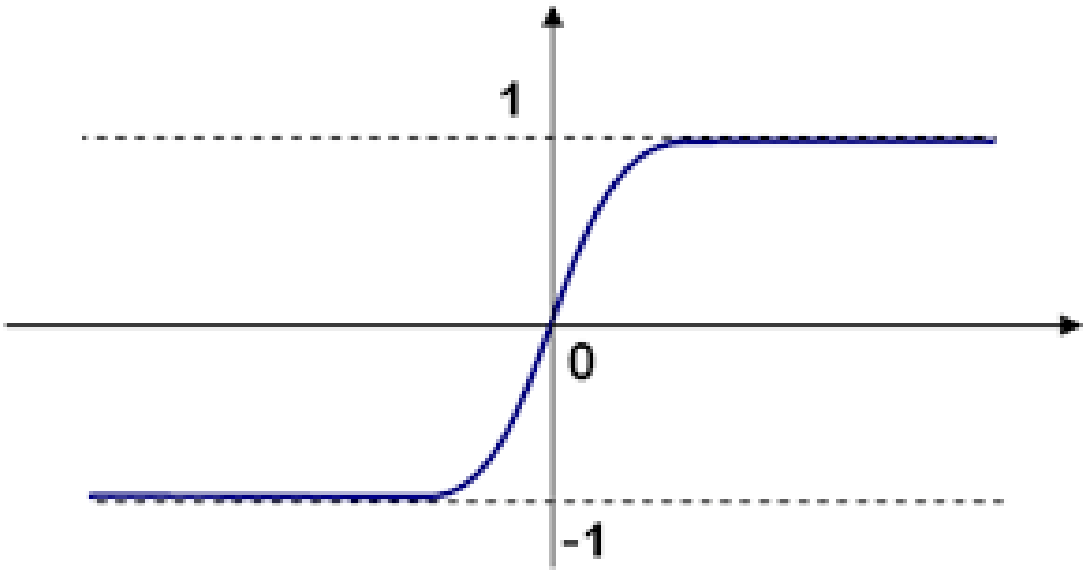

Figure 4.6: Bipolar Sigmoid Activation Function

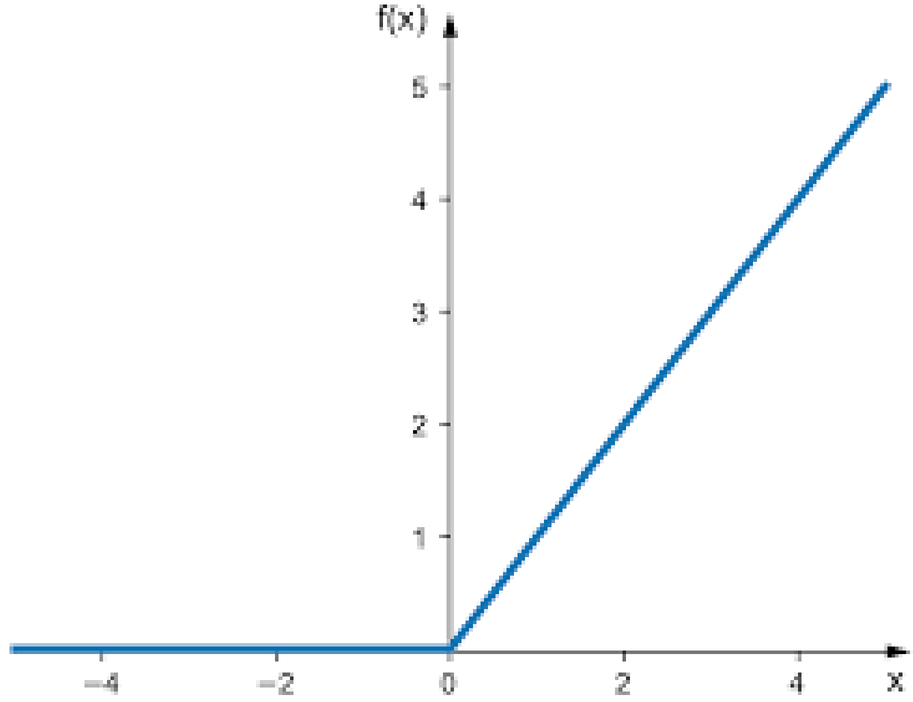

Figure 4.7: ReLU Activation Function

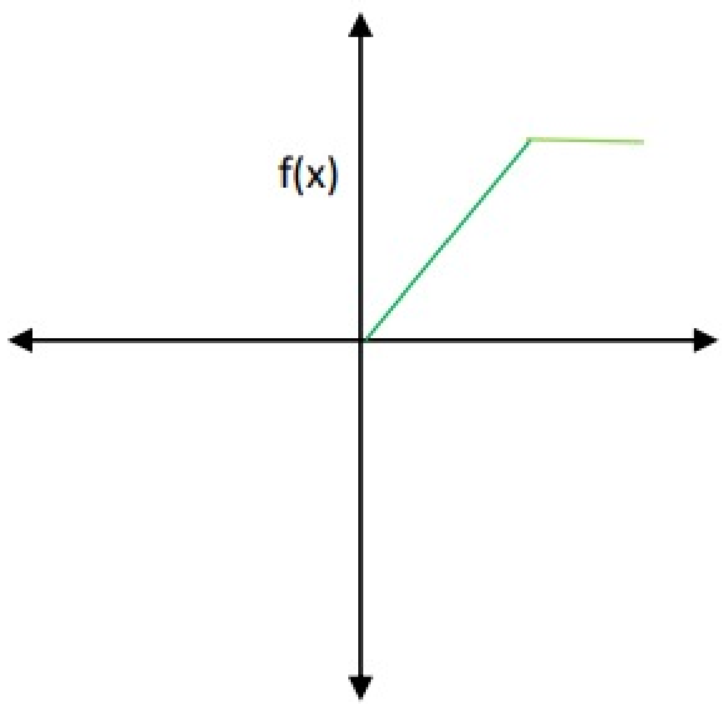

Figure 4.8: Ramp Activation Function

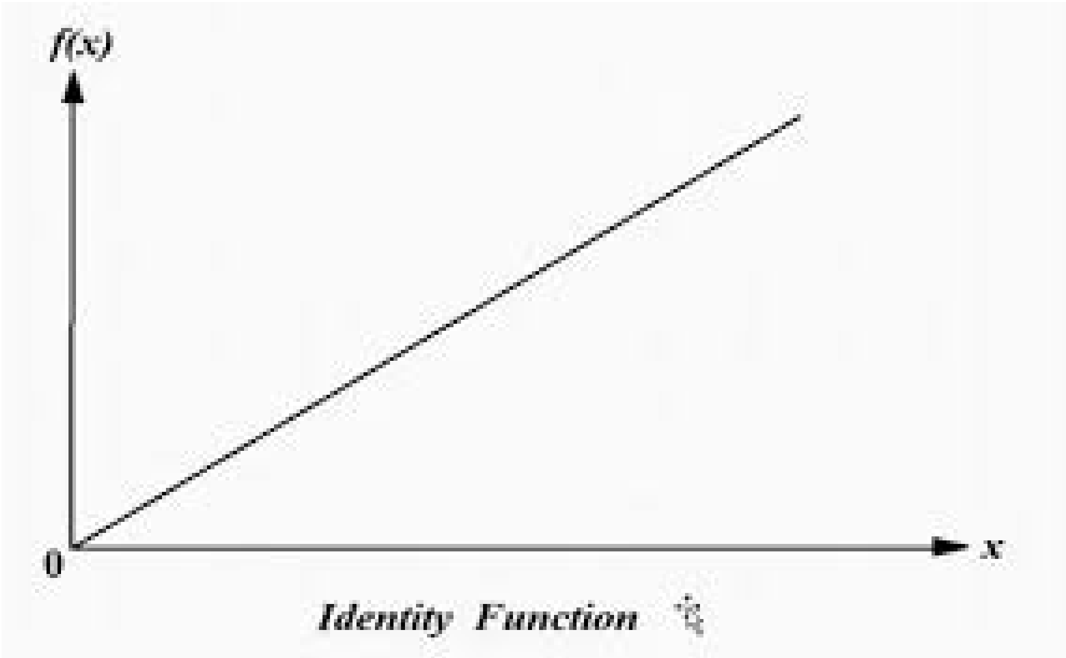

Figure 4.9: Identity Activation Function

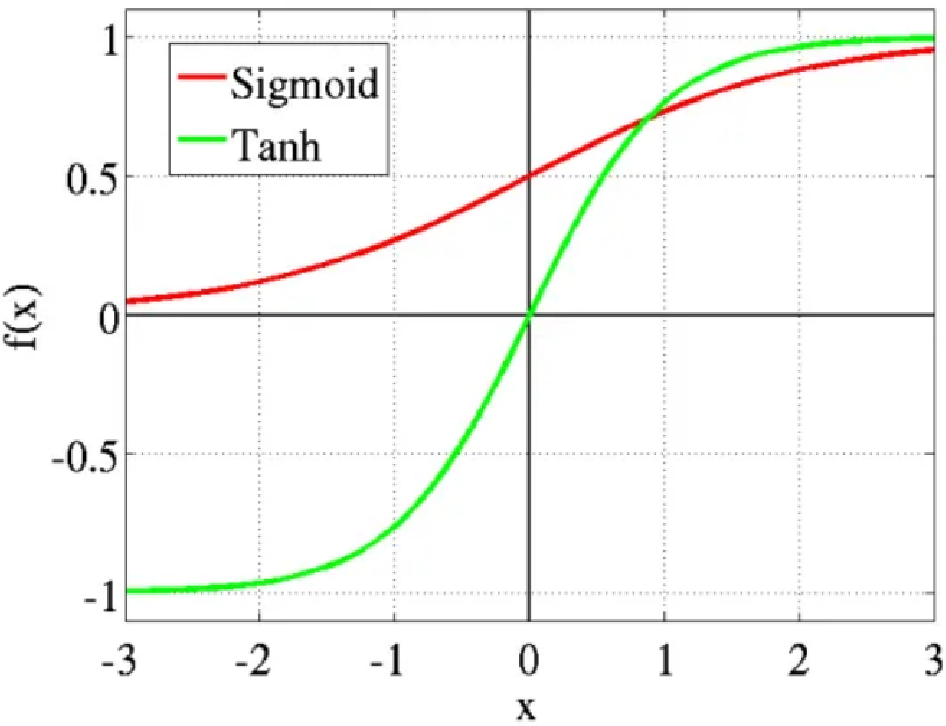

Figure 4.10: tanh Activation Function

then it is random normal or if the weights are initialized from the uniform distribution, it is random uniform. The random values of the weights are taken from the probability distribution. This method helps to breaks the symmetry problem. The main disadvantage of using the zero initialization is the breaking of the symmetry points, this can be eliminated by using the random initialization. The random initialization uses different values of weights this will make the network to learn the complex patterns too.

3. Xavier/Glorot Initialization- Mostly used with the sigmoid activation function. It can also Xavier Uniform Initialization or normalized Xavier/Glorot initialization if values from the uniform or the normal distribution used respectively. This method initializes the weight by taking into account the number of input and output neurons. It was introduced by Xavier Glorot and Yoshua Bengio in 2010. The weights will be initialized in such a way that the variance of the input will be roughly same with the variance of the output. The assumption related with the Xavier initialization is, the input to each neuron is distributed normally with zero mean and unit variance and the output are also assumed to be distributed normally with zero mean and unit variance. By the assumptions the weights are assigned using Gaussian distribution with zero mean and the standard deviation. The use of Xavier initialization will improve the performance of the neural network, particularly used in the deep learning applications where the numbers of layers are large.

4. He Initialization- This initialization technique is used in the layers with ReLU activation function. This technique can be He normal initialization and He uniform initialization based on the values selected. If the selected value is from normal distribution then it is called as the He normal initialization, if the values are from the uniform distribution it is called as the He uniform initialization. This method actually an extension of the Xavier/Glorot initialization. It was introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in 2015. The use of the ReLU activation function will leads to the dead neurons, which are the neurons that always outputs zero, and no output is given. He initialization addresses this problem, by adjusting the variance in each layer weights based on the number of inputs to the layer. As in the Xavier/Glorot initialization, the He initialization also initializes the weights by using a Gaussian distribution with mean zero

ans standard deviation.

4.4.3 Bias

The bias is a scalar that is added with the weighted sum of the inputs. It is a factor that tells us about how the neuron can fire in the output to a given input. The bias is added with the product of the input and the weight, which has an impact on calculating the net input. The positive bias increase the net input whereas the negative bias decreases it. The bias is an important term related with the neural network. Without a bias term, the network will be forced to pass through the origin, this will limits the performance, accuracy and the expressiveness of the model. The bias is normally initialized as zero, then later learned during the training process. The weights and the bias is adjusted during the training, this adjustment is to reduce the cost function, which is used to findout the difference between the predicted and the target value. But in certain cases the bias will leads to the overfitting, this can be reduced by using the regularization techniques like L1 regularization, L2 regularization, these regularization are applied to the bias term, the prevention of overfitting is by the addition of a penalty term with the cost function, this penalty term is based on the magnitude of the bias values.

4.5 BACK-PROPAGATION

These network uses the back-propagation learning algorithm. This is an weight update algorithm in which the gradient descent is used as in a simple perceptron. This will propagates the error from the output nodes to the input nodes. The neurons present in the output layer and the hidden have bias, sometimes these bias will act as weights. Usually these bias values are 1. The back-propagation works based on a chain rule in which the gradient loss function is computed with respect to the weight, ie., computing from the last layer and iterating the values to the backward layer. It can be done in three ways;

1. The feed-forward input pattern

2. Calculation and back-propagation of the errors

3. Updation of weights

Back-propagation is the method of adding weights to the system. A multilayer structure is shown in the Figure4.11, which has two hidden layers, input layer and and an output layer. The number hidden layer can be increased or decreased according to our requirement, The increased number of layers suggests that the system will be more accurate in the training process. Each connection has one weight and layer has a bias in the a simple structure. In case of back-propagation algorithm, as the name suggests the weight setting will be done from backwards ie., from the output section. To start the algorithm, the input values are fed through the preconnected path. For the initial step the weights are assumed randomly, mostly 1 is taken. The calculation of the output from the assumed weight, the output from neuron of each layer is found ie., from input layer to the hidden layers and later to the output layer. The error will be calculated and now the actual back-propagation process starts, the iteration from the output layer to the hidden layers then finally to the input layer. This iterative process will continue until the desired result is obtained.

4.5.1 Back-Propagation Algorithm

1. Inputs arrive through the preconnected path.
2. Weights are randomly chosen.
3. Determine each neuron's output from the input layer to the hidden layer and finally to the output layer.
4. Errors of the output are calculated.
5. The iteration from the output layer to hidden layer and to the input layer starts to reduce this error.
6. This process will be repeated until the final desired output is obtained.

The back-propagation network has several learning factors like learning rate, initial weights, momentum factors.

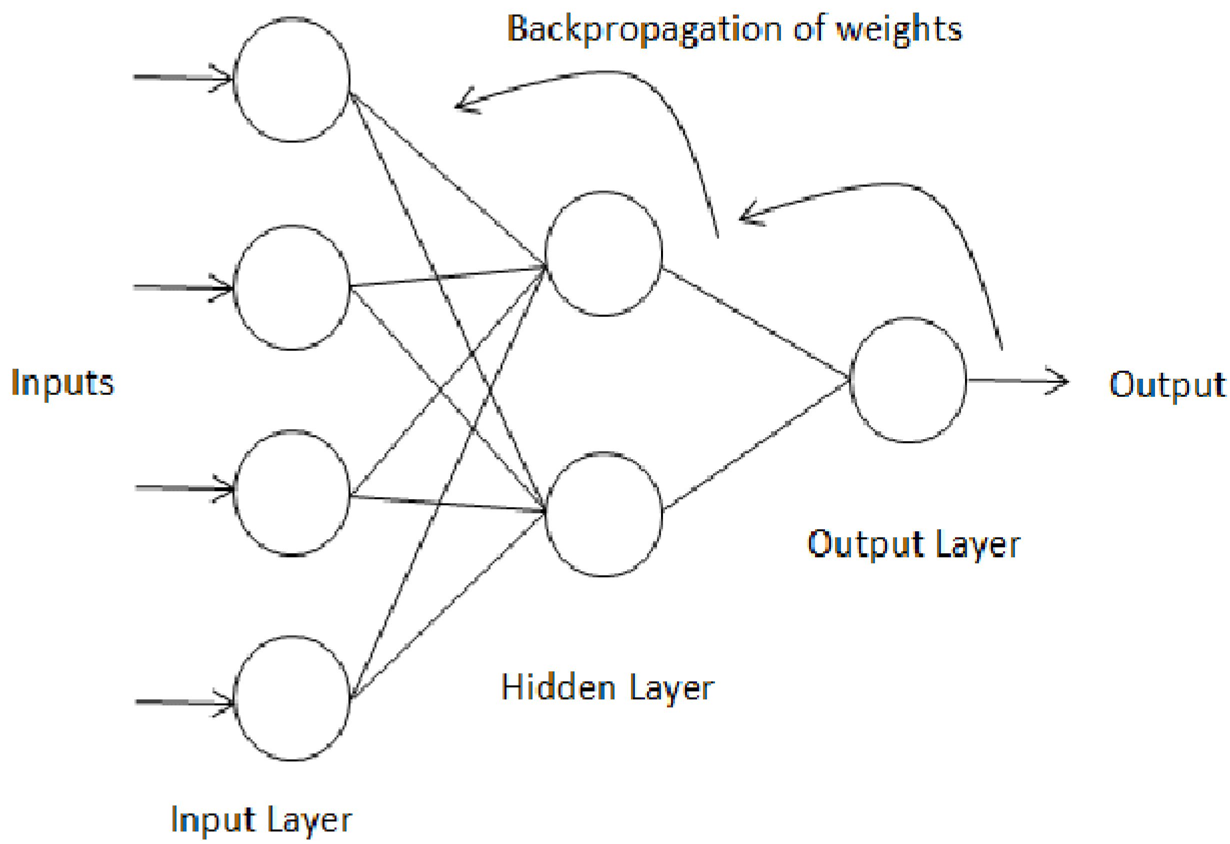

Figure 4.11: Back-Propagation

Initial Weights

The final solution is affected by the initial weight, most cases these weights are chosen randomly. The choice of the initial weight determines the the fast convergence. Initial weights assumption is the important step in the neural network analysis as these determine how quickly will the network converges to the optimal weights and biases. The values of weights are assigned in the beginning of the training process. The weights can be initialized by different techniques, like zero initialization, Xavier/Glorot initialization, He initialization and random initialization, these are explained in the above section. In practical cases, for finding the better method each one should be tried like a trial and error method.

Learning rate α

Learning rate is the hyperparameter that controls the step size at each iteration in the algorithm. It determines how much the weights and the bias should be adjusted. The learning rate will affect the convergence. If the chosen value of α is large then it will converge fastly results mostly in the overshooting. The range of α, the learning rate is ranging from 10^{-3} to 10. A common method of adjusting or setting the learning rate is to use a fixed value, then during the training progress the updation to this value will happen. If the cost function is not deceasing sufficiently, then the learning rate can be decreased to make small updations in the weights and biases. Similarly if the cost function is decreasing, then the learning rate must be made large to get the updates of the weights and biases. The optimizer like Adam, Adagrad, RMSprop can be used for adjusting the learning rates in the neural networks.

Momentum Rate

The momentum rate is also a hyperparameter, that is used for controlling of the weights and biases during the training. It determines the contribution of the previous update on the current update. As mentioned earlier, if the value of learning rate is high then the faster convergence will be the result. To reduce this effect ie., reducing the oscillation is adding up a momentum factor to the gradient descent method. this will be helpful in the cases with large data; especially in our case. The momentum rate will make the cost function to minimum, this will make the update process smooth and avoids the oscillations in the local minima. If the momentum rate is 0 then it means that the previous update has no effect on the current update, if the momentum rate is 1 then the current update is determined by the previous update. Like the case of learning rate, the momentum rate is also set as a fixed value then later during the training process its updation occurs. If the algorithm is overshooting, then the momentum rate should be decreased to reduce the influence on the updates. If the algorithm is stuck in the local minima, then the momentum should be increased to escape from the local minima, to get a better solution. Adaptive momentum techniques an be used for the updating of the momentum rate, like Adamax, Nesterov accelerated gradient these adjust the momentum based on the gradient of the cost function.

4.6 WORK DONE USING NEURAL NETWORK

Training, in simple words is the process of teaching a neural network to perform a task. It is a method by which an appropriate weight and the bias are found out. The training dataset is used to train the network. Various values of the weights and bias will be computed and generated, which will be more accurate with the condition will be used. The training will produce accurate or sightly accurate weights by the process. Commonly used training algorithms are back-propagation and particle swarm optimization algorithm. The particle swarm optimization algorithm is a technique inspired from the nature. Its idea is from the group of fish or flock of birds where every one interact each for an optimal solution. The position of particle is the weight. The back-propagation is the technique in which the weights are found from the backwards ie., from the output layer to hidden layer and finally to the input layer.

The neural network is used here for the fitting of the sensor data. The observed results and the analysis are discussed below. The prediction curve and the performance of the structure are obtained as shown.

4.7 TRAINING RESULTS

The sensor data is used here for the training. Two sensor data are fitted for achieving the temperature compensation. The fit plot provides the curve, which shows that the values are fitted to a good range. The training were done using the MATLAB software. A version release of R2022a is used here for the training. The used training algorithm is *trainlm* (Levenberg-Marquardt back-propagation).

4.7.1 Levenberg-Marquardt back-propagation

The Levenberg-Marquardt algorithm is a technique by which the least square can be minimized, derived from the Newtons method. This method is computing technique by which the second order training process can be computed without finding the Hessian matrix. It is a modification of the standard back-propagation algorithm with improved convergence rate and stability rate. In standard back-propagation the process can be slowdown and the results can be stuck in a local minima, this disadvantage can be minimized using the Levenberg-Marquardt method. By this method, a update parameter rule is modified. Thus, this method can be used in speech recognition, image recognition and control system applications etc.The training function is normally used with the feedforward network. The Hessian matrix is computed as,

$$\mathbf{H} = \mathbf{J}^T\mathbf{J} \tag{4.9}$$

and its gradient is calculated as,

$$\mathbf{g} = \mathbf{J}^T\mathbf{e} \tag{4.10}$$

where $\mathbf{J}$ is the Jacobian matrix, that has the first derivative of the network error, weights and the bias and $\mathbf{e}$ is the network error. This algorithm appears to be the fastest method for training moderate-sized feedforward neural networks (up to several hundred weights).

4.7.2 Fitting of the Sensor Data 1

The training is the process of setting the weights. In the fitting of the data, the Levenberg Marquardt back-propagation algorithm is used. The learning parameter is set by the calculation of the error, damping parameter during the training process itself. For the neural network training, there should be an input and a target, based on the target the input values are made to the values by fitting, the fitting plot is depicted below. The results of the network training are given, Figure 4.12 gives the input curve which is a temperature data. The temperature vs time curve is plotted, the temperature values ranges from -3.6 to 2.3. The curve is step signal, the sensor is made to test in an environment, its temperature are plotted. The variations in the data can be compensated with the use of the neural network. Figure 4.13 is the target curve, which is a voltage curve.

The training results are also added, the training was done for a layer size of 5,5,3,2. The ReLU activation function is used here. As mentioned above the training function used here is *trainlm*, by this algorithm the damping parameter or the mu value (0.01) is made to balance the error for finding the weights. In the MATLAB programming section the mu values can be set by the user. By default the epoch size is 1000, and this program is made to train in an epoch number of 1000. From the target curve, the value range is above 1.5.

The prediction value and the target values have a great importance in the neural network. The predicted values depends upon the network formed, training epochs, layer size etc. In the introductory section of the neural network it is given that, the increased number of layer size will increase the system accuracy and performance. The error is the difference between the predicted and the target value, shown in Figure 4.14. By the use of the network the error has been minimized to range of -0.45 to 0.36.

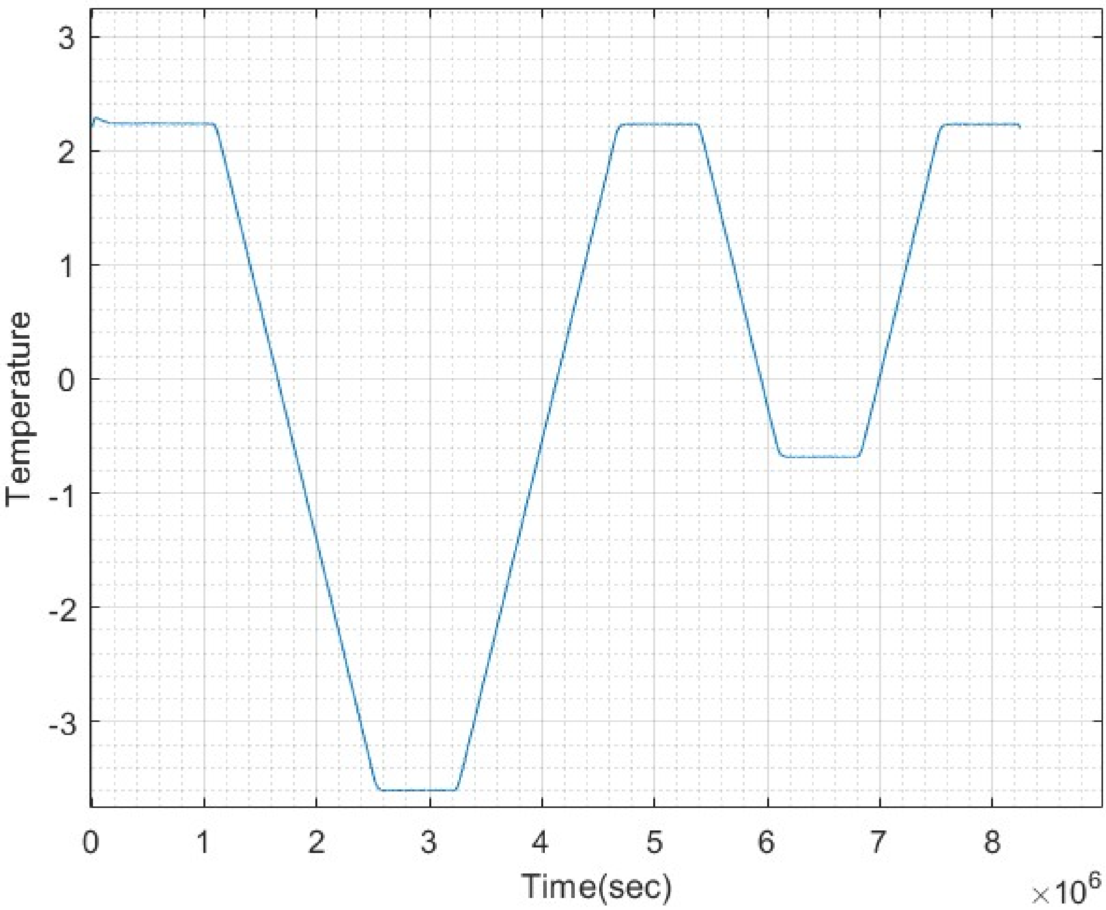

Figure 4.12: Input to the Neural Network

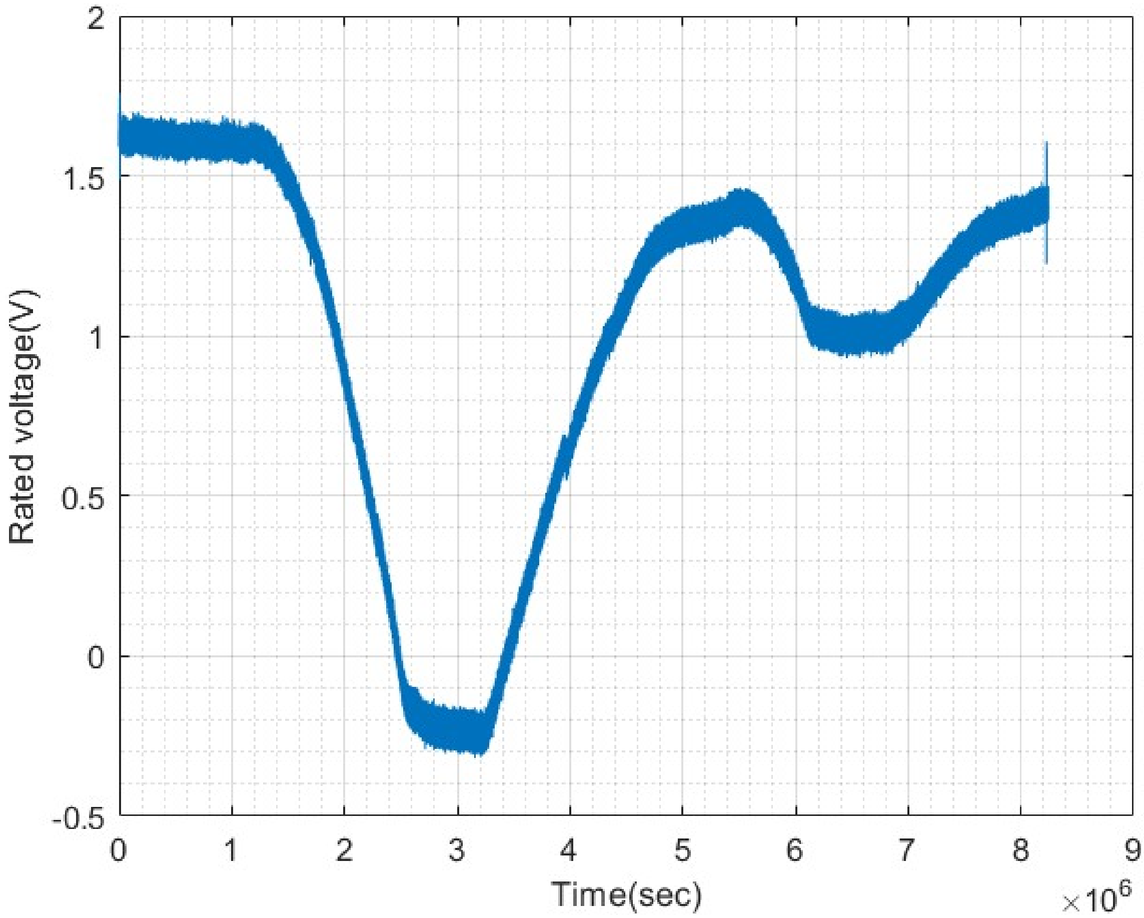

Figure 4.13: Target Plot

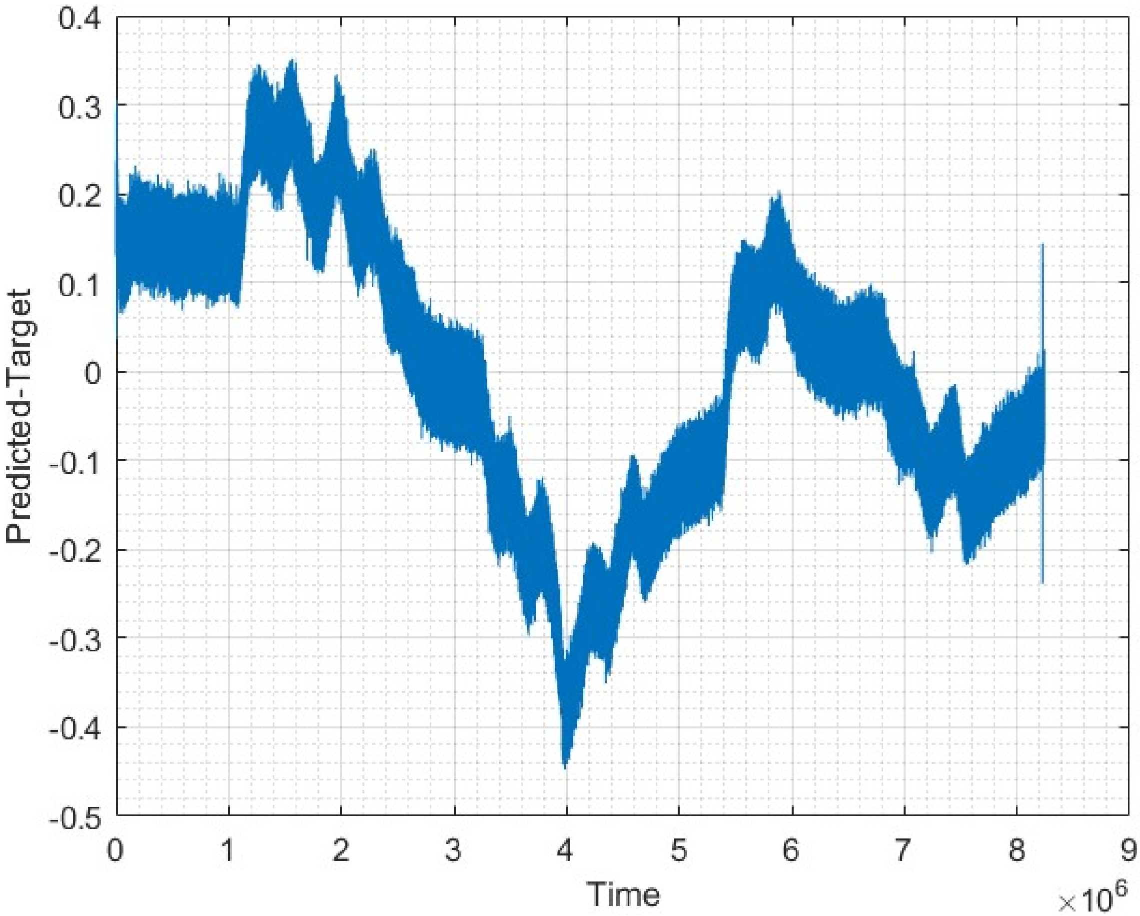

Figure 4.14: Error Curve

The performance curve, Figure 4.15 has the best performance at an iteration of 263 with a value of 0.024541. The training process will stops when the maximum condition or the best performance is obtained. The observation of the training progress is important in the training process, this will allow the user to optimize the hyperparameters, training time, problems regarding the training accuracy. Thus, the performance plot is important.

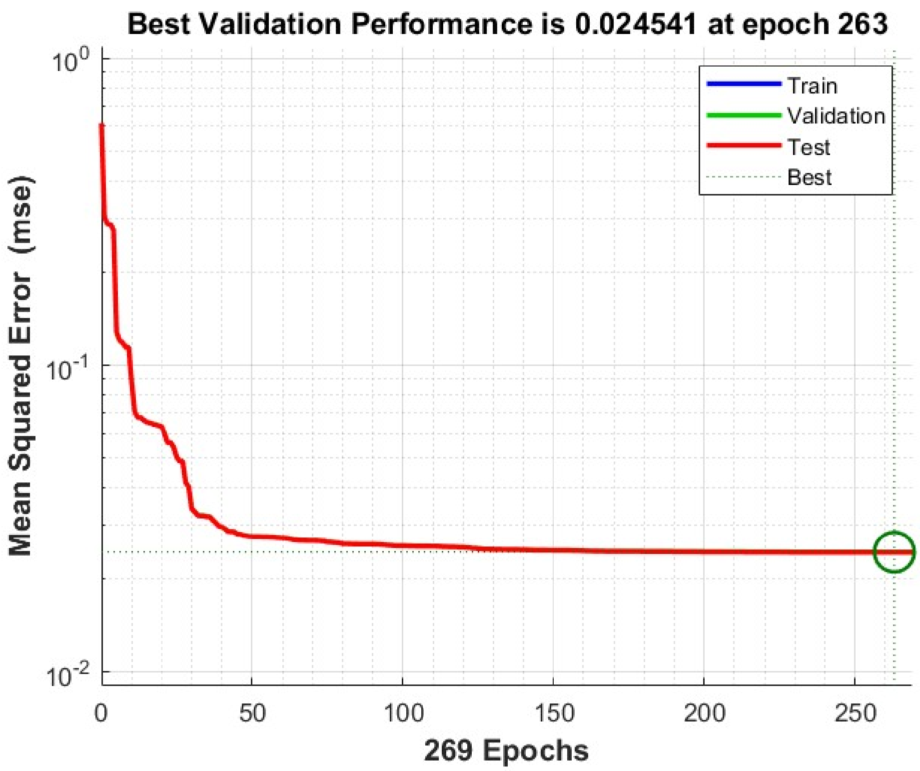

Figure 4.15: Performance Curve

The above plots give the performance, error. The predicted network is used to plot the fit. Figure 4.16 illustrating the fitting ie., the fit plot shows the best fitting of the curve. The input curve is fitted with the target values. The centre blue line shows the fit curve, the fit is obtained at a value of -0.2. In the same plot the error is also plotted with the input values. The best fit result is given in the figure. From the given data 70% data is used for the training, 15% for validation and 15% for testing. This ratio is the standard way dividing based on the number of data and the user requirements the test data and the validation data can be set. In another case case, a new testing data can be fed to the neural network by the user.

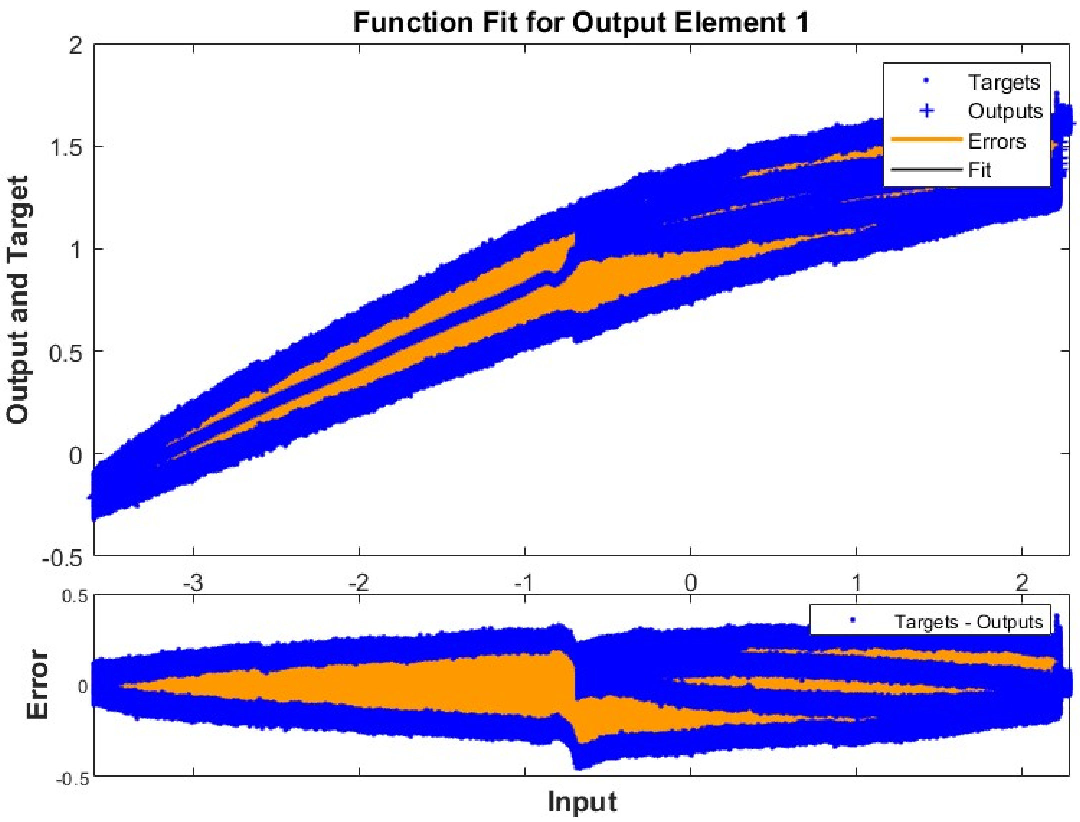

Figure 4.16: Function Fit Plot

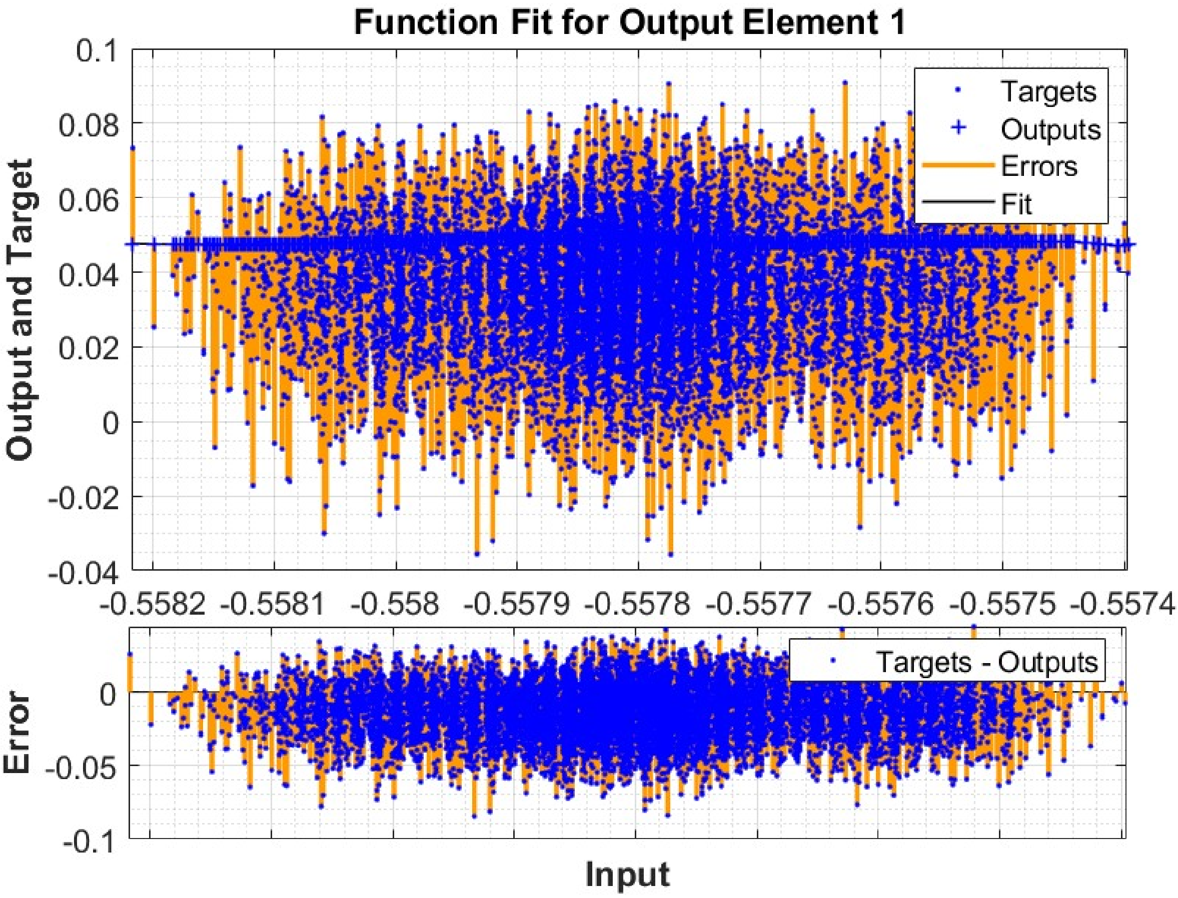

Figure 4.17: Function Fit Plot

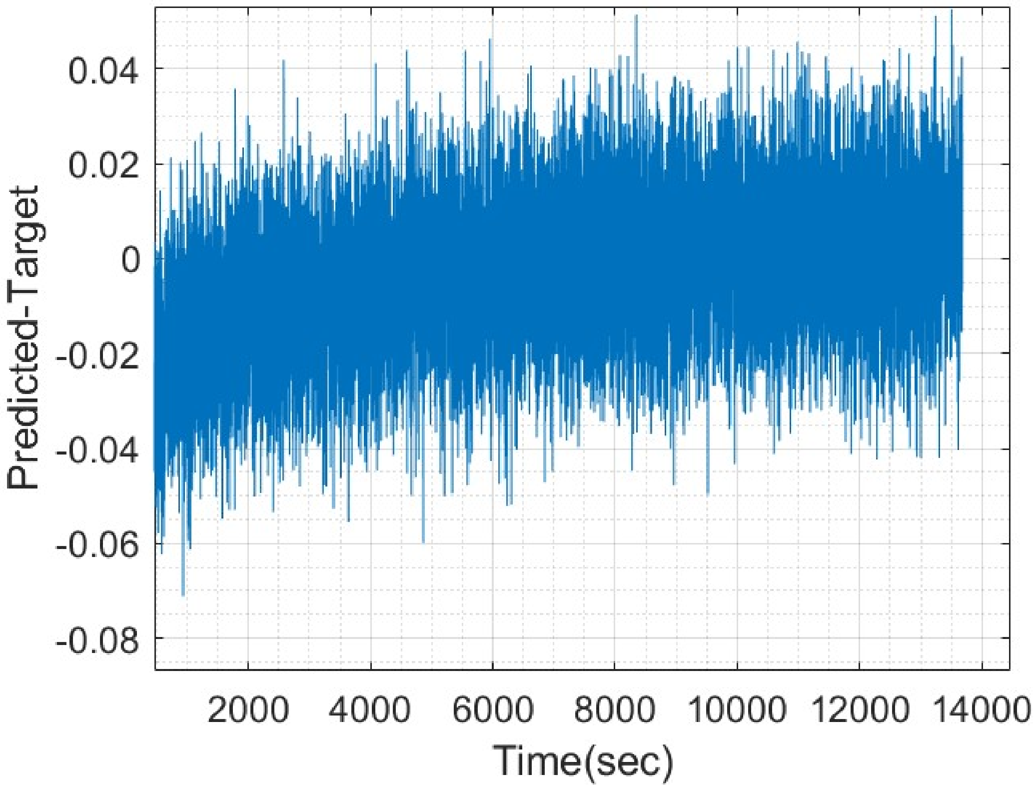

Figure 4.18: Error plot

4.7.3 Fitting of Sensor Data 2

The same sensor data that is used for the compensation with the Kalman filter is fitted using the neural network. The results are discussed below. The input to the network is plotted in Figure 3.3 and the target curve is plotted in Figure 3.6. The data are made to train with the *trainlm* algorithm. The best parameters are plotted. The training process is done with the damping parameter of 0.01. Weights are updated with the back-propagation method. For the training process, the input and the target are needed, based on the target the input is adjusted, as a result, the fit plot is created. The fit gives the value of 0.045, this is shown in the Figure 4.17. The error vs input plot is also obtained in the fit plot.

The error is the difference between the predicted value and the target value. The error plot obtained here is a varying curve. The predicted value can be found out from the neural network. The sensor data is made to train for a layer size of 5,5,3,2. The training is done for an epoch size of 1000, with the damping value of 0.01. The error plot, Figure 4.18 gives the error values in a

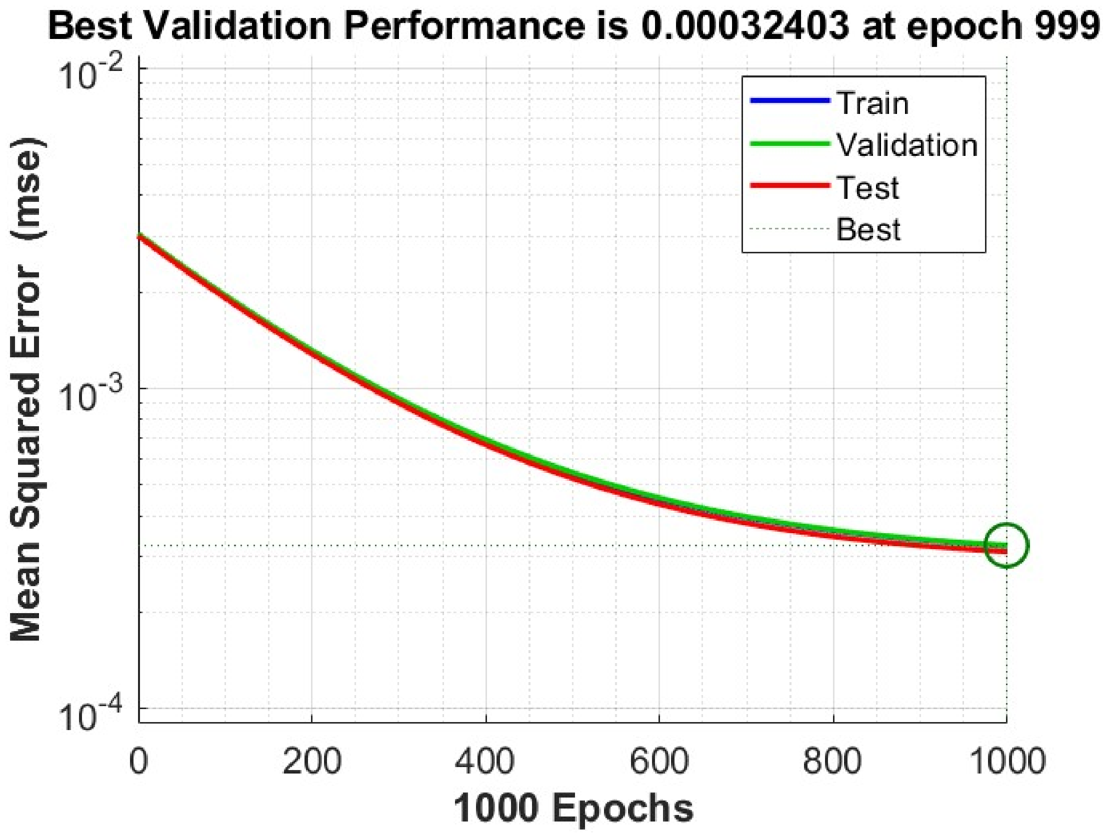

Figure 4.19: Performance Curve

range of -0.0004 to -0.0001 range, the error is also minimized. The Figure 4.19 shows the performance curve. The performance curve is the training progress of the process. The increased number of layer size suggests that the network will be more complicated and the analysis will have more accurate and precise value. The best performance is obtained at epoch 999 and the performance of 0.00032403. The observation of the training progress and the loss is important as this provides the best training.

4.8 SUMMARY

In this chapter the neural network is discussed. From the above sections it is clear that, the artificial neural network is an idea from the biological neurons. The training algorithm used is explained here. The training is done by using the Levenberg-Marquardt back-propagation algorithm, by which training results are obtained. These results are analysed and explained in this chapter. By using the neural network technique two sensor data are fitted, thus the compensation is achieved for the two data. The training is done using MATLAB.

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

MEMS are the lowcost, very effective sensors used in the satellite, navigation, military and its related applications. The optimal filtering will make the sensors to be used in the complex environment. The main advantage of using MEMS sensors over the conventional IMU is that the size compatibility; these small sensors will make the application at the best with the all needed criteria. MEMS sensor, generally gyroscope is a vibrating gyroscope with the MEMS technology.

As these are the real time application system the filtering noise and compensation is very important. Here two strategy is used for the compensation; one is the classic Kalman filter and the other is the neural network. The training were done in the MATLAB software. Kalman filter is a recursive prediction algorithm used for filtering and the compensation. This filter will predict the future state based on the past data. The initialization followed by the measurement and updation are the steps involved in the Kalman filter. These steps are explained in the chapter 3. The workdone using the filter is the compensation of the data.

The neural network is also used for the compensation. The neural network used here is a feedforward network with Levenberg-Marquardt back-propagation algorithm. This training function have a good convergence rate. The training progress is is also observed. The results shows that, the convergence rate is in a good range. Two sensor data compensated and fitted using the neural network. Its results are analyzed and discussed in the chapter4. In both the technique, the results are obtained using the MATLAB software, here release of R2022a is used.

5.2 FUTURE SCOPE

As we all know the technology is changing and going beyond the boundaries, every field has high importance and value in the near future, so the application of the MEMS, its related field, AI will increase, hence the future scope to this work is high. Here I have used the simple neural network for the analysis, as the further work or studies more complex structures can be used. In case of the Kalman filter, the covariance setting or finding its value is an importance step also it can be a disadvantage too. Thus, other filters which is superior to the Kalman filter can also be implemented on the same system, like particle filter. The achievements in the field of machine learning will improve the studies and works in the neural network field.

REFERENCES

[1] A. Persson, "How do we understand the coriolis force?" *Bulletin of the American Meteorological Society*, vol. 79, no. 7, pp. 1373–1386, 1998.

[2] F. Khoshnoud and C. W. de Silva, "Recent advances in mems sensor technology-mechanical applications," *IEEE Instrumentation & Measurement Magazine*, vol. 15, no. 2, pp. 14–24, 2012.

[3] J. W. Judy, "Microelectromechanical systems (mems): fabrication, design and applications," *Smart materials and Structures*, vol. 10, no. 6, p. 1115, 2001.

[4] A. M. Shkel, "Micromachined gyroscopes: challenges, design solutions, and opportunities," in *Smart Structures and Materials 2001: Smart Electronics and MEMS*, vol. 4334. SPIE, 2001, pp. 74–85.

[5] R. Antonello and R. Oboe, "Mems gyroscopes for consumers and industrial applications," in *Microsensors*. IntechOpen, 2011.

[6] B. Preethi, L. Sujatha, and V. Selvakumar, "Design and analysis of mems gyroscope," in *Proceeding of the 2012 COMSOL conference in Bangalore*, 2012.

[7] P. R. George, M. M. Jermila, A. K. Backer, A. Anvar, and N. A. Vishnu, "Drive mode of mems rate sensors with software phase locked loop," in *2021 Fourth International Conference on Microelectronics, Signals & Systems (ICMSS)*. IEEE, 2021, pp. 1–5.

[8] L. Dong and K. Zhang, "Sensing and control of mems accelerometers using kalman filter," in *2012 24th Chinese Control and Decision Conference (CCDC)*. IEEE, 2012, pp. 3074–3079.

[9] H. Chang, L. Xue, W. Qin, G. Yuan, and W. Yuan, "An integrated mems gyroscope array with higher accuracy output," *Sensors*, vol. 8, no. 4, pp. 2886–2899, 2008.

[10] L. Xue, C. Jiang, L. Wang, J. Liu, and W. Yuan, "Noise reduction of mems gyroscope based on direct modeling for an angular rate signal," *Micromachines*, vol. 6, no. 2, pp. 266–280, 2015.

[11] R. P. Leland, "Mechanical-thermal noise in mems gyroscopes," *IEEE Sensors Journal*, vol. 5, no. 3, pp. 493–500, 2005.

[12] J. Leclerc, "Mems for aerospace navigation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, no. 10, pp. 31–36, 2007.

[13] A. Ma'arif, I. Iswanto, A. A. Nuryono, and R. I. Alfian, "Kalman filter for noise reducer on sensor readings," *Signal and Image Processing Letters*, vol. 1, no. 2, pp. 50–61, 2019.

[14] S. Cai, Y. Hu, H. Ding, and H. Chen, "A noise reduction method for mems gyroscope based on direct modeling and kalman filter," *IFAC-PapersOnLine*, vol. 51, no. 31, pp. 172–176, 2018.

[15] M. S. Grewal and A. P. Andrews, *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.

[16] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[17] A. Becker, "Kalman filter," *Dipetik Juli*, vol. 1, p. 2020, 2018.

[18] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. IEEE, 2015, pp. 74–77.

[19] H. Cao, H. Li, X. Sheng, S. Wang, B. Yang, and L. Huang, "A novel temperature compensation method for a mems gyroscope oriented on a periphery circuit," *International Journal of Advanced Robotic Systems*, vol. 10, no. 9, p. 327, 2013.

[20] S.-R. Chiu, C.-Y. Sue, C.-H. Lin, L.-T. Teng, L.-P. Liao, Y.-W. Hsu, and Y.-K. Su, "Active thermal compensation of mems based gyroscope," in *SENSORS, 2012 IEEE*. IEEE, 2012, pp. 1–4.

[21] Q. Tang, X. Wang, Q. Yang, and C. Liu, "Static temperature analysis and compensation of mems gyroscopes," *International Journal of Metrology and Quality Engineering*, vol. 4, no. 3, pp. 209–214, 2013.

[22] M. Cui, Y. Huang, W. Wang, and H. Cao, "Mems gyroscope temperature compensation based on drive mode vibration characteristic control," *Micromachines*, vol. 10, no. 4, p. 248, 2019.

[23] S.-L. Xia, J. Wang, R. Wang, and L. Zhao, "A temperature compensation algorithm based on curve fitting and spline interpolation," *chemical engineering transactions*, vol. 51, pp. 1345–1350, 2016.

[24] G. Araghi *et al.*, "Temperature compensation model of mems inertial sensors based on neural network," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2018, pp. 301–309.

[25] R. Zhu, Y. Zhang, and Q. Bao, "A novel intelligent strategy for improving measurement precision of fog," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 6, pp. 1183–1188, 2000.

[26] S. Sivanandam and S. Deepa, *Principles of soft computing (with CD)*. John Wiley & Sons, 2007.

[27] X. Zeng and T. Martinez, "A noise filtering method using neural networks," in *IEEE International Workshop on Soft Computing Techniques in Instrumentation, Measurement and Related Applications, 2003. SCIMA 2003.* IEEE, 2003, pp. 26–31.

[28] Q. Zhang, Z. Tan, and L. Guo, "Compensation of temperature drift of mems gyroscope using bp neural network," in *2009 International Conference on Information Engineering and Computer Science*. IEEE, 2009, pp. 1–4.

[29] G. Cammarata, S. Cavalieri, and A. Fichera, "A neural network architecture for noise prediction," *Neural Networks*, vol. 8, no. 6, pp. 963–973, 1995.

[30] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.