DESIGN, DEVELOPMENT AND PERFORMANCE EVALUATION OF INTELLIGENT SENSORS

Saroj Kumar Mishra

Department of Electronics & Communication Engineering
National Institute of Technology Rourkela

Design, Development and Performance Evaluation of Intelligent Sensors

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology (Research)
in
Electronics & Communication Engineering
2004-2006

By

Saroj Kumar Mishra

Roll No: 60407003

Under the supervision of

Dr. Ganapati Panda FNAE, FNASc.
Professor

Department of Electronics & Communication Engineering
National Institute of Technology Rourkela

May, 2006

Contents

Department of Electronics & Communication Engineering
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ORISSA, INDIA – 769 008

CERTIFICATE

This is to certify that the thesis titled "***Design, Development and Performance Evaluation of Intelligent Sensors***", submitted to the National Institute of Technology, Rourkela by **Mr. Saroj Kumar Mishra**, Roll No. **60407003** for the award of the degree of Master of Technology (Research) in Electronics & Communication Engineering, is a bona fide record of research work carried out by him under my supervision and guidance.

The candidate has fulfilled all the prescribed requirements.

The thesis, which is based on candidate's own work, has not been submitted elsewhere for a degree/diploma.

In my opinion, the thesis is of standard required for the award of a Master of Technology (Research) degree in Electronics & Communication Engineering.

To the best of my knowledge, he bears a good moral character and decent behavior.

Prof. G. Panda, FNAE, FNASc.
Department of Electronics & Communication Engineering
National Institute of Technology
Rourkela-769 008 (INDIA)
Email: g*anapati.panda@gmail.com*

BIO-DATA OF THE CANDIDATE

Name of the Candidate	:	Saroj Kumar Mishra
Father's Name	:	Bijaya Kumar Mishra
Permanent Address	:	C/O- B.K. Mishra Bapuji Nagar-3rd lane Medical Bank colony Berhampur-760 004. Orissa, India.
Email ID	:	sarojkumishra@yahoo.co.in

ACADEMIC QUALIFICATION

- Continuing **M. Tech. (Research)** in Electronics & Communication Engineering, National Institute of Technology Rourkela, Orissa (INDIA).
- **M.Sc.** in Electronics, *Berhampur University*, Orissa (INDIA).
- **B. Sc.** Physics (Honours), Aska Science College, *Berhampur University*, Orissa (INDIA).

PUBLICATION

- Published/Accepted 06 papers in National and International Conferences;
- Communicated 03 papers to International Journals.

ACKNOWLEDGEMENT

I would like to take this opportunity to extend my deepest gratitude to my teacher and supervisor, Prof. Ganapati Panda, for his continuous encouragement and active guidance. I am indebted to him for the valuable time he has spared for me during this work. He is always there to meet and talk about my ideas, to proofread and mark up my research papers and chapters, and to ask me good questions to help me think through my problems.

I am very much thankful to Prof. K.K. Mahapatra, Head, Department of Electronics and Communication Engineering, for his continuous encouragement. Also, I am indebted to him who provided me all official and laboratory facilities.

I am grateful to Dr. S. Meher and Prof. T.K. Dan for his valuable suggestions and comments during this research period.

My sincere thanks go to Prof. G.S. Rath, Dr. S.K. Patra, Prof. S. Pramanik, Dr. B. Majhi, and Prof. B.D. Sahoo whose valuable suggestions helped me a lot in completing this thesis.

I would like to thank Dr. Debi Prasad Das, Scientist, CEERI, Pilani for his kind support during the on period of my thesis work. In addition, let me thank all my friends Sunil, Manas, Sushant, Pankaj, Rana Babu, Vamsi, Alekhika, Babita Madam, Sabuj, Debi, Nilamani sir, Jitendra sir and Peter for their great support and encouragement during the research period. Also, I am thankful to all the non-teaching staffs of ECE Department for their kind cooperation.

During the course of this work, I am supported in part by the research project of ISRO and VLSI-SMDP sponsored by DIT, Govt. of India. I am really thankful to them.

Last but not the least, I take this opportunity to express my regards and obligation to my parents and family members for educating me in all aspects. I can never forget for their unconditional support and encouragement to pursue my interests.

Saroj Kumar Mishra

ABSTRACT

Many electronic devices, instruments and sensors exhibit inherent nonlinear input-output characteristics. Nonlinearity also creeps in due to change in environmental conditions such as temperature and humidity. In addition, aging of the sensors also introduce nonlinearity. Due to such nonlinearities direct digital readout is not possible. As a result the devices or sensors are used only in the linear region of their characteristics. In other words the usable range of these devices gets restricted due to nonlinearity problem. The accuracy of measurement is also affected if the full range of the instrument is used. The nonlinearity present in the characteristics is usually time-varying and unpredictable as it depends on many uncertain factors stated above. Hence the prime objective of the thesis is to study the nonlinearity problem associated with these devices and suggest novel methods of circumventing these effects by suitably designing intelligent systems. In the present investigation, sensors such as, capacitive pressure sensor (CPS), linear variable differential transformer (LVDT) and control valve actuator are selected for adaptive linearization. In corporation of the intelligent inverse model is in series with the nonlinear device alleviates the associated nonlinearity and permits accurate measurement for the full dynamic range. The existing fixed type nonlinearity compensators are not be effective for time varying nonlinear devices. However, adaptive techniques using Artificial Neural Network (ANN) are better candidates for solving such problems. Therefore, in the present thesis, the following major tasks have been carried out.

(i) Development of adaptive system identification model (direct model) of the CPS for analyzing the inherent time varying nonlinearity present in their characteristics mainly due to the variation in temperatures,

(ii) Development of an inverse model of the CPS for the full range of operations and placing in series with the device for achieving time varying nonlinearity compensation.

These models (direct and inverse) have been developed using:

(a) Adaptive tap-delay filter
(b) The Multi-Layer Perceptron (MLP)
(c) The Radial Basis Function based Neural Network (RBFNN)
(d) The Functional-Link Artificial Neural Network (FLANN)

(iii) To assess the performance of the direct and inverse models in real-time implementation, fixed-point analysis is required. Hence, the effect of fixed-point implementation of the low complexity FLANN based direct and inverse model of the CPS has been analyzed. The performance of fixed-point FLANN model at different word lengths has been evaluated.

(iv) An adaptive nonlinear compensator (or inverse model) of the LVDT by taking the experimental data has been developed. Different ANN models used to develop the adaptive nonlinear compensators are:

(a) The MLP
(b) The FLANN
(c) The Cascaded FLANN (CFLANN)

This models when connected in series with the LVDT offers extended linearity characteristics. The overall system provides accurate measurement for the whole range. The performance comparison in terms of computational complexity and accuracy of measurement of various models has been made.

(v) Nonlinearity compensator of equal percentage control valve actuator have also been developed using the following adaptive techniques:

(a) The LMS filter
(b) The MLP
(c) The FLANN

The performance of various adaptive models and their comparisons has been assessed using three typical sensors. It is demonstrated that the FLANN model provides quite satisfactory performance and involves less complexity. Hence the FLANN based intelligent models are preferable compared to the MLP and RBFNN based compensators.

LIST OF ABBREVIATIONS

ADC	Analog-to-Digital Converter
AKF	Adaptive Kalman Filtering
ANN	Artificial Neural Network
ASP	Adaptive Signal Processing
BP	Back-Propagation algorithm
CFLANN	Cascaded Functional-Link Artificial Neural Network
CPS	Capacitive Pressure Sensor
DAC	Digital-to-Analog Converter
FL	Fuzzy Logic
FLANN	Functional-Link Artificial Neural Network
FPGA	Field Programmable Gate Array
FSO	Full-Scale Output
LMS	Least Mean Square
LVDT	Linear Variable Differential Transformer
MDS	Minimum Detectable Signal
MLP	Multi-Layer Perceptron
MR	Measured Range
MSE	Mean Square Error
PEL	Percentage of Linearity
PIM	Plug-In-Module
RBFNN	Radial Basis Function based Neural Network
RLS	Recursive Least Square
SCC	Switched Capacitor Circuit
VLSI	Very Large Scale Integration

LIST OF FIGURES

Chapter

1

Introduction

THE SENSORS are devices which, for the purpose of measurement, turn physical input quantities into electrical output signals, their output-input and output-time relationship being predictable to a known degree of accuracy at specified environmental conditions. The definition of sensors or transducers according to The Instrument Society of America is *"a device which provides a usable output in response to a specified measurand"*. Here the output is an 'electrical quantity' and measurand is a 'physical quantity' [1.1-1.3]. It can also be defined as an element that senses a variation in input energy to produce a variation in another or same form of energy is called a sensor, whereas, transducer involves a transduction principle which converts a specified measurand into an usable output.

This Chapter deals with the fundamental of the sensors and their characteristics. Section 1.1 deals with the sensor fundamental along with its characteristics. The literature survey is discussed in Section 1.2. Problem formulation of the thesis is depicted in Section 1.3. Finally the Chapter-wise organization of the thesis is presented in Section 1.4.

1.1 Sensor Fundamentals

The sensor consists of several elements or blocks such as sensing element, signal conditioning element, signal processing element and data presentation element [1.2].

Sensing element: This is in contact with the process and gives an output which depends in some way on the variable to be measured. Examples are: thermocouple where millivolt emf depends on temperature, capacitive pressure sensor where capacitance of a chamber depends on pressure, linear variable differential transformer where emf at the secondary coil depends on displacement, etc.

If there is more than one sensing element in a system, the element in contact with the process is termed the primary sensing element, the others secondary sensing elements.

Signal conditioning element: This takes the output of the sensing element and converts it into a form more suitable for further processing, usually a dc voltage, dc current or frequency signal. Examples are: deflection bridge which converts an impedance change into a voltage change, amplifier which amplifies millivolts to volts, etc.

Signal processing element: This takes the output of the conditioning element and converts it into a form more suitable for presentation. Examples are: analog-to-digital converter (ADC) which converts a voltage into a digital form for input to a computer; a microcontroller which calculates the measured value of the variable from the incoming digital data.

Data presentation element: This presents the measured value in a form which can be easily recognized by the observer. Examples are: a simple pointer-scale indicator, chart recorder, and alphanumeric display etc.

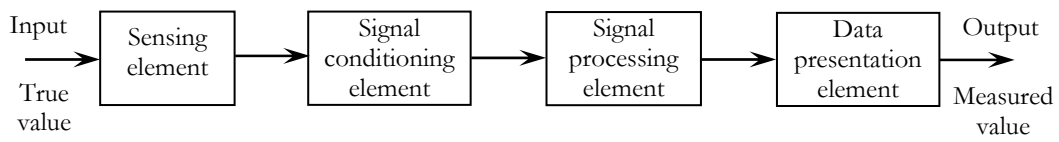


Fig. 1.1 General structure of a sensor

1.1.1 Characteristics of Sensor

Sensors or all measurement systems, have two general characteristics [1.3], i.e. (i) *static characteristics*, and (ii) *dynamic characteristics*.

The ***Static characteristics*** comprise of:

(a) Accuracy: These characteristics specified by error. And it is given by,

$$\varepsilon_a\% = \left(x_m - x_t\right)/x_t \times 100 \tag{1.1}$$

where *t* is for true value, *m* for measured value and variable *x* stands for measurand. This is often expressed for the full scale output and is given by,

$$\varepsilon_{fso}\% = \left(x_m - x_t\right)\cdot 100 \Big/ x_{fso} \tag{1.2}$$

Obviously,

$$\left|\varepsilon_{fso}\right| \le \left|\varepsilon_a\right|$$

For multi error systems the overall performance in terms of error can be assessed either through (i) the worst case approach which assumes that all errors add in the same direction so that the overall error is very high being the linear sum of all the performance errors, or through (ii) the root mean square approach which is optimistic as well as practical, when the total performance error can be assessed as,

$$\varepsilon_o = \left\{\sum_i \varepsilon_i^2\right\}^{\frac{1}{2}} \tag{1.3}$$

(b) Precision: It describes how far a measured quantity is reproducible as also how close it is to the true value.

A term *repeatability* is similar to precision which is the difference in output *y* at a given value of the input *x* when this is obtained in two consecutive measurements. It may be expressed as *% Full-scale Output (FSO)*. Fig. 1.2 shows the repeatability.

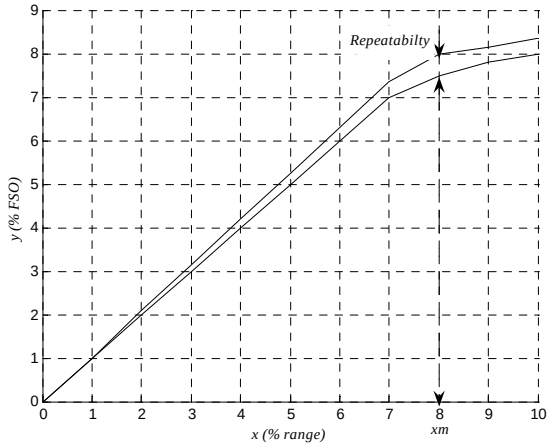


Fig. 1.2 Repeatability in *y-x* coordinates

(c) Resolution: It is defined as the smallest incremental change in the input that would produce a detectable change in the output. This is often expressed as percentage of the measured range, *MR*. The measured range is defined as the difference of the maximum input and the minimum input, $x_{\max} - x_{\min} = MR$. For a detectable output Δy, if the minimum change in x is $(\Delta x)_{\min}$, then the maximum resolution is

$$R_{\max}(\%) = 100(\Delta x)_{\min} / MR \tag{1.4}$$

Over the range of operation an average resolution has also been defined as

$$R_{av}(\%) = 100\sum_{i=1}^{N} \Delta x_i / (n \cdot MR) \tag{1.5}$$

(d) Minimum Detectable Signal (MDS): Noise in a sensor occurs due to many reasons - internal sources, or, fluctuations due to externally generated mechanical and electromagnetic influences. Details of noise are considered on individual merits and often an equivalent noise source is considered for test purposes.

If the input does not contain any noise the minimum signal level that produces a detectable output from the sensor is determined by its noise performance or noise characteristics. For this, the equivalent noise source is connected to the input side of the ideal noiseless sensor to yield an output which is the actual output of the sensor. The MDS is then taken as the RMS equivalent input noise. When a signal exceeds this value it would be detectable signal.

(e)Threshold: This is the smallest input change that produces a detectable output at zero value condition of the measurand.

(f) Sensitivity (*S*): This is defined as the ratio of the incremental output (Δy) to incremental input (Δx), i.e.

$$S = \Delta y / \Delta x \tag{1.6}$$

In normalized form this is written as

$$S_n = \left(\Delta y / \Delta x\right) / \left(y / x\right) \tag{1.7}$$

If sensitivity changes or the output level changes with time, temperature and/or any other parameters without any change in input level, drift is said to occur in the system which often leads to instability.

(g) Selectivity and Specificity: The output of a sensor may change when affected by environmental parameters or other variables and this may appear as an unwanted signal. The sensor is then said to be non-selective. It is customary to define selectivity or specificity by considering a system of n sensors each with output y_k (k=1, 2, …, n). The partial sensitivity S_{jk} is defined as the measure of the k^{th} sensor to these other interfering quantities or variables x_j as

$$S_{jk} = \Delta y_k / \Delta x_j \tag{1.8}$$

A selectivity matrix would thus be obtained with S_{jk} as jk^{th} entry. Obviously, an ideal selective system will have only the diagonal entries S_{jj} in the selectivity matrix. An ideally specific system is characterized by having a matrix with a single element in the diagonal. The selectivity λ is:

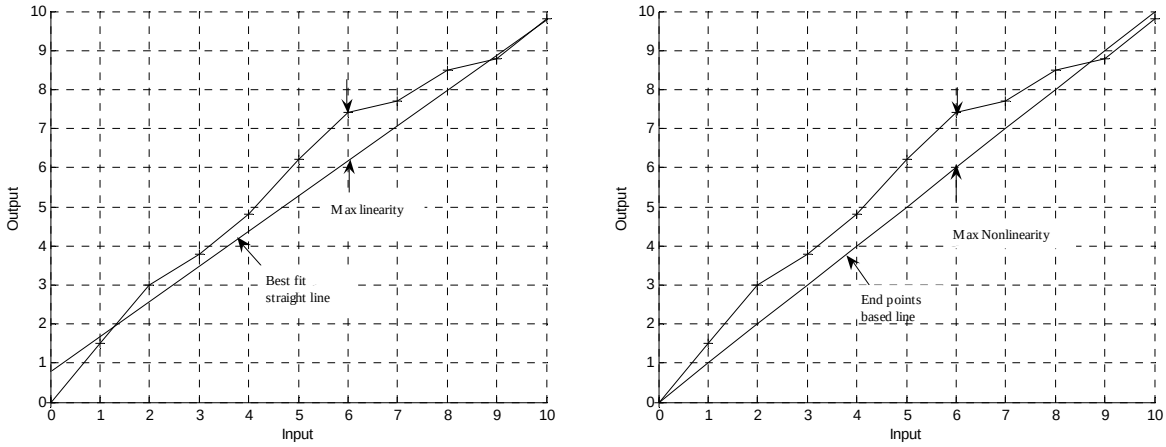
$$\lambda = Min\left[\frac{S_{jj}}{\sum_{k=1}^{n}\left|S_{jk}\right| - \left|S_{jj}\right|}\right],\ j = 1, 2, ..., n \tag{1.9}$$

Thus, for a selective group the denominator tends to zero and $\lambda \to \infty$. Also, specificity is a special case of selectivity.

(h) Nonlinearity: The deviation from linearity, which itself is defined in terms of superposition principles, is expressed as a percentage of the full scale output at a given value of the input. Nonlinearity can, however, be specified in two different ways:

- deviation from best fit straight line obtained by regression analysis, and
- deviation from a straight line joining the end points of the scale.

These are shown in Figs. 1.3 (a) and (b). The maximum nonlinearity in the first method is always less than the maximum nonlinearity in the second method.



a b Fig. 1.3 Nonlinearity with (a) best-fit characteristics, (b) end-points characteristics

A consequence of nonlinearity is distortion, which is defined as the deviation from an expected output of the sensor or transducer. It also occurs due to presence of additional input components.

If deviation at each point of the experimental curve is negligibly small from the corresponding point in the theoretical curve or a curve using least square or other standard fits, the sensor is said to have *conformance* which is quantitatively expressed in *% FSO* at any given value of the input.

(i) Hysteresis: It is the difference in the output of the sensor y for a given input x when x reaches the value in upscale and downscale directions as shown in Fig. 1.4. The causes are different for different types of sensors. In magnetic types, it is lag in alignment of the dipoles, while in semiconductor types it is the injection type slow traps producing the effect, and so on.

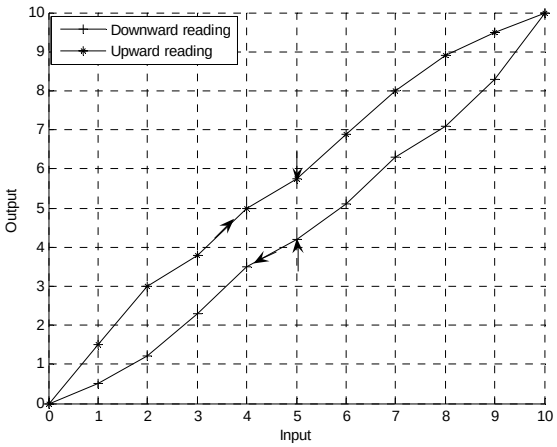


Fig. 1.4 Hysteresis curve

(j) Output Impedance: It is a characteristic, which is to be considered on individual merit. It puts great restriction in interfacing, specifically in the choice of the succeeding stage.

(k) Isolation and Grounding: Isolation is necessary to eliminate or at least reduce undesirable electrical, magnetic, electromagnetic and mechanical coupling among various parts of the system and between the system and the environment. Similarly, grounding is necessary to establish a common node among different parts of the system with respect to which potential of any point in the system remains constant.

The ***dynamic characteristics*** involve determination of transfer function, frequency response, impulse response and step response and then evaluation of the time-dependent outputs. The two important parameters in these connections are:

- Fidelity determined by dynamic error, and
- Speed of response determined by lag.

For determining the *dynamic characteristics* different inputs are given to the sensor and the response characteristics are to be studied. With step input, the specifications in terms of the time constant of the sensor are to be made. Impulse response as well as its Fourier transform are also to be considered for time domain as well as frequency domain studies.

Environmental Parameters: The external variables such as temperatures, pressure, humidity, vibration, etc. which affects the performance of the sensor. Aging is also an important parameter

of the sensor. These parameters are not the ones that are to be sensed. For any environmental parameters, the performance of the sensor can be studied in terms of its effect on the *static* and *dynamic characteristics*. For this study, one environmental parameter at a time is considered variable while others are fixed.

1.2 Literature Survey

The capacitive pressure sensor (CPS), in which the capacitance of a chamber changes with application of pressure finds extensive applications because of its low power consumption and high sensitivity. It is observed that many sensors exhibit nonlinear input-output characteristics. Due to such nonlinearities direct digital readout is not possible. As a result we employ the sensors only in the linear region of their characteristics. Generally, CPSs are known to have no turn-on temperature drift, high sensitivity, robust structure and is less sensitive to environmental effects [1.4]. However, its output is nonlinear with respect to input changes and the sensitivity in the near linear region is not high enough to ignore many stray capacitance effects [1.5]. Several studies have been carried out to fulfill the demand of low cost, high sensitivity, resolution and mass producibility. The nonlinear response characteristics of these sensors give rise to several difficulties including on-chip interface, direct digital readout and calibration. Nonlinearity also creeps in due to change in environmental conditions such as temperature and humidity. In addition aging of the sensors also introduces nonlinearity. Hence, to achieve an easy digital interface for direct readout, the response characteristics of the sensor should be linear and independent of variations in the environmental conditions such as temperature, humidity and aging. To compensate the nonlinear response characteristics of the CPS, several techniques have been suggested. A switched-capacitor charge balancing technique [1.6], a ROM based look-up table method and a nonlinear encoding scheme have been proposed [1.7]. And also the demand of low-cost and small range linear output requirements have led to the development of the piezo-resistance bridge-based integrated pressure sensor. A thin-diaphragm diffused piezo-resistive pressure sensor for biomedical instrumentation has been developed using monolithic IC techniques. Integrated circuit technology offers several important advantages in the fabrication of pressure sensors for biomedical instrumentation. In particular, the ability to control geometries precisely on very small dimensions promises to allow the fabrication of sensors which combine small size with reliability and stability.

Furthermore, the ability to batch fabricate such transducers should make them relatively inexpensive. Pressure transducers which depend on the piezo-resistive effect have several advantages over those based on other pressure sensitive effects. Piezo-resistive transducers operate at low stress levels, exhibit a resistance change which is a linear function of pressure over a wide range and have high sensitivity to pressure changes. Because of this high sensitivity, stylus arrangements to achieve pressure multiplication are not needed; this fact simplifies the assembly into a completed package [1.8]. Also for nonlinearity estimation and to obtain a direct digital readout of a CPS, an Artificial Neural Network (ANN) based modeling technique has been proposed with quite satisfactory performance [1.9,1.10]; however without any consideration of change in the ambient temperature. When we take temperature into consideration, and the ambient temperature changes frequently, the situation becomes very complicated and in this case the problem becomes two-dimensional (2-D) and adaptive signal processing techniques are required to make necessary corrections to obtain a correct digital readout. A scheme of microcomputer based 2-D lookup table method has been proposed [1.11]. Another method of Δ-Σ demodulator and complex signal processing techniques for the sensor model have also been reported [1.12].

By taking temperature into consideration, a 3-layer Multi-Layer Perceptron (MLP) based ANN has been proposed for auto-calibration and nonlinear compensation of a CPS [1.13], under variation of ambient temperature. In this method switched capacitor circuit (SCC) converts the change in capacitance into an equivalent voltage. The intelligent behavior is implanted into the sensor by training the ANN to adapt to a temperature range of -20^0 C to 70^0 C. In the direct modeling, the ANN is trained in a parallel mode to estimate the capacitance of the CPS. This model may be used for the purpose of on-line fault detection and quality control of the sensor during its production. In the inverse modeling, the ANN is trained in a series mode to estimate the applied pressure which is independent of ambient temperature. A plug-in-module (PIM) is proposed to implement the scheme on-line.

In many control system applications Linear Variable Differential Transformer (LVDT) plays an important role to measure the displacement [1.1,1.14]. The performance of the control system depends on the performance of the sensing element. It is observed that the LVDT exhibits the same nonlinear input-output characteristics. Due to such nonlinearities direct digital readout is not possible. As a result we employ the LVDTs

only in the linear region of their characteristics. In other words their usable range gets restricted due to the presence of nonlinearity. If the LVDT is used for full range of its nonlinear characteristics, accuracy of measurement is severely affected.

The nonlinearity present is usually time-varying and unpredictable as it depends on many uncertain factors. Attempts have been made by many researchers to increase the range of linearity of LVDT. In the conventional design of LVDT sophisticated and precise winding machines are used to compensate the nonlinearity effects. The nonlinearity compensation can also be achieved by square coil method in which the core moves perpendicular to the axis instead of along the axis [1.14]. A self-compensated LVDT has been modeled using a dual secondary coil which is insensitive to the variation in excitation current and frequency [1.15]. Some DSP techniques have been employed in LVDT to achieve better sensitivity and to implement the signal conditioning circuits [1.16-1.18].

The control valve actuators are the most common actuators in the process industries. They are found in process plants, manipulating flows to maintain controlled variable at their set points. A control valve acts as a *variable restriction* in a process pipe. By changing its opening it changes the resistance to flow and, thus, the flow itself [1.19]. As discussed earlier sections, many instruments or sensors exhibit inherent nonlinear input-output characteristics. The same effect is also observed in case of the control valve actuators. The equal percentage control valve actuator devices have nonlinearities in their *inherent flow characteristics*. But when it is put in a pipe line, it acts as nearly linear control valve, called *installed characteristics*. But this linearity depends on the pressure difference across the valve. As the pressure difference across the valve varies, it is always required to compensate to get very accurate linear characteristics. These nonlinearities cannot be compensated using feedback linearization techniques because they do not appear in the feedback path but rather in the feed-forward path. Some of other fundamental actuator nonlinearities include friction, deadzone, saturation, backlash and hysteresis. Amongst these the friction, the deadzone and the saturation are *static nonlinearities*, whereas the backlash and the hysteresis are *dynamic nonlinearities*.

Attempts have been made by many researchers to compensate the *static* and *dynamic nonlinearities* present in the actuators. The nonlinear effect due to friction is common in all mechanical

systems. So it cannot be avoided in control systems. Different friction models and their compensation techniques are proposed in [1.20,1.21]. [1.22] described an observer based friction compensation for a dynamical model. Robust adaptive friction compensation of the static friction model is given in [1.23], and using reinforcement adaptive learning neural networks in [1.24]. The deadzone effect is present in many electro-mechanical systems. Standard techniques for overcoming deadzone are variable structure control and dithering [1.25]. The deadzone compensation in nonlinear systems using artificial neural networks (ANN) is proposed in [1.26]. Backlash and hysteresis nonlinearities usually appear in the feedforward path, and their compensators are based on applications of dynamic inversion of nonlinear systems. Tao and Kokotovic [1.27], developed adaptive backlash and hysteresis compensation. The dynamic inversion is well known technique in aerospace applications. A compensated inverse dynamics approach using adaptive and robust control techniques is presented in [1.28]. The magnitude and the rate of actuator inputs exhibit the saturation nonlinearities. When an actuator has reached such an input limit, it is said to be "saturated", since efforts to further increase the actuator output would not result in any variation of the output. To overcome this problem, the general actuator saturation compensator scheme is developed in [1.29]; Design of a robust anti-windup controller based on the Lyapunov approach to accommodate the constraints and disturbance is described in [1.30]. Both static and dynamic compensators using neural networks are considered in [1.31].

1.3 Problem Formulation

The sensors exhibit nonlinear characteristics which limit the dynamic range of these devices. As a result the direct digital readout of the output is not possible for the whole input range of the sensors. In addition the full potential of the sensors cannot be utilized. Thus there is a real challenge for designing and implementing novel sensors which circumvent the nonlinearity problems associated with them. In recent past few attempts have been made but not much has been achieved in this direction. The second problem is the accuracy of measurements in these sensors which is greatly affected by aging of the sensor, temperature and humidity variations.

Both these issues detoriate the accuracy of measurement. Therefore nonlinearity compensation is required to achieve accurate measurement under adverse conditions. Most of the existing nonlinearity compensation techniques work well for fixed and known type of nonlinearities. But

in actual practice the nonlinearity behavior of the sensors changes with time and is usually unknown. The adaptive inverse model of the CPS can be designed and connected in series with these devices, so that the associated nonlinearity can be compensated and direct digital readout is possible for the whole input range. A scheme of direct modeling and inverse modeling of the CPS using different ANN structure has been developed. The direct modeling is proposed for calibration of inputs and estimation of internal parameters of the CPS. The purpose of the direct model is to obtain an ANN model of the CPS in such a way that the outputs of the CPS and the ANN match closely. Once a model of the CPS is available, it may be used for fault detection of the sensor. The CPS (interfaced with SCC) output provides a voltage signal proportional to the capacitance change due to the applied input pressure. And the inverse modeling is proposed for estimation of applied input pressure. The adaptive inverse model can be developed using Least Mean Square (LMS) algorithm, Recursive Least Square (RLS) algorithm [1.32,1.33], and Adaptive Kalman Filtering (AKF) [1.32]. The intelligent techniques such as Artificial Neural Networks (ANN) [1.34-1.36], Fuzzy Logic (FL) and Neuro-Fuzzy algorithms [1.37] are also potential candidates for developing adaptive inverse model of these devices.

In this thesis the following burning issues relating CPS, LVDT, and control valve actuators are analyzed and solution to these problems has been suggested. The adaptive modelings of these instruments have been carried out using LMS and different ANN techniques. The *direct modeling* of these devices provides information about the degree of nonlinearity and how it changes with the change in environment. The *adaptive inverse model* of these nonlinear instruments have also been developed taking environmental conditions into account. Different efficient ANN structures such as Multi-Layer Perceptron (MLP) [1.34,1.38], Functional-Link ANN (FLANN) [1.39-1.41], and Radial Basis Function based NN (RBFNN) [1.34-1.37,1.42] have been designed for nonlinear compensation and their performance has been assessed through fictitious data of the CPS. The proposed techniques are novel ones and are shown to yield excellent performance. The simulation results for CPS, LVDT and control valve actuator for achieving adaptive linearization under different conditions are also presented in this thesis.

The LVDT exhibits nonlinearity behaviour similar to the CPS when the core is moved towards any one of the secondary coils. In the primary coil region (middle) of the characteristics, the core movement is almost linear. Because of that, the range of operation is limited to the primary coil region only. Different attempts have been made

for the linearity enhancement of LVDT. In this thesis, adaptive nonlinearity compensators have been designed using different ANN structures such as MLP, FLANN and cascaded FLANN (CFLANN) based model.

The control valve plays an important role in process industries, the linear type is the most suitable control valve due to its linear relationship between input and output. But the linear type is associated with constant sensitivity for which the pressure is also remains constant. So the application gets restricted. To compensate the nonlinearity present in the equal percentage control valve, an adaptive inverse model have been proposed by using different adaptive algorithms such as LMS, MLP and FLANN techniques. A comparative study has been made between these three models, and observed that the LMS based inverse model is more efficient and computationally less complex.

1.4 Chapter-wise Organization

The chapter-wise organization of the thesis is outlined below.

Chapter-1 Introduction

Brief Introduction

1.1 Sensor Fundamentals

1.1.1 Characteristics of Sensors

1.2 Literature Survey

1.3 Problem Formulation

1.4 Chapter-wise Organization

References

Chapter-2 Intelligent Techniques and Algorithms

Brief Introduction

2.1 Adaptive Techniques

2.1.1 Least Mean Square (LMS) Algorithm

2.1.2 Recursive Least Square (RLS) Algorithm

2.2 Artificial Neural Network (ANN)

2.2.1 Single Neuron Structure

2.2.2 Multilayer Perceptron (MLP)

2.2.3 Functional-Link Artificial Neural Network (FLANN)

Chapter-5 Nonlinear Compensation of LVDT using Different ANN Techniques

Brief Introduction

5.1 Linear Variable Differential Transformer

5.2 Nonlinearity Compensation of LVDT

5.3 Simulation Studies

5.3.1 MLP based Nonlinearity Compensator

5.3.2 FLANN based Nonlinearity Compensator

5.3.3 CFLANN based Nonlinearity Compensator

5.4 Computational Complexity

5.5 Summary and Discussion

References

Chapter-6 Adaptive Nonlinear Compensation of Control Valve Actuator using Soft Computing Techniques

Brief Introduction

6.1 Control Valve Actuator

6.2 Adaptive Inverse Models

6.2.1 LMS based Inverse Model

6.2.2 MLP based Inverse Model

6.2.3 FLANN based Inverse Model

6.3 Summary and Discussion

References

Chapter-7 Conclusions

Brief Introduction

7.1 Scope for Further Research Work

Publications out of Research Work

REFERENCES

[1.1] Hermann K.P. Neubert, "*Instrument Transducers: An Introduction to their performance & design*", 2nd edition, Oxford University Press.

[1.2] John P. Bentley, "*Principles of Measurement Systems*", 3rd edition, Pearson Education.

[1.3] D Patranabis, "*Sensors and Transducers*", 1st edition, Wheeler Publishing, 1997.

[1.4] Y.S. Lee and K.D. Wise, "A batch-fabricated silicon capacitive pressure sensor with low temperature," *IEEE Trans. Electron Devices*, vol. ED-29, pp. 42-48, Jan 1984.

[1.5] Ko, W.H.; Qiang Wang, "Touch mode capacitive pressure sensors for industrial applications," *IEEE Proc. Tenth Annual International Workshop on Microelectromechanical Systems MEMS '97*, pp.284 – 289, Jan. 1997.

[1.6] H. Matsumoto, H. Shimizu and K. Watanabe, "A switched capacitor charge balancing analog-to-digital converters and its application to capacitance measurements," *IEEE Trans. Instrumentation &. Measurement,* Vol. IM-36, pp.873-878, Dec.1987.

[1.7] M. Yamada, T. Takabayashi, S. I. Notoyama and K. Watanabe, "A switched capacitor interface for capacitive pressure sensors," *IEEE Trans. Instrumentation &. Measurement,* Vol.41, pp.81-86, Feb.1992.

[1.8] S. Samaun, K. D. Wise, E. D. Nielsen and J. Angel, "An IC piezoresistive pressure sensor for biomedical instrumentation," *IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, Vol. XIV, Feb 1971.

[1.9] J.C. Patra, G. Panda and R. Baliarsingh, "Artificial neural network based nonlinearity estimation of pressure sensors," *IEEE Trans. Instrumentation &. Measurement,* vol.43, pp.874-881, Dec.1994.

[1.10] G. Panda, S.K. Maharatha and J.K. Satapathy, "On the Development of an Intelligent Sensor using Nonlinear Identification and ANN Techniques," *IETE Technical Review*, Vol. 3, Nos. 4 & 5, July-Oct 1996, pp.247-255.

[1.11] P.N. Mahana and F.N. Trofimenkoff, "Transducer output signal processing using eight-bit microcomputer," *IEEE Trans. Instrumentation &. Measurement,* vol. IM-35, pp.182-186, June 1986.

[1.12] M. Yamada and K. Watanabe, "A capacitive pressure sensor interface using oversampling Δ-Σ demodulation techniques," *IEEE Trans. Instrumentation &. Measurement,* vol.46, pp.3-7, Feb 1997.

[1.13] J.C. Patra, Alex C. Kot, and G. Panda, "An Intelligent Pressure Sensor using Neural Networks," *IEEE Trans. Instrumentation &. Measurement,* vol.49, No.4, pp.829-834, Aug.2000.

[1.14] Y. Kano, S. Hasebe, and H. Miyaji, "New linear variable differential transformer with square coils," *IEEE Transactions on Magnetics*, Vol. 26, Issue: 5, pp.2020 – 2022, Sep 1990.

[1.15] S.C. Saxena, and S.B.L. Sexena, "A self-compensated smart LVDT transducer," *IEEE Trans. Instrumentation &. Measurement,* Vol. 38, Issue: 3, pp.748 – 753, June 1989.

[1.16] G. Y. Tian, Z, X. Zhao, R. W. Baines, N. Zhang, "Computational algorithms foe linear variable differential transformers (LVDTs)" *IEE Proc.-Sci. Meas. Technol.,* Vol. 144, No. 4,pp 189-192, July 1997.

[1.17] D. Crescini, A. Flammini, D. Marioli, and A. Taroni, "Application of an FFT-based algorithm to signal processing of LVDT position sensors" *IEEE Trans. Instrumentation &. Measurement,* Vol. 47, No. 5, pp. 1119-1123 Oct. 1998.

[1.18] R. M. Ford, R. S. Weissbach, and D. R. Loker, "A novel DSP-based LVDT signal conditioner" *IEEE Trans. Instrumentation &. Measurement,* Vol. 50, No. 3, pp. 768-774, June 2001.

[1.19] D. R. Coughanowr, "*Process Systems Analysis and Control*", 2nd edition, McGraw-Hill international editions, 1991.

[1.20] B. Armstrong-Htlouvry, P. Dupont, and C. Canudas de Wit, "A survey of models, analysis tools and compensation methods for the control of machines with fiction," *Automatica*, vol.30, no.7, pp.1083-1138, 1994.

[1.21] C. Canudas de Wit, P. Noel, A. Aubin, and B. Brogliato, "Adaptive friction compensation in robot manipulators: Low velocities," *Int. J. Robot. Res.*, vol.10, no.3, Jun. 1991.

[1.22] H. Olsson and K. J. Astrom, "Observer-based fiction compensation," *Proc. IEEE Conf. Decis. and Control*, Kobe, Japan, Dec. 1996, pp. 4345-4350.

[1.23] S. W. Lee and J. H. Kim, "Robust adaptive stick-slip fiction compensation," *IEEE Trans. Ind. Electron*, vol. 42, no.5, pp. 474-479, Oct. 1995.

[1.24] Y. H. Kim and F. L. Lewis, "Reinforcement adaptive learning neural-net-based friction compensation control for high speed and precision," *IEEE Trans. Control Sysi. Technology*, vol. 8, no. l, pp.118-126, Jan. 2000.

[1.25] C. A. Desoer and S. M. Shahruz, "Stability of dithered nonlinear systems with backlash or hysteresis," *Int. J. Contr.*, vol. 43, no. 4, pp. 1045-1060, 1986.

[1.26] R. R. Selmic, F. L. Lewis, "Deadzone Compensation in Nonlinear Systems using Neural Network", *Proc. of the 37th IEEE conference of Decision & control*, Dec, 1998.

[1.27] G. Tao and P. V. Kokotovic, "*Adaptive Control of Systems with Actuator and Sensor Nonlinearities*", John Wiley & Sons, New York, 1996.

[1.28] Y. D. Song, T. L. Mitchell, and H. Y. Lai, "Control of a class of nonlinear uncertain systems via compensated inverse dynamics approach," *IEEE Trans. Automat. Control*, vol. 39, no. 9, pp. 1866-1871, Sep. 1994.

[1.29] K. J. Astrom and B. Wittenmark, "*Computer-Controlled Systems: Theory and Design*", Prentice Hall, 3rd edition, 1996.

[1.30] W. Niu and M. Tomizuka, "'A Robust Anti-Windup Controller Design for Asymptotic Tracking of Motion Control System Subjected to Actuator Saturation," *The 37th IEEE Conference on Decision and Control*, Tampa, December 1998.

[1.31] R. R. Selmic, V. V. Phoha and F. L. Lewis, "Intelligent control of Actuator Nonlinearities", *Proc. of the 42nd IEEE Conf. on Decis. & Contr.*, Maui, Hawaii USA, Dec, 2003.

[1.32] S. Haykin, "*Adaptive Filter Theory*," 4th Edition, Pearson Education Asia, 2002.

[1.33] B. Widrow and S.D. Sterns, "*Adaptive Signal Processing*" Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1985.

[1.34] S. Haykin, "*Neural Networks: A comprehensive foundation*" 2nd Edition, Pearson Education Asia, 2002.

[1.35] Dayhoff E.J., "*Neural Network Architecture – An Introduction*" Van Norstand Reilold, New York, 1990.

[1.36] Bose N.K., and Liang P., "*Neural Network Fundamentals with Graphs, Algorithms, Applications*", TMH Publishing Company Ltd, 1998.

[1.37] Richard O. Duda, Peter E. Hart and David G. Stork, "*Pattern Classification*", 2nd edition, John Wiley & Sons, INC.2001.

[1.38] Fu L.M., Hsu H.H., and Principe J.C., "Incremental Back Propagation Learning Networks", *IEEE trans. on Neural Network*, vol.7, no.3, pp.757-762, 1996.

[1.39] Carpenter G.A., and Grossberg S., "The Art of Adaptive Pattern Recognition by Self-Organizing Neural Network", *IEEE Computer Mag.*, vol.21, no.3, pp.77-88, March 1988.

[1.40] Pao Y.H., Park G.H., and Sobjic D.J., "Learning and Generalization Characteristics of the Random Vector Function", Neuro Computation, 6: 163-180, 1994.

[1.41] Patra J.C., and Pal R.N., "A Functional Link Artificial Neural Network for Adaptive Channel Equalization", Signal Processing 43(1995) 181-195, vol.43, no.2, May 1995.

[1.42] Jungsik Lee, Beach, C.D., Tepedelenlioglu, N, "Channel equalization using radial basis function network", *IEEE International Conference on Neural Networks*, Vol. 4, Page(s):1924-1928, 3-6 June 1996.

Chapter

2

Intelligent Techniques and Algorithms

IN RECENT YEARS, a growing field of research in "Adaptive Systems" has resulted in a variety of adaptive automations whose characteristics in limited ways resemble certain characteristics of living systems and biological adaptive processes. An adaptive automation is a system whose structure is alterable or adjustable in such a way that its behavior and performance improves by its environment. A simple example of an adaptive system is the automatic gain control used in radio and television receiver. The most important factor in adaptive system is its time-varying and self-adjusting performance. Their characteristic depends upon the input signal. If a signal is applied to the input of adaptive system to test its response characteristic, the system adapts to this specific input and thereby changes its parameters. The adaptation procedure is carried out using different algorithms such as the Least Mean Square (LMS), Recursive Least Square (RLS) [2.1,2.2] etc. In many real world problems these algorithms do not perform satisfactorily. Hence, based on the different neural architecture of human brain,

different Artificial Neural Algorithms are developed [2.3-2.6]. These are capable of mapping the input and output nonlinearly.

This Chapter deals with different types of adaptive algorithms, which are used as a tool to compensate nonlinearity problem of different sensors. The present Chapter is organized as follows: Section 2.1 deals with the adaptive filtering techniques mainly the LMS and RLS algorithms. Different types of Artificial Neural Network (ANN) such as Multi-Layer Perceptron (MLP), Functional-Link ANN (FLANN) and Radial Basis Function based Neural Network (RBFNN) along with their associated training algorithms are discussed in Section.2.2 and its subsections. Finally the summary and discussion is presented in Section 2.3.

2.1 Adaptive Techniques

Filter is a primary subsystem in any signal processing system. Filters are employed to remove undesirable signal components from the desired signal. In Adaptive Filters the coefficients can be changed from time to time depending on the situation. Here the filter updates its coefficients from the knowledge of the past input and the present error. The error is generated from the reference input and actual output. The update procedure depends upon the different algorithms used.

2.1.1 Least Mean Square (LMS) Algorithm

The general architecture of the LMS based adaptive filter is depicted in Fig. 2.1. The X is N^{th} input pattern having one unit delay in each instant. This process is also called as adaptive linear combiner [2.1,2.2]. Let $\mathbf{X}_k = \begin{bmatrix} x_k & x_{k-1} & \dots & x_{k-L+1} \end{bmatrix}^{\mathrm{T}}$ form of the L-by-1 tap input vector. Where L-1 is the number of delay elements; these inputs span a multidimensional space denoted by $\aleph_k$. Correspondingly, the tap weights $\mathbf{W}_k = \begin{bmatrix} w_{0k} & w_{1k} & \dots & w_{(L-1)k} \end{bmatrix}^{\mathrm{T}}$ form the elements of the L-by-1 tap weight vector. The output is represented as,

$$y_k = \sum_{l=0}^{L-1} w_{lk} x_{k-l} \tag{2.1}$$

The output can be represented in vector notation as

$$y_k = \mathbf{X}_k^{\mathrm{T}}\mathbf{W}_k = \mathbf{W}_k^{\mathrm{T}}\mathbf{X}_k \tag{2.2}$$

Generally for the adaptive linear combiner the other data include a "*desired response*" or "*training signal*", d_k. This is accomplished by comparing the output with the desired response to obtain an "*error signal*" e_k and then adjusting or optimizing the weight vector to minimize this signal. The error signal is,

$$e_k = d_k - y_k \tag{2.3}$$

The weights associated with the network are then updated using the LMS algorithm [2.1]. The weight updates equation for n^{th} instant

$$w_k(n+1) = w_k(n) + \Delta w_k(n) \tag{2.4}$$

It can be further derived as

$$w_k(n+1) = w_k(n) + 2\cdot\eta\cdot e_k(n)\cdot\mathbf{X}_k^{\mathrm{T}} \tag{2.5}$$

where η is the learning rate parameter ($0 \leq \eta \leq 1$). This procedure is repeated till the Mean Square Error (MSE) of the network approaches a minimum value. The MSE at the time index k may be defined as, $\xi = E\left[e_k^2\right]$, where $E[.]$ is the expectation value or average of the signal.

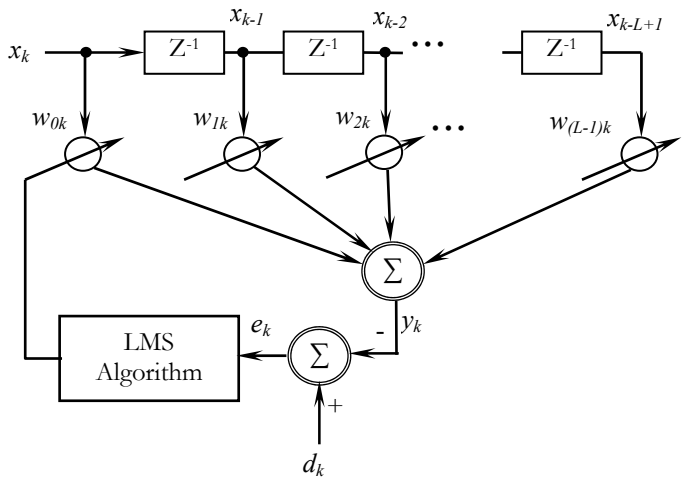


Fig. 2.1 Adaptive filter using LMS algorithm

2.1.2 Recursive Least Square (RLS) Algorithm

The RLS algorithm basically minimizes the sum of squared errors up to last signal sample [2.2]. The recursive weight update equation is given by:

$$w(n+1) = w(n) + \Gamma_{xx}^{-1}(n+1)x(n+1)e(n+1) \tag{2.6}$$

where,

$$\Gamma_{xx}^{-1}(n+1) = \Gamma_{xx}^{-1}(n) - \frac{\left(\Gamma_{xx}^{-1}(n)x(n)\right)^T \Gamma_{xx}^{-1}(n)x(n)}{1 + x^T(n)\Gamma_{xx}^{-1}(n)x^T(n)} \tag{2.7}$$

and,

$$\Gamma_{xx}(n) = \sum_{k=0}^{n} x(k)x^T(k) \tag{2.8}$$

Under low noise conditions the convergence is guaranteed within $2N$ iterations where N is the filter order.

2.2 Artificial Neural Network (ANN)

Artificial neural network (ANN) takes their name from the network of nerve cells in the brain. Recently, ANN has been found to be an important technique for classification and optimization problem [2.3-2.5]. McCulloch and Pitts have developed the neural networks for different computing machines. There are extensive applications of various types of ANN in the field of communication, control and instrumentation. The ANN is capable of performing nonlinear mapping between the input and output space due to its large parallel interconnection between different layers and the nonlinear processing characteristics. An artificial neuron basically consists of a computing element that performs the weighted sum of the input signal and the connecting weight. The sum is added with the bias or threshold and the resultant signal is then passed through a nonlinear function of sigmoid or hyperbolic tangent type. Each neuron is associated with three parameters whose learning can be adjusted; these are the connecting weights, the bias and the slope of the nonlinear function. For the structural point of view a NN may be single layer or it may be multilayer.

In multilayer structure, there is one or many artificial neurons in each layer and for a practical case there may be a number of layers. Each neuron of the one layer is connected to each and every neuron of the next layer. The functional-link ANN is another type of single layer NN. In this type of network the input data is allowed to pass through a functional expansion block where the input data are nonlinearly mapped to more number of points. This is achieved by using trigonometric functions, tensor products or power terms of the input. The output of the functional expansion is then passed through a single neuron.

The learning of the NN may be supervised in the presence of the desired signal or it may be unsupervised when the desired signal is not accessible. Rumelhart developed the Back-propagation (BP) algorithm, which is central to much work on supervised learning in MLP [2.3]. A feed-forward structure with input, output, hidden layers and nonlinear sigmoid functions are used in this type of network. In recent years many different types of learning algorithm using the incremental back-propagation algorithm [2.8], evolutionary learning using the nearest neighbor MLP [2.9] and a fast learning algorithm based on the layer-by-layer optimization procedure [2.10] are suggested in literature. In case of unsupervised learning the input vectors are classified into different clusters such that elements of a cluster are similar to each other in some sense. The method is called competitive learning [2.11], because during learning sets of hidden units compete with each other to become active and perform the weight change. The winning unit increases its weights on those links with high input values and decreases them on those with low input values. This process allows the winning unit to be selective to some input values. Different types of NNs and their learning algorithms are discussed in sequel.

2.2.1 Single Neuron Structure

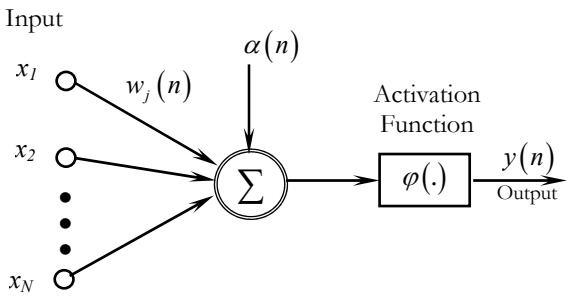


Fig. 2.2. Structure of a single neuron

The basic structure of an artificial neuron is presented in Fig. 2.2. The operation in a neuron involves the computation of the weighted sum of inputs and threshold [2.3-2.5]. The resultant signal is then passed through a nonlinear activation function. This is also called as a perceptron, which is built around a nonlinear neuron; whereas the LMS algorithm described in the preceding sections is built around a linear neuron. The output of the neuron may be represented as,

$$y(n)=\varphi\left[\sum_{j=1}^{N} w_j(n)x_j(n)+\alpha(n)\right] \tag{2.9}$$

where $\alpha(n)$ is the threshold to the neurons at the first layer, $w_j(n)$ is the weight associated with the j^{th} input, N is the no. of inputs to the neuron and $\varphi(.)$ is the nonlinear activation function. Different types of nonlinear function are shown in Fig. 2.3.

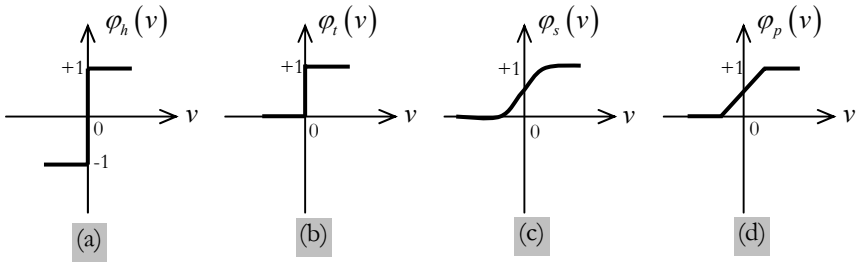


Fig. 2.3. Different types of nonlinear activation function,
(a) Signum function or hard limiter,
(b) Threshold function,
(c) Sigmoid function,
(d) Piecewise Linear

Signum Function: For this type of activation function, we have

$$\varphi(v)=\begin{cases}1 & if\ v>0\\ 0 & if\ v=0\\ -1 & if\ v<0\end{cases} \tag{2.10}$$

Threshold Function: This function is represented as,

$$\varphi(v)=\begin{cases}1 & if\ v\geq 0\\ 0 & if\ v<0\end{cases} \tag{2.11}$$

Sigmoid Function: This function is S-shaped, is the most common form of the activation function used in artificial neural network. It is a function that exhibits a graceful balance between linear and nonlinear behaviour.

$$\varphi(v) = \frac{1}{1+e^{-av}} \tag{2.12}$$

where v is the input to the sigmoid function and a is the slope of the sigmoid function. For the steady convergence a proper choice of a is required.

Piecewise-Linear Function: This function is

$$\varphi(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq +\frac{1}{2} \end{cases} \tag{2.13}$$

where the amplification factor inside the linear region of operation is assumed to be unity. This can be viewed as an approximation to a nonlinear amplifier.

2.2.2 Multi-Layer Perceptron (MLP)

In the multilayer neural network or multilayer perceptron (MLP), the input signal propagates through the network in a forward direction, on a layer-by-layer basis. This network has been applied successfully to solve some difficult and diverse problems by training in a supervised manner with a highly popular algorithm known as the error back-propagation algorithm [2.3,2.4]. The scheme of MLP using four layers is shown in Fig. 2.4. $x_i(n)$ represent the input to the network, f_j and f_k represent the output of the two hidden layers and $y_l(n)$ represents the output of the final layer of the neural network. The connecting weights between the input to the first hidden layer, first to second hidden layer and the second hidden layer to the output layers are represented by w_{ij}, w_{jk} and w_{kl} respectively.

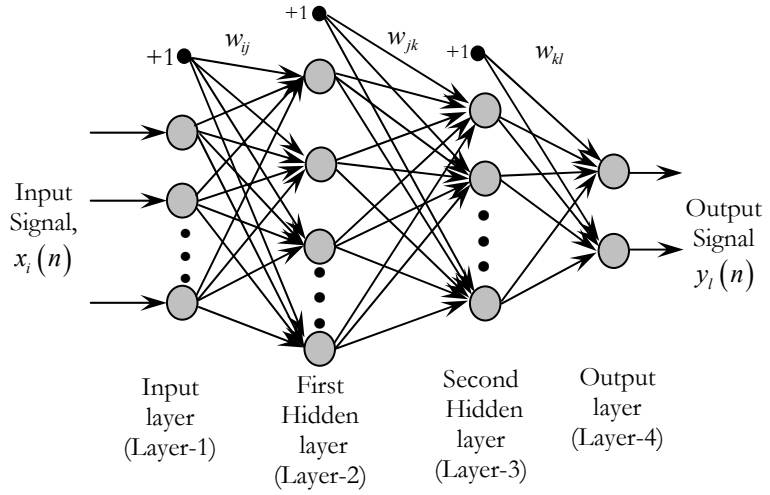


Fig. 2.4 Structure of multilayer perceptron

If P_1 is the number of neurons in the first hidden layer, each element of the output vector of first hidden layer may be calculated as,

$$f_j = \varphi_j\left[\sum_{i=1}^{N} w_{ij}x_i(n) + \alpha_j\right], \quad i = 1,2,3,\ldots N,\ j = 1,2,3,\ldots P_1 \tag{2.14}$$

where α_j is the threshold to the neurons of the first hidden layer, N is the no. of inputs and $\varphi(.)$ is the nonlinear activation function in the first hidden layer of (2.12) type. The time index n has been dropped to make the equations simpler. Let P_2 be the number of neurons in the second hidden layer. The output of this layer is represented as, f_k and may be written as

$$f_k = \varphi_k\left[\sum_{j=1}^{P_1} w_{jk}f_j + \alpha_k\right], \quad k=1, 2, 3, \ldots, P_2 \tag{2.15}$$

where, α_k is the threshold to the neurons of the second hidden layer. The output of the final output layer can be calculated as

$$y_l(n) = \varphi_l\left[\sum_{k=1}^{P_2} w_{kl}f_k + \alpha_l\right], \quad l=1, 2, 3, \ldots, P_3 \tag{2.16}$$

where, α_l is the threshold to the neuron of the final layer and P_3 is the no. of neurons in the output layer. The output of the MLP may be expressed as

$$y_l(n) = \varphi_n\left[\sum_{k=1}^{P_2} w_{kl}\varphi_k\left(\sum_{j=1}^{P_1} w_{jk}\varphi_j\left\{\sum_{i=1}^{N} w_{ij}x_i(n)+\alpha_j\right\}+\alpha_k\right)+\alpha_l\right] \tag{2.17}$$

Back-propagation (BP) Algorithm

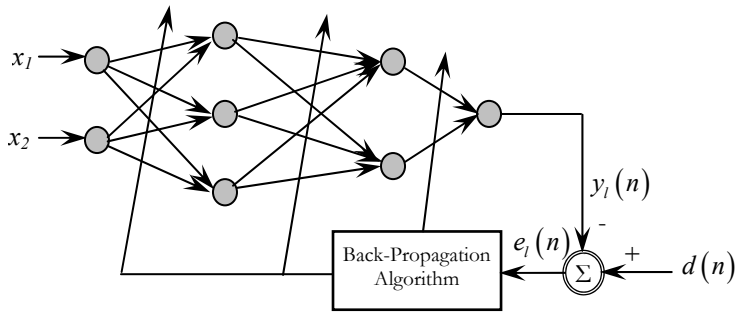


Fig. 2.5 Neural network using BP algorithm

An MLP network with 2-3-2-1 neurons (2, 3, 2 and 1 denote the number of neurons in the input layer, the first hidden layer, the second hidden layer and the output layer respectively) with the back-propagation (BP) learning algorithm, is depicted in Fig. 2.5. The parameters of the neural network can be updated in both sequential and batch mode of operation. In BP algorithm, initially the weights and the thresholds are initialized as very small random values. The intermediate and the final outputs of the MLP are calculated by using (2.14), (2.15.), and (2.16.) respectively.

The final output $y_l(n)$ at the output of neuron l, is compared with the desired output $d(n)$ and the resulting error signal $e_l(n)$ is obtained as

$$e_l(n) = d(n) - y_l(n) \tag{2.18}$$

The instantaneous value of the total error energy is obtained by summing all error signals over all neurons in the output layer, that is

$$\xi(n) = \frac{1}{2}\sum_{l=1}^{P_3} e_l^2(n) \tag{2.19}$$

where P_3 is the no. of neurons in the output layer.

This error signal is used to update the weights and thresholds of the hidden layers as well as the output layer. The reflected error components at each of the hidden layers is computed using the errors of the last layer and the connecting weights between the hidden and the last layer and error obtained at this stage is used to update the weights between the input and the hidden layer. The thresholds are also updated in a similar manner as that of the corresponding connecting weights. The weights and the thresholds are updated in an iterative method until the error signal becomes minimum. For measuring the degree of matching, the Mean Square Error (MSE) is taken as a performance measurement.

The updated weights are,

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \tag{2.20}$$

$$w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk}(n) \tag{2.21}$$

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \tag{2.22}$$

where, $\Delta w_{kl}(n), \Delta w_{jk}(n)$ and $\Delta w_{ij}(n)$ are the change in weights of the second hidden layer-to-output layer, first hidden layer-to-second hidden layer and input layer-to-first hidden layer respectively. That is,

$$\begin{aligned} \Delta w_{kl}(n) &= -2\mu \frac{d\xi(n)}{dw_{kl}(n)} = 2\mu e(n) \frac{dy_l(n)}{dw_{kl}(n)} \\ &= 2\mu e(n) \varphi_l' \left[\sum_{k=1}^{P_2} w_{kl} f_k + \alpha_l \right] f_k \end{aligned} \tag{2.23}$$

Where, μ is the convergence coefficient ($0 \le \mu \le 1$). Similarly the $\Delta w_{jk}(n)$ and $\Delta w_{ij}(n)$ can be computed [2.3].

The thresholds of each layer can be updated in a similar manner, i.e.

$$\alpha_l(n+1) = \alpha_l(n) + \Delta\alpha_l(n) \tag{2.24}$$

$$\alpha_k(n+1) = \alpha_k(n) + \Delta\alpha_k(n) \tag{2.25}$$

$$\alpha_j(n+1) = \alpha_j(n) + \Delta\alpha_j(n) \tag{2.26}$$

where, $\Delta\alpha_l(n), \Delta\alpha_k(n)$ and $\Delta\alpha_j(n)$ are the change in thresholds of the output, hidden and input layer respectively. The change in threshold is represented as,

$$\begin{aligned}\Delta\alpha_l(n) &= -2\mu\frac{d\xi(n)}{d\alpha_l(n)} = 2\mu e(n)\frac{dy_l(n)}{d\alpha_l(n)} \\ &= 2\mu e(n)\varphi_l'\left[\sum_{k=1}^{P_2} w_{kl} f_k + \alpha_l\right]\end{aligned} \tag{2.27}$$

2.2.3 Functional-link Artificial Neural Network (FLANN)

Pao originally proposed FLANN and it is a novel single layer ANN structure capable of forming arbitrarily complex decision regions by generating nonlinear decision boundaries [2.7]. Here, the initial representation of a pattern is enhanced by using nonlinear function and thus the pattern dimension space is increased. The functional link acts on an element of a pattern or entire pattern itself by generating a set of linearly independent function and then evaluates these functions with the pattern as the argument. Hence separation of the patterns becomes possible in the enhanced space. The use of FLANN not only increases the learning rate but also has less computational complexity [2.13]. Pao *et al* [2.12] have investigated the learning and generalization characteristics of a random vector FLANN and compared with those attainable with MLP structure trained with back propagation algorithm by taking few functional approximation problems. A FLANN structure with two inputs is shown in Fig. 2.6.

Mathematical Derivation of FLANN

Let **X** is the input vector of size $N\times1$ which represents N number of elements; the n^{th} element is given by:

$$\mathbf{X}(n) = x_n, 1 \le n \le N \tag{2.28}$$

Each element undergoes nonlinear expansion to form M elements such that the resultant matrix has the dimension of $N \times M$.

The functional expansion of the element x_n by power series expansion is carried out using the equation given in (2.29)

$$s_i = \begin{cases} x_n & \text{for } i = 1 \\ x_n^l & \text{for } i = 2, 3, 4, \ldots, M \end{cases} \tag{2.29}$$

where $l = 1, 2, \cdots, M$.

For trigonometric expansion, the

$$s_i = \begin{cases} x_n & \text{for } i = 1 \\ \sin\left(l\pi x_n\right) & \text{for } i = 2, 4, \ldots, M \\ \cos\left(l\pi x_n\right) & \text{for } i = 3, 5, \ldots, M+1 \end{cases} \tag{2.30}$$

Where $l = 1, 2, \cdots, M/2$. In matrix notation the expanded elements of the input vector **E**, is denoted by **S** of size $N \times (M+1)$.

The bias input is unity. So an extra unity value is padded with the **S** matrix and the dimension of the **S** matrix becomes $N \times Q$, where $Q = \left(M + 2\right)$.

Let the weight vector is represented as **W** having Q elements. The output y is given as

$$y = \sum_{i=1}^{Q} s_i \, w_i \tag{2.31}$$

In matrix notation the output can be,

$$\mathbf{Y} = \mathbf{S} \cdot \mathbf{W}^{\mathrm{T}} \tag{2.32}$$

At k^{th} iteration the error signal $e(k)$ can be computed as

$$e(k) = d(k) - y(k) \tag{2.33}$$

Let $\xi(k)$ denotes the cost function at iteration k and is given by

$$\xi(k) = \frac{1}{2}\sum_{j=1}^{P} e_j^2(k) \tag{2.34}$$

where P is the number of nodes at the output layer.

The weight vector can be updated by least mean square (LMS) algorithm, as

$$w(k+1) = w(k) - \frac{\mu}{2}\hat{\nabla}(k) \tag{2.35}$$

where $\hat{\nabla}(k)$ is an instantaneous estimate of the gradient of ξ with respect to the weight vector $w(k)$. Now

$$\begin{aligned}\hat{\nabla}(k) &= \frac{\partial \xi}{\partial w} = -2e(k)\frac{\partial y(k)}{\partial w} = -2e(k)\frac{\partial [w(k)s(k)]}{\partial w} \\ &= -2e(k)s(k)\end{aligned} \tag{2.36}$$

Substituting the values of $\hat{\nabla}(k)$ in (2.35) we get

$$w(k+1) = w(k) + \mu e(k)s(k) \tag{2.37}$$

where μ denotes the step-size $(0 \le \mu \le 1)$, which controls the convergence speed of the LMS algorithm.

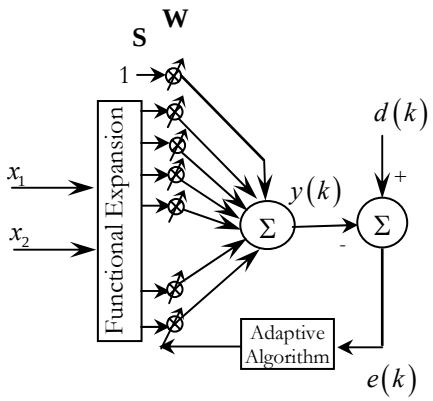


Fig. 2.6 Structure of the FLANN model

2.2.4 Radial Basis Function based NN (RBFNN) Technique

The Radial Basis Function based neural network (RBFNN) consists of an input layer made up of source nodes and a hidden layer of large dimension [2.3-2.5,2.14]. The number of input and output nodes is maintained same and while training the same pattern is simultaneously applied at the input and the output. The nodes within each layer are fully connected to the previous layer as shown in the Fig. 2.7. The input variables are each assigned to a node in the input layer and pass directly to the hidden layer without weights. The hidden nodes contain the radial basis functions (RBFs) which are Gaussian in characteristics. Each hidden unit in the network has two parameters called a center (μ), and a width (σ) associated with it. The Gaussian function of the hidden units is radially symmetric in the input space and the output of each hidden unit depends only on the radial distance between the input vector x and the center parameter μ for the hidden unit.

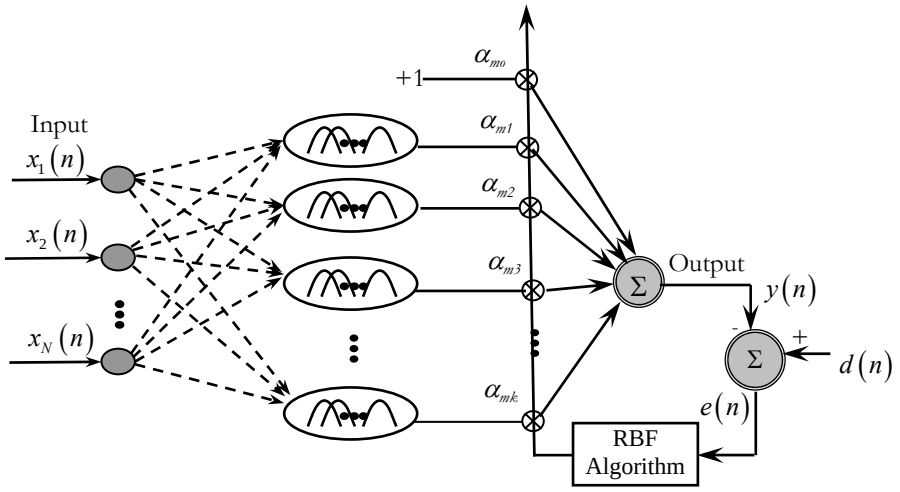


Fig. 2.7 Structure of radial basis function based neural network

The Gaussian function gives the highest output when the incoming variables are closest to the center position and decreases monotonically as the distance from the center decreases. The response of each hidden unit is scaled by its connecting weights (α_{mi}'s) to the output units and then summed to produce the final network output and. The overall network output $y(n)$ at the time index n is therefore

$$y(n) = \alpha_{m0} + \sum_{k=1}^{K} \alpha_{mk} \phi_k (x_j); \qquad j = 1, 2, ..., N \tag{2.38}$$

For each input x_j, N represents the no. of inputs, K = number of hidden units, α_{mk} = connecting weight of the k^{th} hidden unit to output layer, α_{m0} = bias term, m is the number of output.

The value of $\phi_k (x_j)$ is given by

$$\phi_k(x_j) = \exp(-\frac{1}{\sigma_k^2} \left\| x_j - \mu_k \right\|^2) \tag{2.39}$$

Where μ_k is the center vector for the k^{th} hidden unit and σ_k is the width of the Gaussian function and $\| \ \|$ denotes the Euclidean norm.

The parameters of the RBFNN are updated using the RBF algorithm. The RBF algorithm is an exact analytical procedure for evaluation of the first derivative of the output error with respect to network parameters. In the present paper we apply a three layer RBF network with nonlinear output units.

Let the error vector at the n^{th} instant is $e(n) = d(n) - y(n)$, where $d(n)$ = desired output vector and $y(n)$ = estimated output vector. Let $\xi(n) = \frac{1}{2} \sum_P e^2(n)$. The update equations for the center and width of the Gaussian function as well as the connecting and bias weights are derived as

$$\mu_k (n+1) = \mu_k (n) + \Delta \mu_k (n) \tag{2.40}$$

$$\sigma_k (n+1) = \sigma_k (n) + \Delta \sigma_k (n) \tag{2.41}$$

$$\alpha_{mk} (n+1) = \alpha_{mk} (n) + \Delta \alpha_{mk} (n) \tag{2.42}$$

$$\alpha_{m0} (n+1) = \alpha_{m0} (n) + \Delta \alpha_{m0} (n) \tag{2.43}$$

where $\Delta\mu_k(n)$, $\Delta\sigma_k(n)$ are the change of the centers and spread of the Gaussian functions; the $\Delta\alpha_{mk}(n)$ and $\Delta\alpha_{m0}(n)$ are the change in weights and the threshold of the RBFNN. These are computed by taking the partial derivative of $\xi(n)$ with respect to different network parameters. The key equations obtained are stated in (2.44) to (2.47).

$$\Delta\mu_k(n) = -\eta\frac{\partial\xi}{\partial\mu_k} = 2\eta e(n)\phi_k(x_j)\alpha_{mk}\frac{(x_j-\mu_k)}{\sigma_k^2} \tag{2.44}$$

$$\Delta\sigma_k(n) = -\eta\frac{\partial\xi}{\partial\sigma_k} = 2\eta e(n)\phi_k(x_j)\alpha_{mk}\frac{(x_j-\mu_k)^2}{\sigma_k^3} \tag{2.45}$$

$$\Delta\alpha_{mk}(n) = -\eta\frac{\partial\xi}{\partial\alpha_{mk}} = \eta e(n)\phi_k(x_j) \tag{2.46}$$

$$\Delta\alpha_{m0}(n) = -\eta\frac{\partial\xi}{\partial\alpha_{m0}} = \eta e(n) \tag{2.47}$$

where η is the learning rate parameter ($0 \le \eta \le 1$). By applying each input patterns, the change in center location, width of the Gaussian function as well as the connecting weights and bias weights are calculated.

2.3 Summary and Discussion

The various adaptive techniques used in the thesis are presented in this Chapter. The single layer network or perceptron and an adaptive filter using the LMS algorithm are naturally related, as evidenced by their weights updates. However, the perceptron and LMS algorithm differ from each other in some fundamental respects:

- The LMS algorithm uses a linear neuron, whereas the perceptron uses the nonlinear neuron.
- The learning process in the perceptron is performed for a finite number of iterations and then stops. In contrast, continuous learning takes place in the LMS algorithm.

For MLP, the back propagation learning is the standard algorithm. The back-propagation algorithm derives its name from the fact that the partial derivatives of the cost function (performance measure) with respect to the free parameters (synaptic weights and biases) of the network are determined by back-propagating the error signals (computed by output neurons) through the network, layer by layer.

The structure of an RBFNN is unusual in that the constitution of its hidden units is entirely different from that of its output units. Each hidden node consists of a Gaussian function. The learning algorithm of RBFNN is more or less equivalent to LMS algorithm.

REFERENCES

[2.1] B. Widrow and S.D. Sterns, "*Adaptive Signal Processing"* Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1985.

[2.2] S. Haykin, "*Adaptive Filter Theory*", 4th edition, Pearson Education Asia, 2002.

[2.3] S. Haykin, "*Neural Networks: A comprehensive foundation*" 2nd Edition, Pearson Education Asia, 2002.

[2.4] Dayhoff E.J., "*Neural Network Architecture – An Introduction*" Van Norstand Reilold, New York, 1990.

[2.5] Bose N.K., and Liang P., "*Neural Network Fundamentals with Graphs, Algorithms, Applications*", TMH Publishing Company Ltd, 1998.

[2.6] Richard O. Duda, Peter E. Hart and David G. Stork, "*Pattern Classification*", 2nd edition, John Wiley & Sons, INC.2001.

[2.7] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison Wesley, Reading, Massachusetts, 1989.

[2.8] Fu L.M., Hsu H.H., and Principe J.C., "Incremental Back Propagation Learning Networks", *IEEE Trans. on Neural Network,* vol.7, no.3, pp.757-762, 1996.

[2.9] Zhao Q. and Higuchi T., "Evolutionary Learning of Nearest-Neighbor MLP", *IEEE Trans. on Neural Network,* vol.7, no.3, pp.762-768, 1996.

[2.10] Wang G.J., and Chen C.C., "A Fast Multilayer Neural Network Training Algorithm based on the Layer-By-Layer Optimization Procedure", *IEEE Trans. on Neural Network,* vol.7, no.3, pp.768-776, 1996.

[2.11] Carpenter G.A., and Grossberg S., "The Art of Adaptive Pattern Recognition by Self-Organizing Neural Network", *IEEE Computer Mag.* vol.21, no.3, pp.77-88, March 1988.

[2.12] Pao Y.H., Park G.H., and Sobjic D.J., "Learning and Generalization Characteristics of the Random Vector Function", *Neuro Computation,* 6: 163-180, 1994.

[2.13] Patra J.C., and Pal R.N., "A Functional Link Artificial Neural Network for Adaptive Channel Equalization", *Signal Processing 43(1995)* 181-195, vol.43, no.2, May 1995.

[2.14] Jungsik Lee, Beach, C.D., Tepedelenlioglu, N, "Channel equalization using radial basis function network", *IEEE International Conference on Neural Networks*, Vol. 4, Page(s):1924-1928, 3-6 June 1996.

Chapter

3

Direct and Inverse modeling of Capacitive Pressure Sensor using Intelligent Techniques

THE CAPACITIVE PRESSURE SENSOR (CPS), in which the capacitance of a chamber changes with the application of pressure, finds extensive usage because of its low power consumption and high sensitivity as discussed in Chapter-1. The CPS exhibits nonlinearity in its transfer characteristics [3.1-3.3]. This prevents its direct digital readout and restricts its dynamic range. The second problem is the accuracy of measurements in these sensors, which is greatly affected specially by aging of the sensor as well as variations in environmental temperature and humidity. These changes also introduce nonlinearity in the device characteristics, which is usually time varying and unpredictable as it depends on many uncertain factors.

This chapter deals with the design and development of the direct model as well as the inverse model of the CPS. The direct model estimates the CPS; alternatively it develops the electronic model of the CPS. This scheme is very similar to a system identification problem in control systems. This adaptive model of the sensor is developed using different ANN techniques such that the outputs of the sensor and the model will be identical. And the inverse model compensates the nonlinearity present in the sensor. It is analogous to the channel equalization scheme of a digital communication system to cancel the adverse effect of the channel.

Section 3.1 deals with the mathematical modeling of the CPS. In Section 3.2, the direct modeling of the CPS along with the simulation results of all types of ANN modeling has been discussed. The inverse modeling along with the simulation results has been depicted in Section 3.3. Finally, Section 3.4 shows the summary and discussion on different models developed.

3.1 Capacitive Pressure Sensor (CPS)

3.1.1 Model of CPS

A commonly used CPS is shown in Fig. 3.1. Here one plate is a fixed metal disc and the other is a flexible flat circular diaphragm, clamped around its circumference [3.2,3.3]. The dielectric material is air ($\varepsilon \approx 1$). The diaphragm is an elastic sensing element which is bent into a curve by the applied pressure P (Nm^{-2}). The deflection $d(r)$ at any radius r is given by,

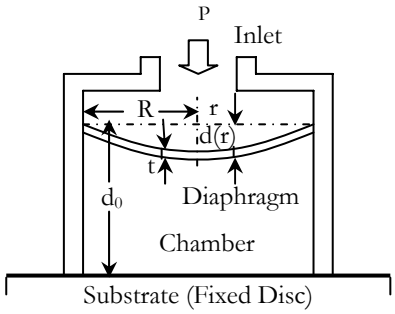


Fig. 3.1 Typical structure of a CPS

$$d(r) = \frac{3}{16}\frac{\left(1-\rho^2\right)}{Et^3}\left(R^2 - r^2\right)^2 P \quad \text{(m)} \tag{3.1}$$

where, R = radius of the circular diaphragm (m), t = thickness of diaphragm (m), E =Young's modulus or elastic modulus (Nm^{-2}), and ρ =Poisson's ratio.

The capacitance of the chamber is,

$$C(P) = \varepsilon_0 \varepsilon_r \int_0^R \frac{2\pi r}{d_0 - d(r)} dr \tag{3.2}$$

where, ε_0 is the permittivity of free space (vacuum), and ε_r is the relative permittivity or dielectric constant of the insulating material.

Putting the value of $d(r)$, obtained in (3.1), into (3.2) and integrating it, the capacitance may be expressed as a function of the fractional pressure x:

$$C(x) = C_0 \frac{1}{\sqrt{x}} \tanh^{-1}\left(\sqrt{x}\right) \tag{3.3}$$

where

$$x = \frac{P}{P_{\max}} \quad (\leq 1) \tag{3.4}$$

$$P_{\max} = \frac{16}{3} \frac{E}{\left(1-\rho^2\right)} \frac{t^3 d_0}{R^4} \tag{3.5}$$

It is clear that $P_{\max}$ is the maximum allowed pressure which causes a center deflection equal to the chamber depth d_0. The offset capacitance (C_0) when $P = 0$ is,

$$C_0 = \frac{\pi R^2 \varepsilon_0 \varepsilon_r}{d_0} \tag{3.6}$$

Expanding (3.3) by Taylor series, we have

$$\begin{aligned} C(x) &= C_0 \sum_{n=0}^{\infty} \frac{x^n}{2n+1} \\ &\cong C_0 \frac{1-(2/3)x}{1-x} + O\left(x^2\right) \end{aligned} \tag{3.7}$$

Where $O\left(x^2\right)$ denotes the residual term of second-order and is considered to be negligible. It is seen that some of the practical sensors use stepped or square diaphragms for better sensitivity. Even though the diaphragm construction is different for the capacitive sensors, their response characteristics are hyperbolic in nature and can be expressed by the relationships provided in (3.3) and (3.7):

$$C(x) = C_0 \frac{1-\alpha x}{1-x} = C_0 + \Delta C(x) \tag{3.8}$$

Where

$$\Delta C(x) = C_0 \frac{1-\alpha}{1-x} x \tag{3.9}$$

The parameters α and $P_{\max}$ of each sensor are determined by the measured capacitances. Error between the measured and approximated values is less than 1%. Therefore, a CPS can be regarded reasonable as a nonlinear capacitor. Plot of (3.8) is shown in Fig. 3.2, and it is observed that the CPS is quite nonlinear in its full range.

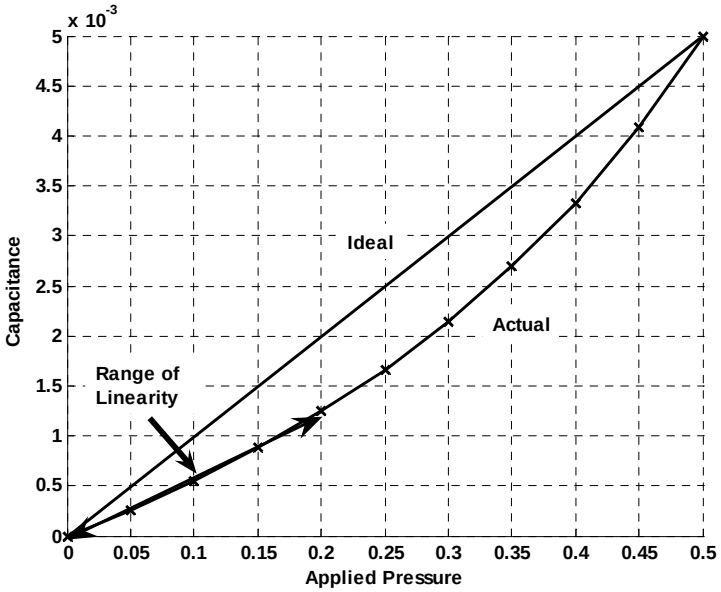


Fig. 3.2 Range of linear region of the CPS

Input-output relation of CPS taking pressure (P) and temperature (T) as inputs:

The input/output equation for the CPS [3.4-3.8] as a function of pressure (*P*) and temperature (*T*) is,

$$C_N = C(P,T)/C_0$$
$$C(P,T) = C_0 g_1(T) + \Delta C(P,T_0) g_2(T) \tag{3.10}$$

Where, C_N is the normalized capacitance. C_0 is the sensor capacitance when $P = 0$ at the reference temperature T_0; When pressure is applied to the CPS, the change in capacitance, $\Delta C(P, T_0) = C_0 \cdot \gamma$, where,

$$\gamma = P_N \frac{1-\alpha}{1-P_N} \tag{3.11}$$

where, α = sensitivity parameter of the CPS which varies with the geometrical structure, $P_N = \text{normalized pressure} = P/P_{\max}$ ($P_{\max}$ is the maximum permissible applied pressure).

Assume that the change in capacitance (or voltage) with respect to temperature is linear. Hence, the linear functions are, $g_i(\cdot)\ (i = 1, 2)$ and, $g_i(T) = 1 + \beta_i \cdot T_N$, where, $T_N = \text{normalized temperature} = \frac{T - T_0}{T_{\max}}$, and β_i is the coefficients which determines the influence of the temperature on the sensor characteristics. Hence, (3.10) can be rewritten as (3.12), and the characteristics of the CPS is at different temperatures is shown in Fig. 3.3.

$$C_N = (1 + \beta_1 \cdot T_N) + P_N \cdot \frac{1-\alpha}{1-P_N}(1 + \beta_2 \cdot T_N) \tag{3.12}$$

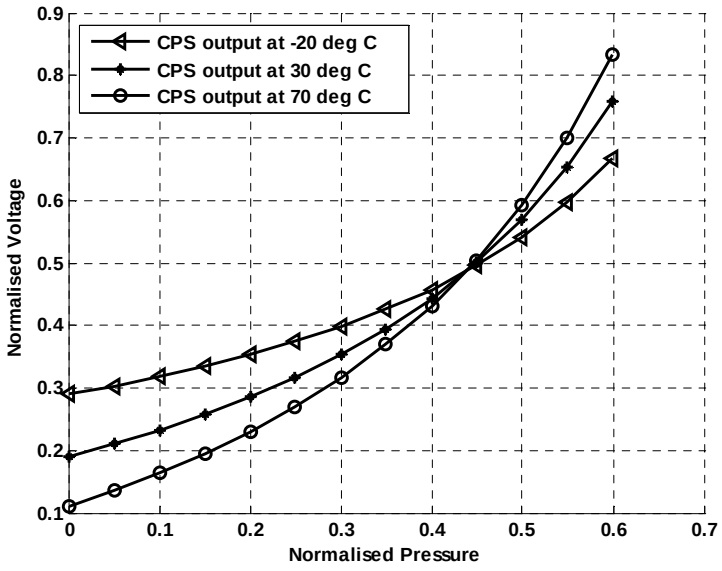


Fig. 3.3 Range of linear region of the CPS at different temperatures

3.1.2 Switched Capacitor Circuit (SCC)

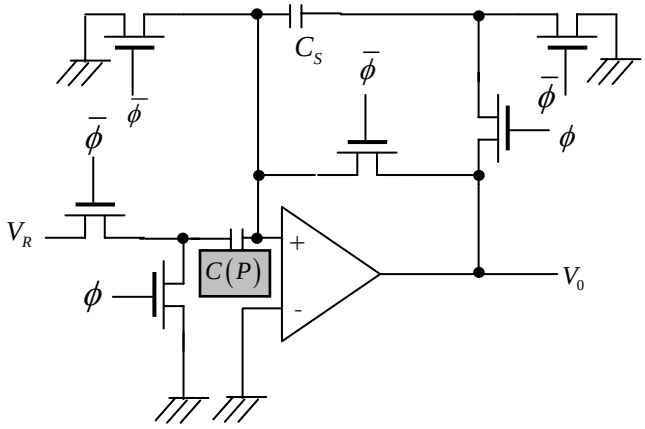


Fig. 3.4 Switched Capacitor Interface circuit

Fig. 3.4 shows a switched capacitor circuit (SCC) for interfacing the CPS [3.5,3.6], where $C(P)$ denotes the CPS. As defined earlier, the SCC output provides an equivalent voltage signal proportional to the capacitance change in the CPS due to applied pressure. The circuit operation can be controlled by a reset signal ϕ. When $\bar{\phi} = 1(+5\text{v})$, $C(P)$ charges to the reference voltage V_R while the capacitor C_S is discharged to ground. And, when $\phi = 1$, the total charge $C(P).V_R$ stored in the $C(P)$ is transferred to C_S and producing an output voltage, $V_0 = K \cdot C(P)$, where $K = -V_R/C_S$. It may be noted if ambient temperature changes, then the SCC output also changes though the applied pressure remains same. By choosing proper values of C_S and V_R, the normalized SCC output can be adjusted in such a way that $V_N = C_N$. The SCC does not take into account the effect of temperature variation in the voltage measurement. The intelligent behavior is implemented in the sensor by training the RBFNN to adapt any temperature change.

3.2 Development of Intelligent Models of the CPS

A scheme of direct modeling and inverse modeling [3.7] of the CPS using different ANN structure has been proposed in this section. The direct modeling is proposed for calibration of inputs and estimation of internal parameters of the CPS. And the inverse modeling is proposed for estimation of applied input pressure.

3.2.1 Direct Modeling

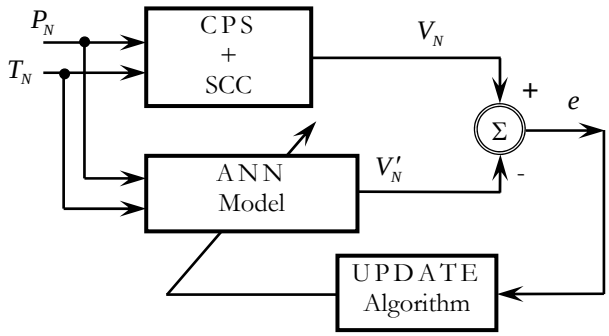


Fig. 3.5 A scheme of direct modeling of a CPS and SCC using ANN based model

The direct modeling is analogous to that of the system identification problem in control system. The purpose of the direct model is to obtain an ANN model of the CPS in such a way that the outputs of the CPS and the ANN match closely. Once a model of the CPS is available, it may be used for fault detection of the sensor. The CPS (interfaced with SCC) output provides an equivalent voltage signal proportional to the capacitance change due to the applied input pressure [3.4, 3.8]. By applying pressure to the CPS, the response is the change in capacitance value with respect to the applied pressure. By connecting the switched capacitor interface circuit (SCC) [3.9, 3.10] with the CPS to get the equivalent voltage signal proportional to the change in capacitance is obtained. Fig. 3.5 represents a scheme of direct model of the CPS using ANN based approach. Since the temperature also affects the output voltage of the sensor, both normalized pressure (P_N) and normalized temperature (T_N) are used as input to the CPS. The output voltage (V_N) of the CPS and that of the ANN model (V'_N) are compared to produce the error signal, e. This error information is used to update the ANN model. The model of CPS has been developed by separately applying all types of neural models such as MLP, FLANN and RBFNN.

3.2.2 Inverse Modeling

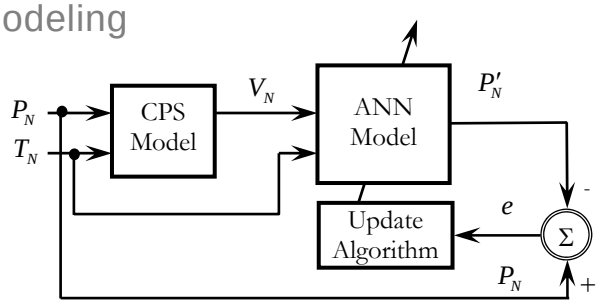


Fig. 3.6 Development of an inverse ANN model of CPS

A scheme of inverse modeling of a CPS using different ANN techniques for estimation of applied pressure is shown in Fig. 3.6. This is analogous to the channel equalization scheme used in a digital communication receiver to cancel the adverse effects of the channel on the data being transmitted. To obtain a direct digital readout of the applied pressure, an inverse model of the CPS may be used in cascade with it to compensate for the adverse effects on the CPS output due to the nonlinear response characteristics and the variations with ambient temperature. The generation of training-set and test-set patterns is similar to that of the direct modeling scheme. However, in the inverse modeling scheme, the normalized temperature T_N and the CPS output V_N are taken as input patterns, and the normalized input pressure P_N is taken as the desired output pattern in the ANN model.

3.3 Simulation Studies

Different ANN technique based direct and inverse models have been simulated extensively in MATLAB 7.0 environment. The SCC output voltage V_N is obtained experimentally at the reference temperature of 25^0C for different values of normalized pressure chosen between 0.1 and 0.7 with an interval of 0.05. Thus, these 13 pairs of input-output data constitute a set of patterns at the reference temperature. The functions $g_1(t)$ and $g_2(t)$ are generated by taking the values of β_1 and β_2 to -2.0×10^{-3} and 7.0×10^{-3}. The geometrical structure of the CPS α is taken as 0.64. From the available CPS pattern set at the reference temperature, i.e., $P_N \sim C_N$, and with the knowledge of functions $g_1(t)$ and $g_2(t)$, eight sets of patterns (each containing 13 pairs of input-output data) are obtained at an interval of 10^0C ranging from -20^0C to 100^0C. Details of each ANN based direct and inverse modeling is described below:

3.3.1 The MLP based Direct Modeling

Simulation studies for MLP network are carried out to obtain a direct model of the CPS. For the simulation study, a two-layer MLP with a 3-5-1 structure is chosen for direct modeling of a CPS as shown in Fig. 2.4 of Chapter 2. (3, 5, and 1 denote the number of nodes including the bias units in the input layer, the first layer, i.e. the hidden layer and the output layer of the ANN, respectively) [3.8]. The hidden layer and the output layer contains the tanh(.) type activation function. The back-propagation (BP) algorithm [3.12], in which

both the learning rate and the momentum rate are chosen as 0.5 and 1 respectively, is used to adapt the weights of the MLP. The normalized temperature (T_N) and the normalized applied pressure (P_N) are used as input pattern, and the SCC output ($V_N = C_N$) is used as the desired pattern to the MLP. After application of each pattern, the ANN weights are updated using the BP algorithm. Completion of all patterns of all the training sets constitutes one iteration of training. To make the learning complete and effective, 10,000 iterations are made to train the ANN. Then, the weights of the MLP are frozen and stored in an EPROM. During the testing phase, the frozen weights are loaded into the MLP model. Then the inputs, P_N and T_N from the test set are fed to this model. The model output is computed and compared with the actual output to verify the effectiveness of the model. The CPS response characteristics of the chosen values at different temperatures $T = -20^0\text{C}, 30^0\text{C}$ and 70^0C are plotted in Fig. 3.7.

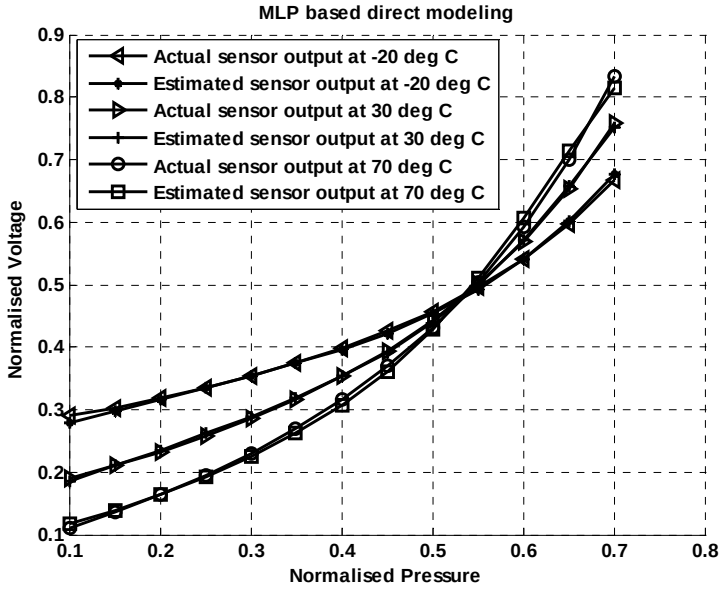
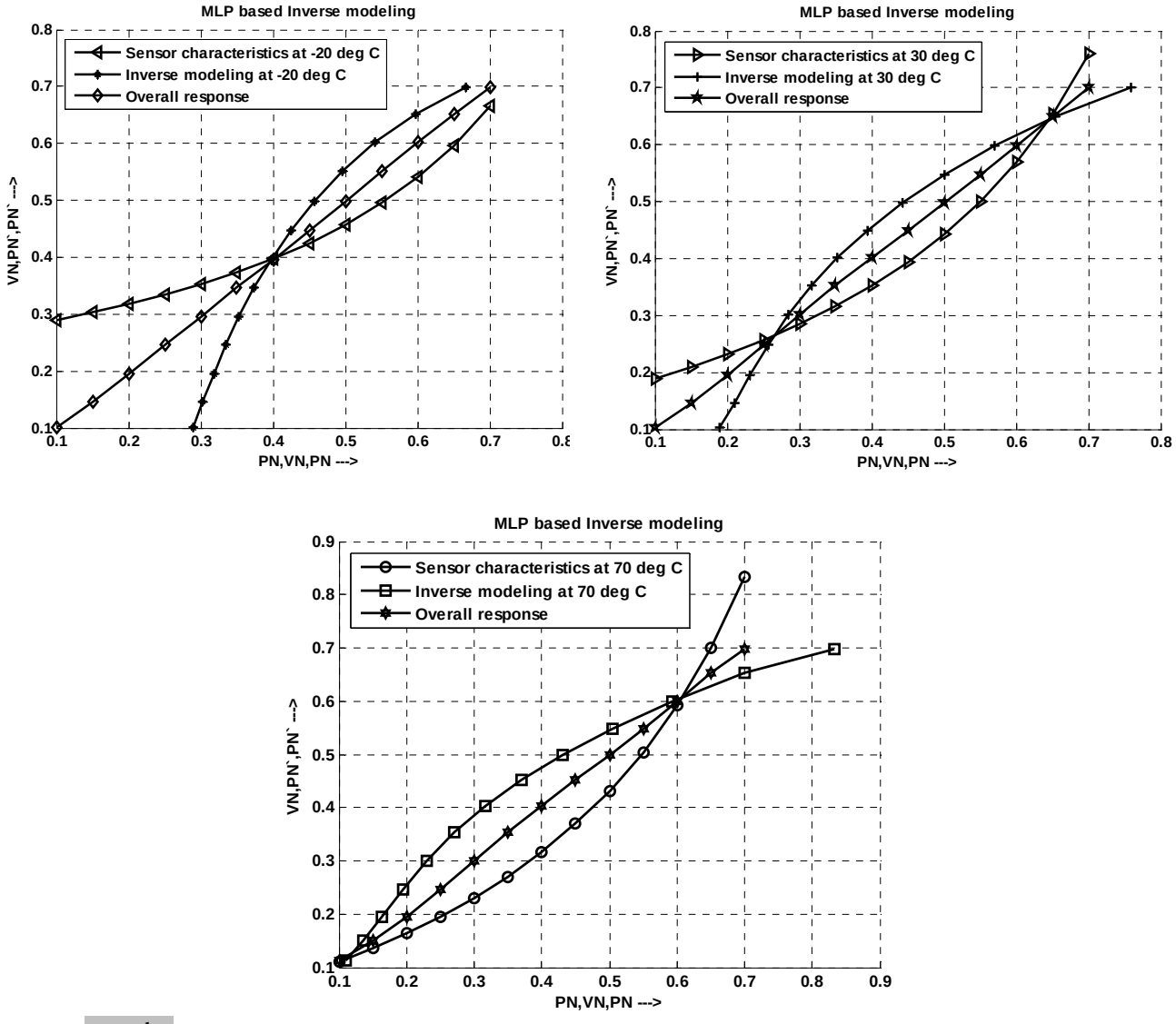


Fig. 3.7 Plots of true and estimated forward characteristics of the CPS at –20^0 C, 30^0 C and 70^0 C by MLP

3.3.2 The MLP based Inverse Modeling

In the inverse modeling the same MLP with a 3-5-1 structure is chosen for simulation [3.8]. The MLP is trained in a similar manner as in the case of direct modeling. In this case also, all the 13 patterns corresponding to temperature values -20^{0}C, 30^{0}C and 70^{0}C are applied during training. The learning rate and the momentum parameter are as same as in the direct modeling i.e. 0.5 and 1 respectively. The MLP is trained for 10,000 iterations using the BP algorithm; the evolved

weights are frozen, and stored in the memory. In the testing phase, the CPS output V_N is applied as input to the MLP network along with the normalized temperature T_N. Then, the estimated pressure P_N' is obtained from the output of the MLP model. The plots of inverse model at different temperatures $T = -20^0C, 30^0C$ and 70^0C are shown in Fig. 3.8.



a b

c

Fig. 3.8 Plots of forward, inverse and overall characteristics of the CPS by MLP; (a) at -20^0C, (b) at 30^0C; and (c) at 70^0C.

Axes Parameters:

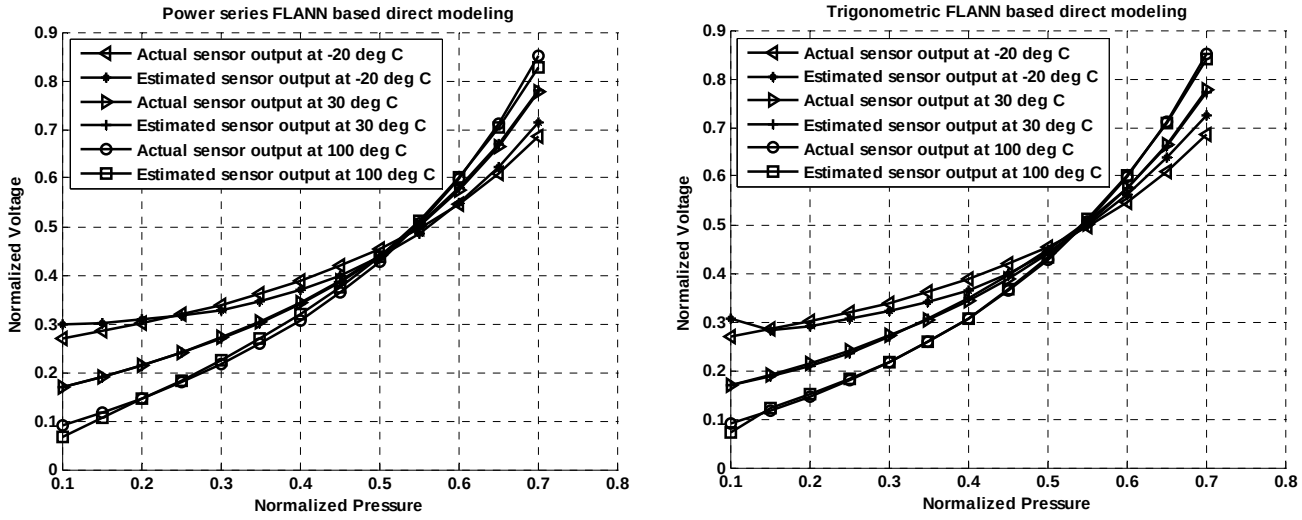
P_N = Normalized pressure

V_N = Normalized voltage (output of the CPS+SCC)

P_N' = Estimated pressure

3.3.3 The FLANN based Direct Modeling

As stated in Chapter 2, the FLANN is a single layered network [3.9,3.10]. Two different experiments are carried out to obtain the direct model of the CPS. In the first case the power series FLANN with 10 number of functional expansions of each input (P_N and T_N) and one product term $P_N \cdot T_N$ is taken. Second is the trigonometric expansion (sin(.) and cos(.) expansions) of input with the product term of $P_N \cdot T_N$ are considered. The learning rate is chosen as 0.035 in both the FLANN update algorithms is used to update the weights of the FLANN model. The normalized temperature (T_N) and the normalized applied pressure (P_N) are used as the input patterns, and the SCC output ($V_N = C_N$) is used as the desired pattern to the FLANN model. 10,000 iterations are used (for both the cases) for training of these FLANN models completely. Then, the weights of the FLANN models are frozen and stored in the memory. During the testing phase the inputs, P_N and T_N are fed to this model. The FLANN model output is computed and compared with the actual output to verify the effectiveness of these proposed models. The results of the modeling of CPS using two FLANN structures are shown in Fig. 3.9 (a) and (b).

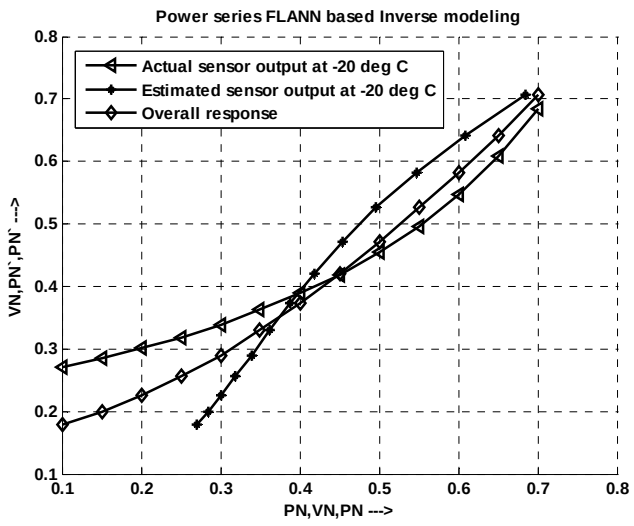


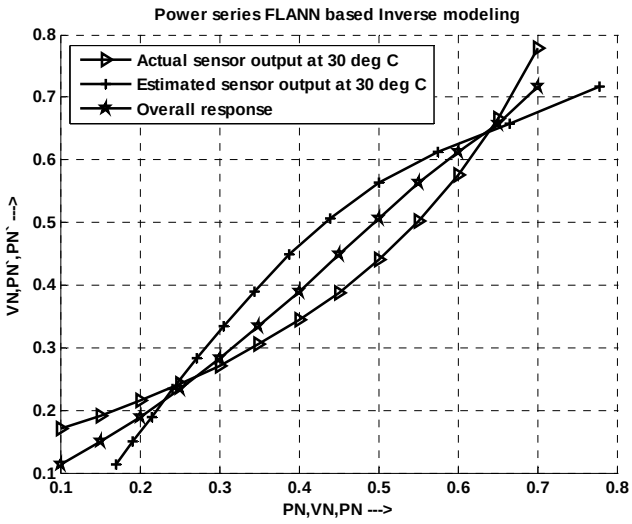
a b

Fig. 3.9 Plots of true and estimated forward characteristics of the CPS at -20^0 C, 30^0 C and 100^0 C
(a) power series FLANN (b) trigonometric series FLANN

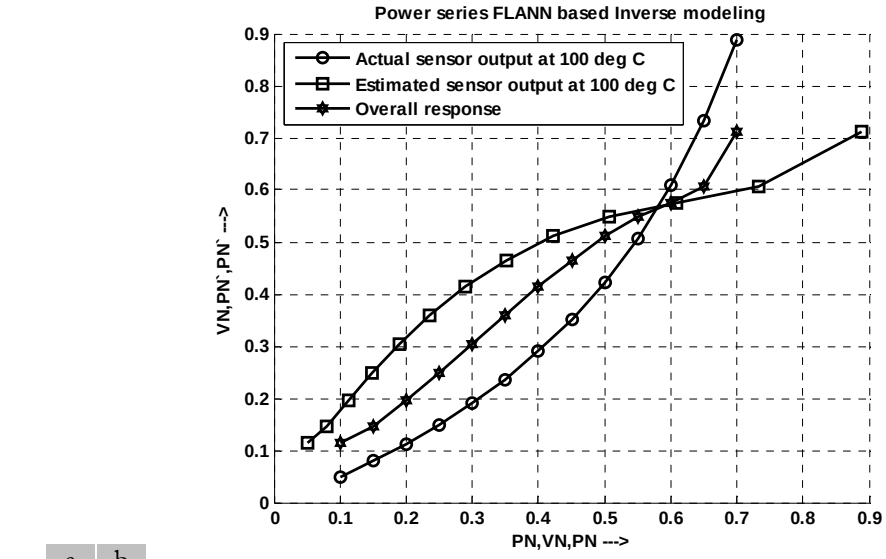
3.3.4 The FLANN based Inverse Modeling

For the inverse modeling of the CPS the power series and the trigonometric FLANNs are successfully trained. In case of the power series FLANN, 12 number of expansions of each input (V_N and T_N) along with the product term of the inputs are computed. The desired signal to the model is the input applied pressure, P_N. To train the network effectively, 50,000 iterations are chosen. The learning rate is set at 0.025. In trigonometric FLANN model, 10 sets of sin(.) expansions and 10 sets of cos(.) expansions with the product term are also obtained. In this case the learning rate is 0.03. Both the FLANN models are trained similar to that of direct modeling. All 13 patterns of inputs corresponding to temperature values of -20^{0}C to 100^{0}C with an interval of 10^{0}C are applied during the training period until the training is complete. Once the ANN model is successfully trained, the weights are frozen for testing. The plots of the inverse model of the power series FLANN and the trigonometric FLANN models at different temperature values of -20^{0}C, 30^{0}C and 100^{0}C are shown in Fig. 3.10 and Fig. 3.11 respectively. From the computed results, it is observed that the trigonometric FLANN model yields improved results compared to the power series counterpart.





Contd…



Page break

a b c

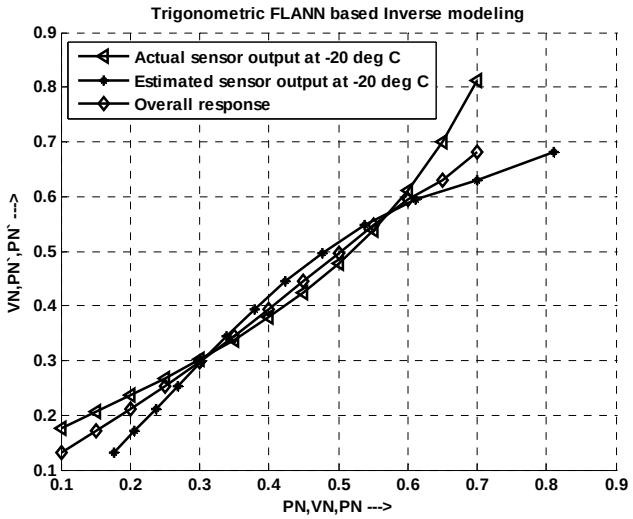
Fig. 3.10 Plots of forward, inverse and overall characteristics of the CPS by power series FLANN; (a) at –20^{0}C, (b) at 30^{0}C; and (c) at 100^{0}C.

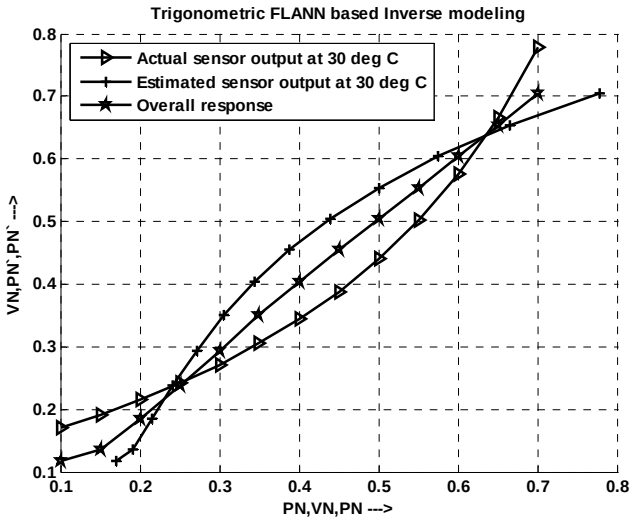
Axes Parameters:

P_N = Normalized pressure

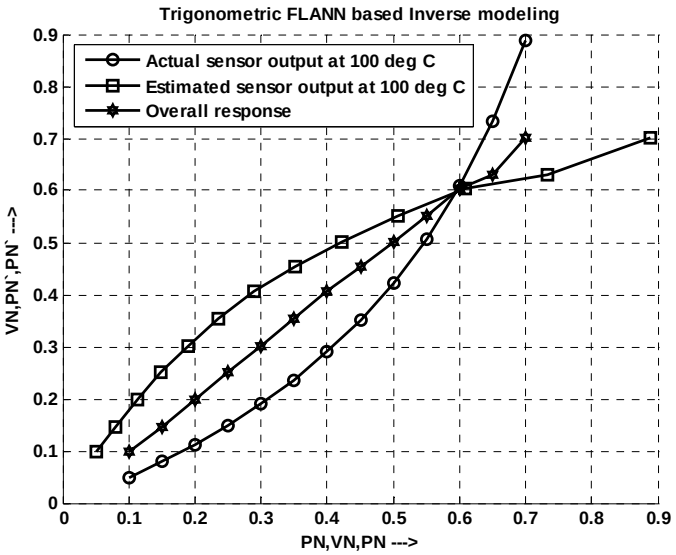
V_N = Normalized voltage (output of the CPS+SCC)

P_N' = Estimated pressure





Contd…



Page break

a b c

Fig. 3.11 Plots of forward, inverse and overall characteristics of the CPS by trigonometric FLANN; (a) at -20^0C, (b) at 30^0C; and (c) at 100^0C.

Axes Parameters:

P_N = Normalized pressure

V_N = Normalized voltage (output of the CPS+SCC)

P_N` = Estimated pressure

3.3.5 The RBFNN based Direct Modeling

Unlike the MLP, the RBFNN is a single layered network [3.12,3.13]. The detailed theory of RBFNN is depicted in Chapter 2. For the simulation, an RBFNN with a 2-6-1 structure is chosen for direct modeling of a CPS. (2, 6, and 1 denote the number of nodes in the input layer, the first layer including the bias units, i.e. the hidden layer and the output layer of the ANN, respectively) [3.14]. The update algorithm, in which both the learning rate is chosen as 0.05, is used to adapt the weights of the RBFNN. The normalized temperature (T_N) and the normalized applied pressure (P_N) are used as input pattern, and the SCC output ($V_N = C_N$) is used as the desired pattern to the RBFNN. In this case the temperature range is -50^0C to 80^0C. After application of all patterns, the ANN weights are updated using the update algorithm. This process is repeated till the mean square error (MSE) is minimized. Once the training is complete,

the RBFNN model will be equivalent to the CPS. To make the training successful, 10,000 iterations are needed. Once the training is complete the weights of the RBFNN can be frozen. During the testing phase, the frozen weights of RBFNN model are used for testing. Then the inputs, P_N and T_N from the test set are fed to this model. The model output is computed and compared with the actual output to verify the effectiveness of the model. The plot of the direct modeling at various temperatures, $T = -50^0 C, 10^0 C$ and $80^0 C$ are shown in Fig. 3.12.

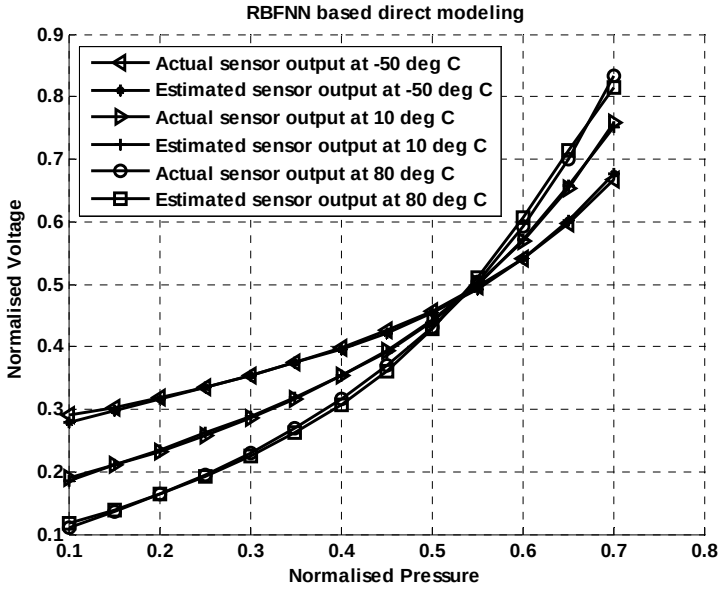
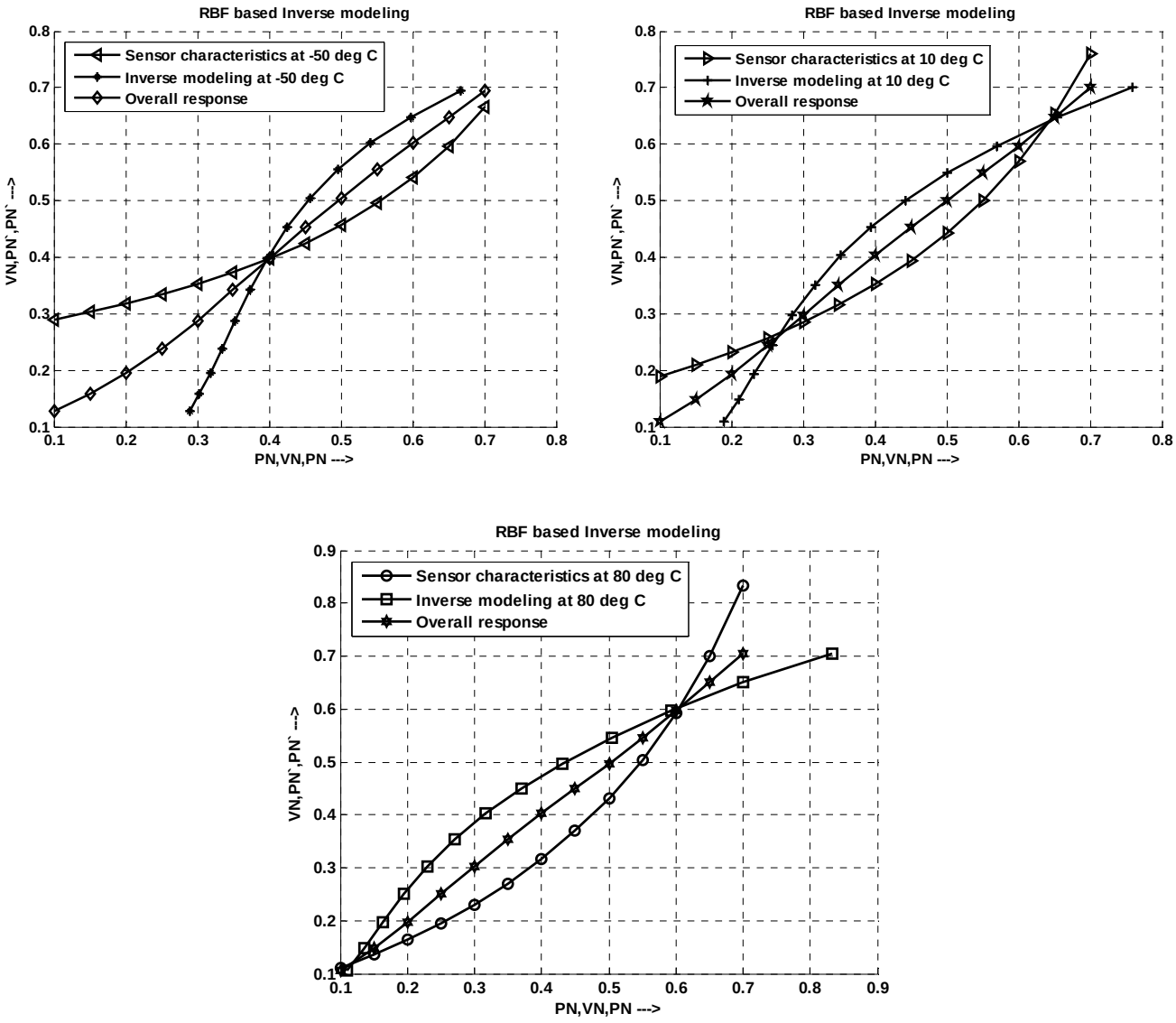


Fig. 3.12 Plots of true and estimated forward characteristics of the CPS at –50^0 C, 10^0 C and 80^0 C by RBFNN

3.3.6 The RBFNN based Inverse Modeling

For inverse model, same 2-6-1 structured RBFNN has been chosen for simulation. Each hidden node contains a radial basis type function [3.14]. The output of the CPS (V_N) and the normalized temperature (T_N) are the inputs to the RBFNN based inverse model. The normalized input pressure is then compared with the output of the inverse model P_N' and produces error signal. Then the weights of the network are trained by the update algorithm to match both the outputs. In the updation process the learning rate is chosen as 0.001. And 50,000 iterations are needed to successfully train the network. The plots of inverse model at different temperatures $T = -50^0 C, 10^0 C$ and $80^0 C$ are shown in Fig. 3.13.



a b

c

Fig. 3.13 Plots of forward, inverse and overall characteristics of the CPS by RBFNN; (a) at -50^0C, (b) at 10^0C; and (c) at 80^0C.

Axes Parameters:

P_N = Normalized pressure

V_N = Normalized voltage (output of the CPS+SCC)

P_N` = Estimated pressure

3.4 Summary and Discussion

This chapter presents the development of direct modeling and the inverse modeling of the CPS with various ANN techniques. The direct modeling is proposed to estimate the nonlinearity present in a CPS. It is more or less equivalent to the system identification problem in control systems. The direct model is also used for fault detection of the sensor. Various ANN (MLP, FLANN and RBFNN) based direct models are proposed in this chapter.

Inverse models are developed to compensate the nonlinearity present in a CPS. The inverse modeling is analogous to the channel equalization problem in communication system. The inverse model is itself a nonlinear system; however its characteristics is the inverse of the sensor characteristics. It is connected in cascade with the sensor to achieve the overall linearity over a wide range. It has been developed by using different ANN techniques. It is observed that the linearity is high in case of MLP and RBFNN. However, if we increase the number of expansions in FLANN, the percentage of linearity can be high. The FLANN network involves less computational complexity but provides satisfactory performance. Therefore FLANN based models are easily implementable and may be preferred for real-time applications.

REFERENCES

[3.1] Hermann K.P. Neubert, *"Instrument Transducers: An Introduction to their performance & design"*, 2nd edition, Oxford University Press.

[3.2] John P. Bentley, *"Principles of Measurement Systems"*, 3rd edition, Pearson Education.

[3.3] G.C. Meijer, "Concepts and focus points for intelligent sensor systems", *Sensor and Actuators: A Physical*, 41-42 (1994), 183-191.

[3.4] D.D. Bruyker, A. Cozma and Robert Puers, "A combined piezoresistive / capacitive pressure sensor with self-test function on thermal actuation", *Proc. of IEEE conference on solid state sensors & actuators*, 16th-19th June, 1997, Chicago.

[3.5] H. Matsumoto, H. Shimizu, and K. Watanabe, "A switched capacitor charge balancing analog-to-digital converters and its application to capacitor measurement", *IEEE Trans. Instrumentation & Measurement*, vol. IM-36, pp. 873-878, Dec. 1987.

[3.6] M. Yamada, T. Takebayashi, S.I. Notoyama, and K. Watanabe, "A switched capacitor interface for capacitive pressure sensors", *IEEE Trans. Instrumentation & Measurement*, vol. 41, pp. 81-86, Feb. 1992.

[3.7] M. Yamada and K. Watanabe, "A capacitive pressure sensor interface using oversampling Δ-Σ demodulation techniques", *IEEE Trans. Instrumentation & Measurement*, vol. 46, pp. 3-7, Feb. 1997.

[3.8] J.C. Patra, A.C. Kot and G. Panda, "An intelligent pressure sensor using neural networks", *IEEE Trans. Instrumentation & Measurement*, vol. 49, No. 4, Aug.2000.

[3.9] J.C. Patra, G. Panda, and R. Baliarsingh, "Artificial neural network-based nonlinearity estimation of pressure sensors", *IEEE Trans. Instrumentation & Measurement*, vol. 43, No. 6, Dec. 1994.

[3.10] S.K. Rath, J.C. Patra, and A.C. Kot, "An intelligent pressure sensor with self-calibration capability using artificial neural network", *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, pp.2563 – 2568, 8th-11th, Oct. 2000.

[3.11] B. Widrow, S.D. Stearns, "*Adaptive Signal Processing*", 2nd edition, Pearson Education Asia.

[3.12] S. Haykin, "*Neural Networks-A comprehensive foundation*", Second edition, Pearson Education Asia, 2002.

[3.13] Y.H. Pao, "*Adaptive Pattern Recognition and Neural Networks*", Reading, MA: Addison-Wesley, 1989.

[3.14] G. Panda, D.P. Das, **S.K. Mishra**, S.K. Panda, S. Meher and K.K. Mahapatra, "Development of an Efficient Intelligent Pressure Sensor using RBF based ANN", *Proceedings of International Conference on Intelligent Signal Processing & Robotics (ISPR-2004),* (ITS2 in CD), 20th-23rd Feb, 2004, IIIT, Allahabad.

Chapter

4

Performance Evaluation of Fixed-Point FLANN based CPS model

THIS CHAPTER deals with the fixed-point analysis of the FLANN based direct and inverse model of the CPS. To assess the performance of the direct and the inverse model in actual implementation a fixed-point analysis is required. In the previous Chapter it is proposed that the FLANN model involves less computational complexity offering almost similar performance as that of other types of ANN [4.1-4.2]. Due to involvement of less computational complexity and better performance, the power series FLANN is considered for implementation. Computer simulation studies have been carried out to demonstrate the performance of fixed-point FLANN structures of both the direct and the inverse models of the CPS. This is primarily contributed by the word lengths used in the implementation. The present study has been made to assist to choose the desired word length for implementation.

In Section 4.1, the fixed-point analysis of the CPS for both direct and inverse modelings has been carried out. Section 4.2 deals with the simulation studies of fixed-point implementation. The effect of word lengths on the performance of the CPS is demonstrated in Section 4.3.

4.1 Fixed-point Analysis

The input-output relation of the CPS as a function of pressure (P_N) and temperature (T_N) is derived in (3.12) of Chapter-3. In this fixed-point analysis, the inputs P_N and T_N are represented into $(P_N)_F$ and $(T_N)_F$ with *t*-bits. We denote these variables as,

$$(P_N)_F = scale(P_N, t) \quad \text{and} \quad (T_N)_F = scale(T_N, t) \tag{4.1}$$

where,

$scale(x,t)$ is a function, which represents the input x in *t*-bits.

Then (3.12) (of Chapter-3) can be written as,

$$(V_N)_F = add\left\{ f_1(T), mul(\gamma, f_2(T), t), t\right\} \tag{4.2}$$

where,

$add(x,y,t)$ is the function that calculates fixed-point addition of the inputs x and y, with *t*-bits.

Similarly, $mul(x,y,t)$ is the function that performs the fixed-point multiplication of inputs x and y, using finite length of *t*-bit.

4.1.1 Fixed-point FLANN Algorithm for Direct Modeling

Fig. 4.1 shows a direct modeling scheme of a CPS, where $(P_N)_F$ and $(T_N)_F$ stands for fixed-point representation of P_N and T_N respectively. Similarly $(V_N)_F$ and $(V_N')_F$ represent the fixed-point outputs of a CPS and fixed-point FLANN model. In fixed-point mode the error signal is

also represented by $(e)_F$. The magnitude of the fixed-point variables depends on the word length and the type of arithmetic used to represent the numbers.

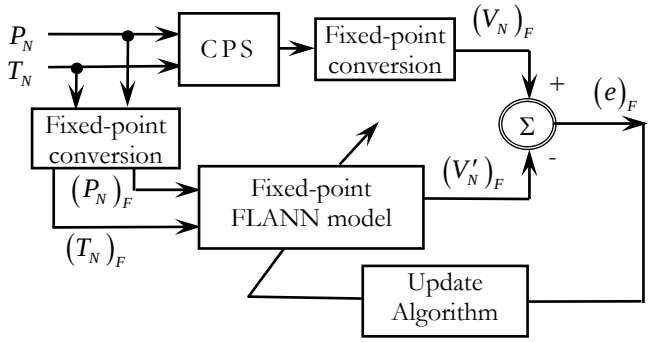


Fig. 4.1 A scheme of direct modeling of a CPS using fixed-point FLANN model

Algorithm

The output of the fixed-point FLANN model V_N' is calculated by expanding the input elements nonlinearly into power series expansion as,

$$\begin{aligned} V_N{}' = {} & w_1 \cdot P_N + w_2 \cdot P_N^2 + w_3 \cdot P_N^3 + w_4 \cdot P_N^4 + w_5 \cdot P_N^5 + \\ & w_6 \cdot T_N + w_7 \cdot T_N^2 + w_8 \cdot T_N^3 + w_9 \cdot T_N^4 + w_{10} \cdot T_N^5 \end{aligned} \tag{4.3}$$

Using fixed-point operations (4.3) may be expressed as,

$$\left(V_N{}'\right)_F = \left(\begin{array}{l} \left(w_1 \cdot (P_N)_F\right)_F + \left(w_2 \cdot \left(P_N^2\right)_F\right)_F + \cdots \\ \qquad + \left(w_5 \cdot \left(P_N^5\right)_F\right)_F + \left(w_6 \cdot (T_N)_F\right)_F + \\ \qquad \left(w_7 \cdot \left(T_N^2\right)_F\right)_F + \cdots + \left(w_{10} \cdot \left(T_N^5\right)_F\right)_F \end{array} \right)_F \tag{4.4}$$

Thus (4.4) can be obtained by summing up the partial fixed-products.

The power terms of inputs can be computed in fixed-point mode as,

$$\begin{aligned} \left(P_N^{i+1}\right)_F &= mul\left(\left(P_N^i\right)_F, (P_N)_F, t\right) \\ \left(T_N^{i+1}\right)_F &= mul\left(\left(T_N^i\right)_F, (T_N)_F, t\right) \end{aligned}, \quad \text{for } i = 0,1,2,3,4. \tag{4.5}$$

These power terms are multiplied in fixed-point arithmetic with the respective weights to give fixed point results as,

$$\left.\begin{aligned}
x_1 &= mul\left(w_1,\left(P_N\right)_F,t\right)\\
x_2 &= mul\left(w_2,\left(P_N^2\right)_F,t\right)\\
x_3 &= mul\left(w_3,\left(P_N^3\right)_F,t\right)\\
x_4 &= mul\left(w_4,\left(P_N^4\right)_F,t\right)\\
x_5 &= mul\left(w_5,\left(P_N^5\right)_F,t\right)\\
x_6 &= mul\left(w_6,\left(T_N\right)_F,t\right)\\
x_7 &= mul\left(w_7,\left(T_N^2\right)_F,t\right)\\
x_8 &= mul\left(w_8,\left(T_N^3\right)_F,t\right)\\
x_9 &= mul\left(w_9,\left(T_N^4\right)_F,t\right)
\end{aligned}\right\} \quad (4.6)$$

Similarly the fixed-point additions are performed on these partial products as,

$$\begin{aligned}
y_1 &= add\left(x_1,x_2,t\right)\\
y_{i+1} &= add\left(y_i,x_{i+2},t\right), \quad \text{for } i = 0,1,2,\ldots,8.
\end{aligned} \quad (4.7)$$

Finally, the overall fixed-point output of the FLANN is given by,

$$\left(V_N{}'\right)_F = y_9 \quad (4.8)$$

The FLANN algorithm is used in fixed-point mode to update the weights [4.5,4.6]. It can be expressed as,

$$\left(w_i\left(n+1\right)\right)_F = \left(w_i\left(n\right)\right)_F + \left(2\lambda e\left(n\right)x\left(n\right)\right)_F \quad (4.9)$$

The second term is,

$$\left(\Delta w(n)\right)_F = mul\left(2,\left(mul\left(\left(mul\left(\lambda,\left(e(n)\right)_F,t\right)\right),\left(x(n)\right)_F,t\right)\right),t\right) \tag{4.10}$$

$$\left(w_i(n+1)\right)_F = add\left(\left(w_i(n)\right)_F,\left(\Delta w(n)\right)_F,t\right) \tag{4.11}$$

where $[x(n)]_F$ is the fixed-point representation of the FLANN input vectors computed as before,

$$\left[x(n)\right]_F = \left[\left(P_N\right)_F \quad \left(P_N^2\right)_F \quad \cdots \quad \left(P_N^5\right)_F \quad \left(T_N\right)_F \quad \left(T_N^2\right)_F \quad \cdots \quad \left(T_N^5\right)_F\right] \tag{4.12}$$

The details of fixed-point FLANN based direct model is depicted in the Fig. 4.2.

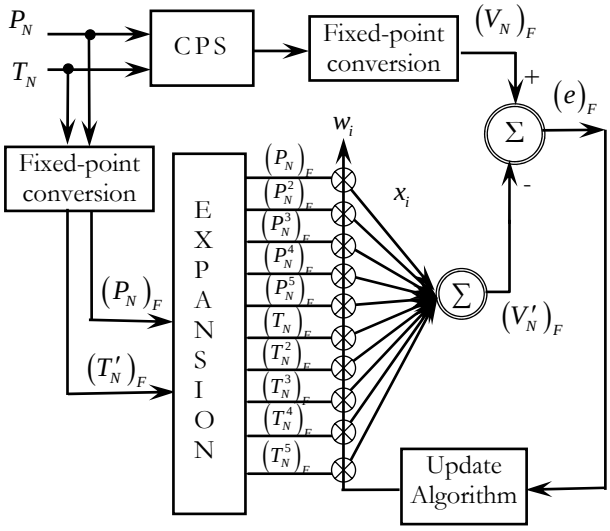


Fig. 4.2 Direct modeling of CPS using fixed-point FLANN used for simulation study

4.1.2 Fixed-point FLANN Algorithm for Inverse Modeling

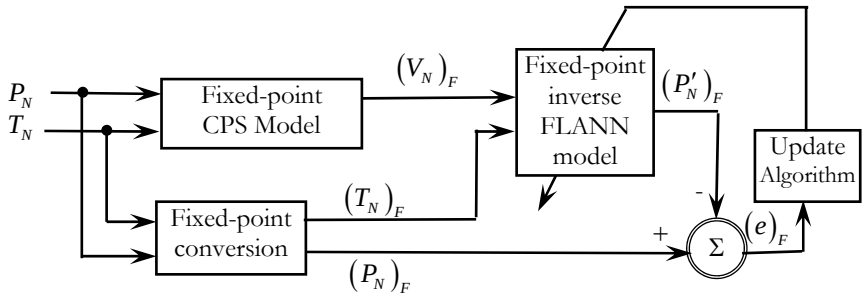


Fig. 4.3 Fixed-point inverse modeling of CPS by FLANN

Algorithm

Referring to Fig. 4.3 and 4.4, the estimated output from the fixed-point power series FLANN model of 5th order expansion can be computed as,

$$\left(P_N{}'\right)_F = \left(\begin{array}{c} \left(W_1 \cdot \left(V_N\right)_F\right)_F + \left(W_2 \cdot \left(V_N^2\right)_F\right)_F + \cdots + \left(W_5 \cdot \left(V_N^5\right)_F\right)_F \\ \left(W_6 \cdot \left(T_N\right)_F\right)_F + \left(W_7 \cdot \left(T_N^2\right)_F\right)_F + \cdots + \left(W_{10} \cdot \left(T_N^5\right)_F\right)_F \end{array} \right)_F \tag{4.13}$$

where $\left(V_N\right)_F$ is the *t*-bit representation of output of the CPS, and $\left(T_N\right)_F$ is the *t*-bit representation of normalized temperature.

The other power terms of inputs to the FLANN model can be calculated similar to the direct modeling and may be represented by:

$$\begin{array}{l} \left(V_N^{i+1}\right)_F = mul\left(\left(V_N^i\right)_F, \left(V_N\right)_F, t\right) \\ \left(T_N^{i+1}\right)_F = mul\left(\left(T_N^i\right)_F, \left(T_N\right), t\right) \end{array}, \text{ for } i = 0,1,2,3,4. \tag{4.14}$$

The partial products of the FLANN model is termed as,

$$\left.\begin{array}{l} X_1 = mul\left(W_1, \left(V_N\right)_F, t\right) \\ X_2 = mul\left(W_2, \left(V_N^2\right)_F, t\right) \\ X_3 = mul\left(W_3, \left(V_N^3\right)_F, t\right) \\ X_4 = mul\left(W_4, \left(V_N^4\right)_F, t\right) \\ X_5 = mul\left(W_5, \left(V_N^5\right)_F, t\right) \\ X_6 = mul\left(W_6, \left(T_N\right)_F, t\right) \\ X_7 = mul\left(W_7, \left(T_N^2\right)_F, t\right) \\ X_8 = mul\left(W_8, \left(T_N^3\right)_F, t\right) \\ X_9 = mul\left(W_9, \left(T_N^4\right)_F, t\right) \end{array}\right\} \tag{4.15}$$

$$X_{10} = mul\left(W_{10}, \left(T_N^5\right)_F, t\right)$$

Now by adding all these fixed-point products in *t*-bits each gives rise to,

$$\begin{aligned} Y_1 &= add\left(X_1, X_2, t\right) \\ Y_{i+1} &= add\left(Y_i, X_{i+2}, t\right), \text{ for } i = 1, 2, \ldots 8. \end{aligned} \tag{4.16}$$

Hence the overall output of the fixed-point FLANN model is given by,

$$\left(P_N{}'\right)_F = Y_9 \tag{4.17}$$

The FLANN algorithm used to update the weights [4.3] in fixed-point mode is given by,

$$\left(W_i\left(n+1\right)\right)_F = \left(W_i\left(n\right)\right)_F + \left(2\lambda e\left(n\right)X\left(n\right)\right)_F \tag{4.18}$$

The second term is,

$$\left(\Delta W\left(n\right)\right)_F = mul\left(2, \left(mul\left(\left(mul\left(\lambda, \left(e\left(n\right)\right)_F, t\right)\right), \left(X\left(n\right)\right)_F, t\right)\right), t\right) \tag{4.19}$$

Thus, (4.22) can be obtained as,

$$\left(W_i\left(n+1\right)\right)_F = add\left(\left(W_i\left(n\right)\right)_F, \left(\Delta W\left(n\right)\right)_F, t\right) \tag{4.20}$$

where, $\left[X\left(n\right)\right]_F$ is the fixed-point representation of FLANN input vectors computed as before

$$\left[X\left(n\right)\right]_F = \left[\left(V_N\right)_F \quad \left(V_N^2\right)_F \cdots \left(V_N^5\right)_F \quad \left(T_N\right)_F \quad \left(T_N^2\right)_F \cdots \left(T_N^5\right)_F\right] \tag{4.21}$$

The fixed-point FLANN based inverse model for simulation is depicted in the Fig. 4.4.

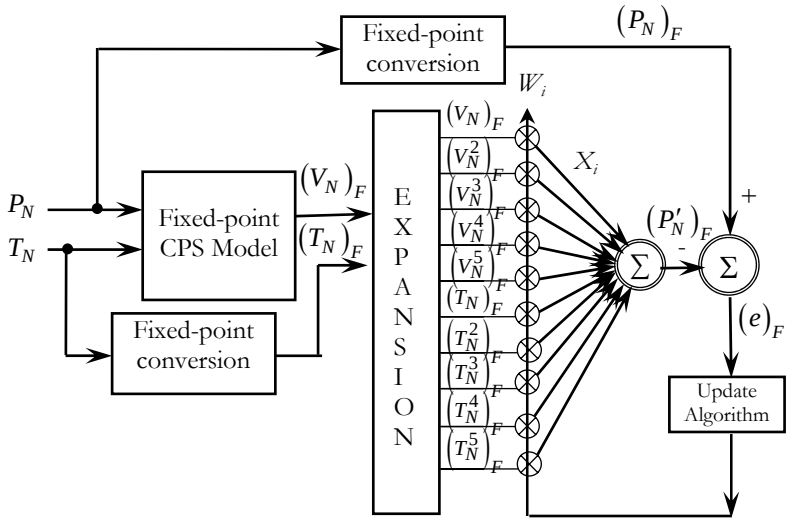


Fig. 4.4 Inverse modeling of CPS by fixed-point FLANN used for simulation study

4.2 Simulation Studies

Both the fixed-point direct and inverse models shown in Figs. 4.3 and 4.4 of the CPS are simulated in this section. The actual CPS model, which is a function of both pressure and temperature, is assumed to have the nonlinear form as in (3.12). The direct modeling block is simulated for the following conditions,

P_N = 0.0 to 0.8 at a step of 0.1,

T = 30^{0}C,

a = 0.7,

β_1 = -2.0×10^{-3} and β_2 = 7.0×10^{-3}

The CPS and the SCC blocks are mathematically represented by (4.2). The output of this sensor is given by V_N. The same input is applied to the FLANN model and the corresponding output is calculated by (4.4). Both infinite precision as well as finite precision FLANN models are simulated. In the fixed-point FLANN model, the word length t is varied between 4-bit to 16-bit

at an interval of 4-bit. The output of the FLANN model is represented as V_N'. The two outputs are compared and the error is used to update the parameter of the FLANN model. After some iterations of training the minimum mean square error attains a steady state value, at that point of time the training is stopped and the FLANN parameters are frozen. Now fresh normalized pressure is applied and the corresponding outputs of the CPS, both floating-point FLANN and fixed-point FLANN models are obtained. The plots of fixed-point FLANN model outputs are shown in Fig. 4.5. The plots clearly indicate that at low bit length (e.g. t = 4 and 8-bit) the match between true and estimated response is observed to be poorer. The agreement improves when the bit length is increased to 12 and it becomes excellent when the bit length is 16. Thus the bit length substantially affects the response of the direct model.

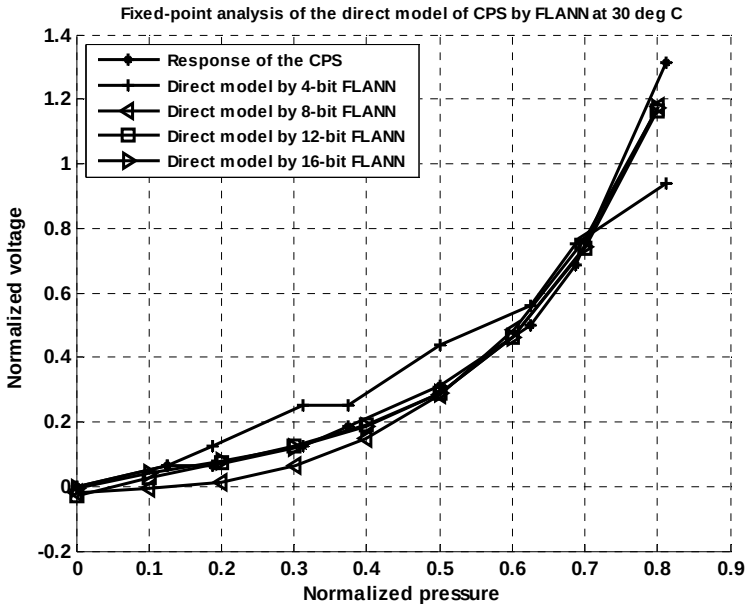
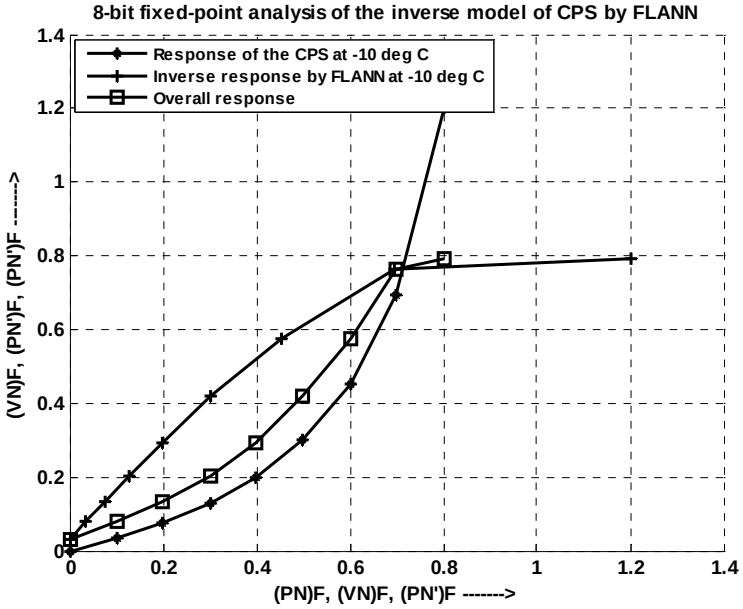
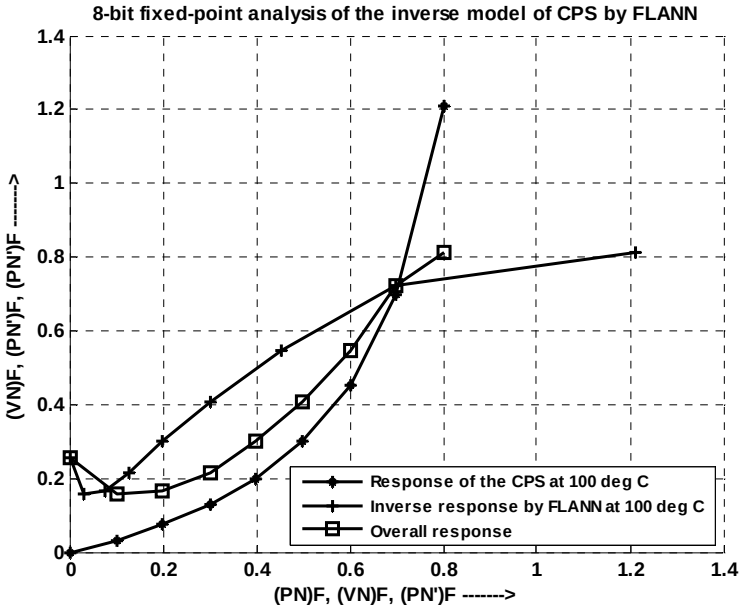


Fig. 4.5 Simulation results of fixed-point FLANN direct model at 30^0C, of the word lengths t = 4, 8, 12 and 16-bits.

The plot of the fixed-point FLANN based inverse model is shown in Fig. 4.4. In this case the inverse model is connected in series and is of fixed-point type. The same parameters, as used for direct model, are used for this simulation also. The results of simulations (direct, inverse and resultant) at different bit lengths and at different temperatures are shown in Figs. 4.6 through Fig. 4.8 [4.4,4.5]. These plots clearly show that at low bit length, the resultant plot between $\left(P_N\right)_F$ and $\left(P_N'\right)_F$ are not quite linear. This means that at low bit length (e.g. t = 8-bit) the linearization offered by fixed-point FLANN model is poor. Fig.

4.6 shows the performance of the inverse model corresponding 8-bit representation of data.





a

b

Fig. 4.6 Plots of forward, inverse and overall characteristics of the CPS by fixed-point FLANN at t = 8-bits; (a) at -10^0C, (b) at 100^0C.

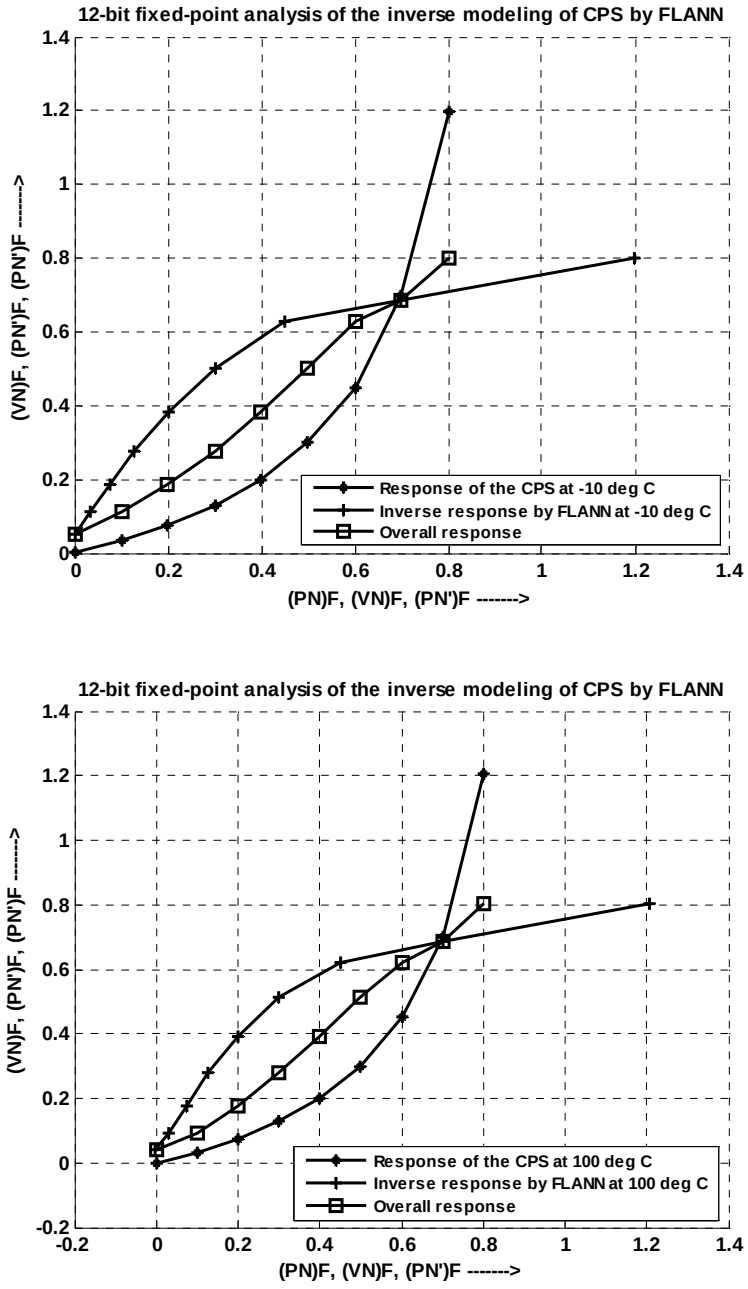
Axes Parameters:

(PN)F = fixed-point normalized pressure

(VN)F = fixed-point normalized voltage (output of the CPS+SCC)

(PN')F = fixed-point estimated pressure

When the bit length is increased to 12 the linearization behaviour improves. This is shown in Fig. 4.7 (a) and (b).



a

b

Fig. 4.7 Plots of forward, inverse and overall characteristics of the CPS by fixed-point FLANN at t = 12-bits; (a) at -10^0C, (b) at 100^0C.

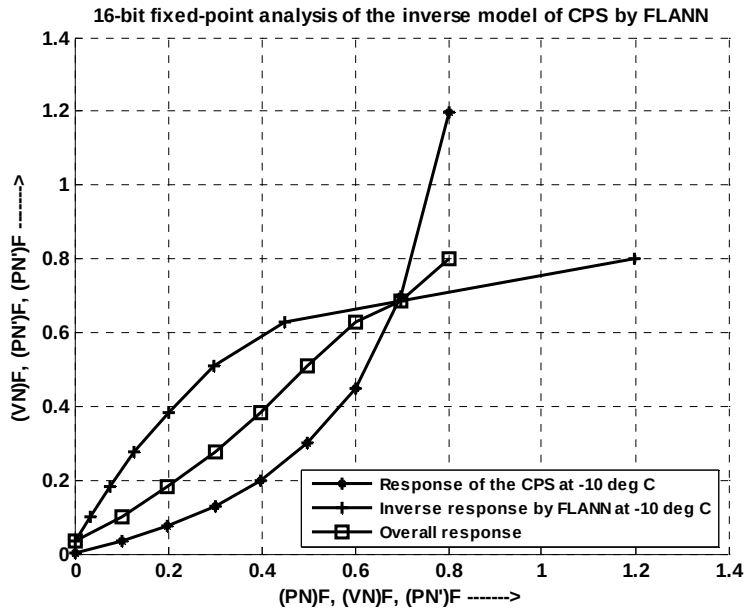
Axes Parameters:

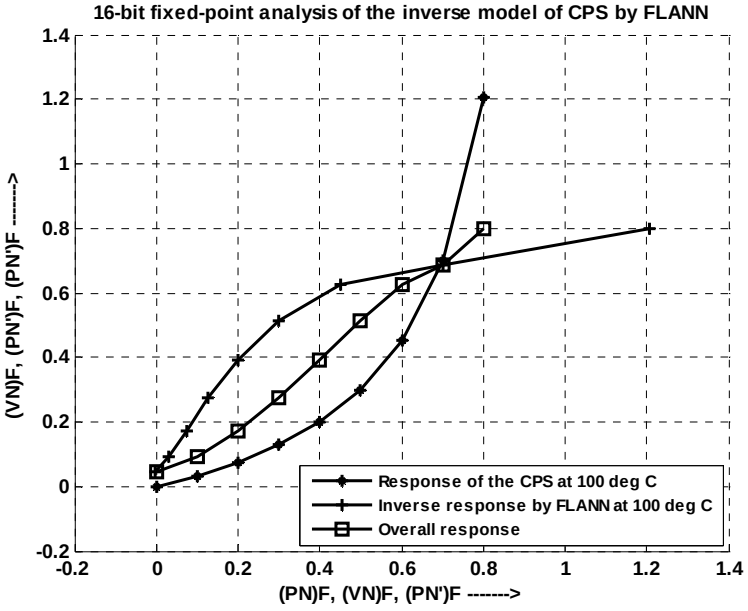
(PN)F = fixed-point normalized pressure

(VN)F = fixed-point normalized voltage (output of the CPS+SCC)

(PN')F = fixed-point estimated pressure

At 16-bit word length the fixed-point effect is greatly reduced which is evident from Fig. 4.8.





a

b

Fig. 4.8 Plots of forward, inverse and overall characteristics of the CPS by fixed-point FLANN at t = 16-bits; (a) at -10^0C, (b) at 100^0C.

Axes Parameters:

(PN)F = fixed-point normalized pressure

(VN)F = fixed-point normalized voltage (output of the CPS+SCC)

(PN')F = fixed-point estimated pressure

4.3 Summary and Discussion

In floating-point implementation the performance of the direct and inverse models of various sensors is quite good. However, it is shown that the word length greatly affects the linearity performance of these sensors. Thus to achieve an appreciation performance, judicious choosing of word length is very helpful. The present Chapter provides in-depth information on this burning issue.

REFERENCES

[4.1] J.C. Patra, Alex C. Kot, and G. Panda, "An Intelligent Pressure Sensor Using Neural Networks," *IEEE Trans. on Instrumentation & Measurement*, vol.49, No.4, pp.829-834, Aug.2000.

[4.2] J.C. Patra and A. van den Bos, "Modeling of an intelligent pressure sensor using functional link artificial neural networks", *ISA Transactions*, 39 (2000), pp. 15-27.

[4.3] B. Widrow, S.D. Stearns, "*Adaptive Signal Processing*", 2nd edition, Pearson Education Asia.

[4.4] **S.K. Mishra**, G. Panda, D.P. Das, S.K. Pattanaik and M.R. Meher, "Performance Evaluation of Fixed-point Inverse FLANN model of Intelligent Pressure Sensors," *Proceedings of NCC-2005*, pp.647-650, 28th-30th Jan. 2005, IIT, Kharagpur.

[4.5] G.Panda and **S.K.Mishra**, "Performance Evaluation of Fixed-point FLANN model of Intelligent Pressure Sensors", communicated to *IETE Journal of Research*, on Oct, 2005.

Chapter

5

Nonlinear Compensation of LVDT using Different ANN Techniques

THIS CHAPTER deals with the nonlinearity compensation of Linear Variable Differential Transformer (LVDT). In many practical control systems LVDT is used as the sensing element for displacement [5.1,5.2]. The performance of the control system depends upon the performance of the sensing elements. Many researchers have worked to design LVDT with high linearity [5.3-5.7]. In its conventional design methodology achieving high linearity involves complex design task. Sophisticated and precise winding machines are used to achieve that. It is difficult to have all LVDT fabricated in a factory at a time to be equally linear. LVDT having different nonlinearity present in a control system malfunctions at times because of the difference in sensor characteristics.

In Section 5.1, the electrical characteristics of LVDT are presented. Section 5.2 deals with the nonlinearity compensation of LVDT and its experimental set-up. The simulation studies are

carried out using different ANN models (MLP, FLANN and CFLANN) and the experimental dataset of a standard LVDT is dealt in Section 5.3. The results obtained from different models and their computational complexities are presented in Section 5.4. The summary and discussion of this Chapter is depicted in Section 5.5.

5.1 Linear Variable Differential Transformer (LVDT)

The LVDT consists of the primary coil and the secondary coil. The secondary coil itself has two coils connected differentially for providing the output. The two secondary coils are located on the two sides of the primary coil on the bobbin or sleeve and these two output windings (secondary coils) are connected in opposition to produce zero output at the middle position of the armature [5.1,5.2]. The lengths of primary and two identical halves of the secondary coils are b and m respectively. The coils have an inside radius r_i and an outside radius of r_0. The spacing between the coils is d. Inside the coils a ferromagnetic armature of length L_a and radius r_i (neglecting the bobbin thickness) moves in an axial direction. The number of turns in the primary coil is n_p and n_s is the number of turns in each secondary coil. The cross-sectional view of LVDT is shown in Fig. 5.1.

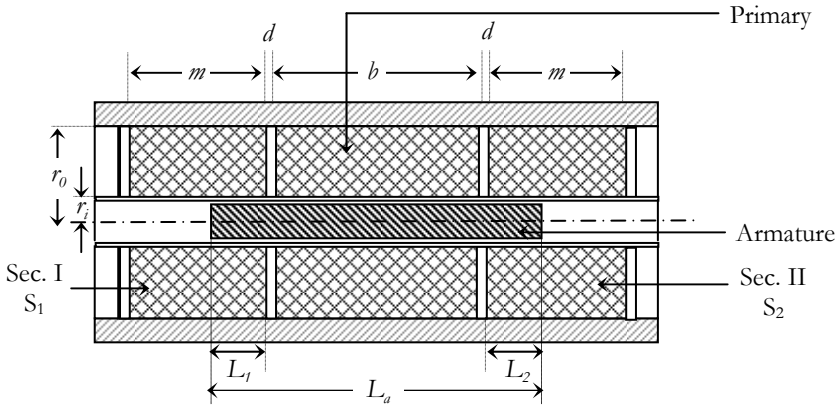


Fig. 5.1 Cross-sectional view of an LVDT

With a primary sinusoidal excitation voltage, V_p and the current I_p (r.m.s.) of frequency f, the r.m.s. voltage e_1 induced emf in the secondary coil S_1 [5.1] is,

$$e_1 = \frac{4\pi^3}{10^7} \cdot \frac{f I_p n_p n_s}{\ln\left(r_0/r_i\right)} \cdot \frac{2L_2 + b}{mL_a} x_1^2,$$

and in coil S_2,

$$e_2 = \frac{4\pi^3}{10^7} \cdot \frac{fI_p n_p n_s}{\ln(r_0/r_i)} \cdot \frac{2L_1 + b}{mL_a} x_2^2.$$

where

x_1 = Distance penetrated by the armature towards the secondary coil S_1, and x_2 = Distance penetrated by the armature towards the secondary coil S_2,

The differential voltage $e = e_1 - e_2$ is thus given by

$$e = k_1 x\left(1 - k_2 x^2\right) \tag{5.1}$$

where $x = \frac{1}{2}(x_1 - x_2)$ is the armature displacement and,

$$k_1 = \frac{16\pi^3 fI_p n_p n_s (b + 2d + x_0) x_0}{10^7 \ln(r_0/r_i) mL_a} \quad \left(Vm^{-1}\right) \tag{5.2}$$

with $x_0 = \frac{1}{2}(x_1 + x_2)$ and

$$k_2 = 1 \Big/ (b + 2d + x_0) x_0 \tag{5.3}$$

k_2 is a nonlinearity factor in (5.1), the non-linearity term ε being

$$\varepsilon = k_2 x^2 \tag{5.4}$$

For a given accuracy and maximum displacement the over-all length of the transducer is minimum for $x_0 = b$,assuming that at maximum displacement the armature does not emerge from the secondary coils. Taking the length of armature $L_a = 3b + 2d$, neglecting *2d* compared with *b* and using (5.2), (5.1) can be simplified as

$$e = \frac{16\pi^3 f I_p n_p n_s}{10^7 \ln\left(r_0/r_i\right)} \cdot \frac{2b}{3m}\left(1 - \frac{x^2}{2b^2}\right) \tag{5.5}$$

For a given primary sinusoidal excitation, the secondary output voltage e is nonlinear with respect to displacement x. This is shown in Fig. 5.2 in which the linear region of the plot is indicated as x_m. This limitation is inherent in all differential systems and methods of extending the range have been proposed by various authors mainly by appropriate design and arrangement of the coils. Some of these are:

(i) Balanced linear tapered secondary coils. But the improvement in linearity is not significant [5.3].

(ii) Over-wound linear tapered secondary coil improves the linearity to a certain extent [5.4].

(iii) Balanced over-wound linear tapered secondary coils. Its performance is similar to (ii) [5.5].

(iv) Balanced profile secondary coils helps in extending linearity range by proper profiling of the secondary coils [5.6].

(v) Use of complementary tapered winding extends the linearity range but the winding is quite complicated as it involves sectionalised winding [5.7].

All these methods improve the linearity performance to some extent and are based on improvement on constructional design. The methods are nonelectronic in nature and hence are nonadaptive. As a result they do not account for the nonlinearity contributed due to environmental changes. Hence there is a need to devise intelligent methods for adaptively compensating the nonlinearity contributed due to internal construction and temperature variation of LVDT. Incorporating the sensing feature of malfunctioning and adaptively compensating capability makes the LVDT an intelligent one.

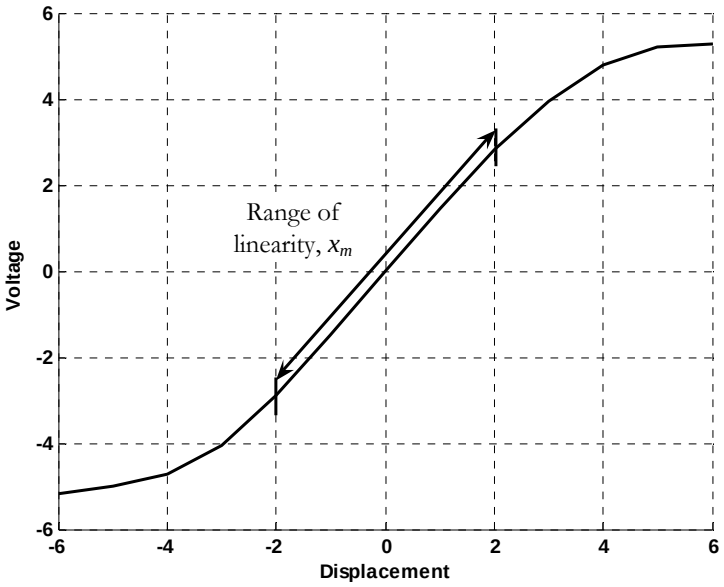


Fig. 5.2 Range of linear region of LVDT

5.2 Nonlinearity Compensation of LVDT

The proposed nonlinearity compensation scheme is shown in Fig. 5.3. In this scheme the LVDT can be controlled by a displacement actuator. The main controller gives an actuating signal to the displacement actuator, which displaces the core of the LVDT. The differential voltage of the LVDT after being demodulated does not keep linear relationship with the displacement. The nonlinearity compensator can be developed by using different ANN techniques like MLP, FLANN and cascaded FLANN. The output of the ANN based nonlinearity compensator is compared with the desired signal (actuating signal of the displacement actuator) to produce an error signal. With this error signal, the weight vectors of the ANN model are updated. This process is repeated till the mean square error (MSE) is minimized. Once the training is complete, the LVDT together with the ANN model acts like a linear LVDT with enhanced dynamic range.

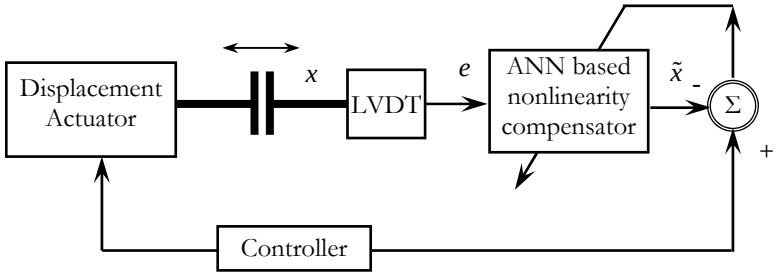


Fig. 5.3 Scheme of an intelligent nonlinearity compensation of LVDT

The practical set-up of the LVDT along with the ANN based nonlinearity compensator after the training is shown in Fig. 5.4.

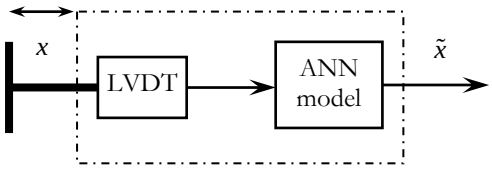


Fig. 5.4 Practical set-up of LVDT after training

The percentage of linearity (*PEL*) of the LVDT can be computed as

$$PEL = \frac{(desired\ output - estimated\ output)}{desired\ output} * 100 \tag{5.6}$$

5.3 Simulation Studies

To demonstrate the effectiveness these ANN based nonlinear compensators, computer simulation studies are carried out using experimental data obtained from a typical LVDT having following specifications.

Number of turns in two secondary coils, S_1 and S_2, n_s = 3300 turns each wound over it uniformly with the two coils separated by a Teflon ring.
Core diameter is, $2r_i$ = 4.4 mm,
Core length is, L_a = 4.5 mm.
The primary sinusoidal excitation voltage, V_p = 10V (peak-to-peak) of frequency, f = 5 kHz. V_p is 6.712 V_{rms}.
Primary winding resistance, r_p = 260ohm.
Secondary winding resistance, s_{r1} = 426.8 ohm.
Secondary winding resistance, s_{r2} = 414 ohm.

Two secondaries are wound in opposite directions. The pitch is kept around 0.02-0.03 mm. The experimentally measured data is presented in Table-5.1.

TABLE-5.1: EXPERIMENTAL MEASURED DATA

Displacement (x in mm)	Differential Output voltage (e_{rms})	Demodulated voltage output (e)
-30	4.085	-5.185
-25	3.956	-5.017
-20	3.731	-4.717
-15	3.221	-4.039
-10	2.359	-2.896
-5	1.273	-1.494
Null position, 0	0.204	0.001
5	1.153	1.462
10	2.226	1.810
15	3.118	3.962
20	3.748	4.799
25	4.050	5.225
30	4.085	5.276

5.3.1 MLP based Nonlinearity Compensator

The differential or demodulated voltage e at the output of LVDT is normalized by dividing each value with the maximum value. The normalized voltage output e is subjected to input to the MLP based nonlinearity compensator. In case of the MLP, we used different neurons with different layers. However, the 1-100-50-1 network is observed to perform better hence it is chosen for simulation. Each hidden layer and the output layer contains tanh(.) type activation function. The output of the MLP based nonlinearity compensator is compared with the normalized input displacement of the LVDT. The BP algorithm, in which both the learning rate and the momentum rate are chosen as 0.02 and 1 respectively, is used to adapt the weights of the MLP. Applying various input patterns, the ANN weights are updated using the BP algorithm. To enable complete learning, 10,000 iterations are made. Then, the weights of the various layers of the MLP are frozen and stored in the memory. During the testing phase, the frozen weights are used in the MLP model. Fig. 5.5 shows the nonlinearity compensation of LVDT by MLP. In this model the *PEL* is obtained to be 3.1120 %.

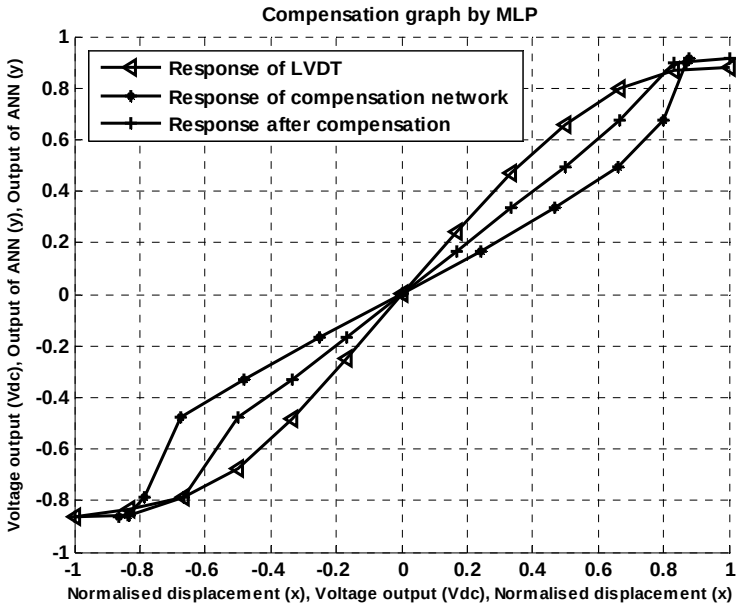


Fig. 5.5 Response due to MLP (1-100-50-1) network

5.3.2 FLANN based Nonlinearity Compensator

In case of the FLANN model, the differential output of the LVDT is functionally expanded. In this case we have used trigonometric functional expansion because it provides better nonlinearity compensation compared to others. The expanded elements are then multiplied with a set of weights, and then summed to yield the final output. The output of the FLANN model is compared with the normalized input displacement of the LVDT to give the error signal. Using the error and the input to the inverse model the weights are updated by the simple Least Mean Square (LMS) algorithm. Two different experiments are carried out using the same data set of Table-5.1. The first experiment is carried out with less number of functional expansions i.e. P=10 with 10,000 iterations. In the second experiment the number of functional expansions is chosen to be 50 and the number of iterations is 5,000. Fig. 5.6 shows the response of the LVDT and the nonlinear compensator for the first experiment. Fig. 5.7 shows the same results for the second experiment. From these plots, it is observed that there is significant improvement in linearity by FLANN based compensator with P=50. The model FLANN with 10 functional expansions has achieved *PEL* about 0.3251 %; whereas with 50 functional expansions the

PEL value has increased to 5.12×10^{-15} %. Therefore in the second experiment the nonlinearity compensation is better than the first one [5.13].

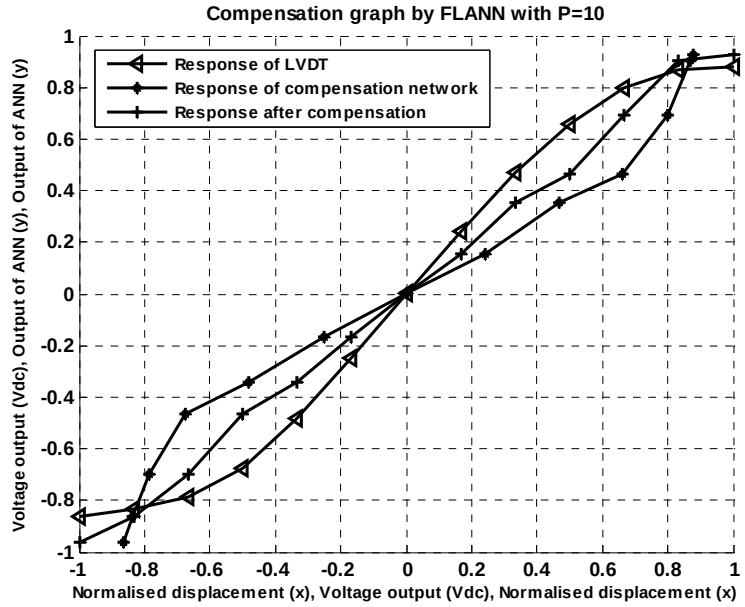


Fig. 5.6 Response due to trigonometric series FLANN with $P = 10$

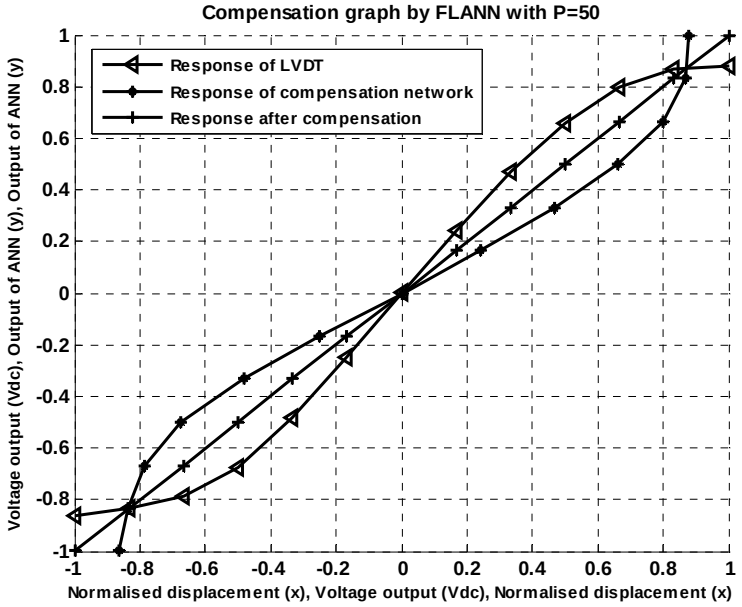


Fig. 5.7 Response due to trigonometric series FLANN with $P = 50$

5.3.3 CFLANN based Nonlinearity Compensator

The linearity characteristics of LVDT can further be improved by the proposed method by employing Cascaded FLANN (CFLANN) structure, which essentially consists of two-stage FLANN models. In this case each FLANN structure uses less number of nonlinear expansions as compared to that needed for one-stage FLANN. Fig. 5.8 shows a 2-stage nonlinear expansion scheme of the differential voltage e. The differential voltage e at the output of LVDT is subjected to trigonometric expansion (M_1 values) to achieve nonlinearly mapped values. These are then multiplied with a set of weights and then added to produce an intermediate output (stage-I). This output value undergoes further expansion (M_2 values) which are then weighted and summed to generate the final output (stage-II). It is compared with the desired signal (actuating signal of the displacement actuator) to produce an error signal. With this error signal, the weight vectors (2 sets) of the CFLANN model are updated. This process is repeated till the mean square error (MSE) is minimized. Once the training is complete, the LVDT together with the proposed model acts like a linear sensor with enhanced dynamic range.

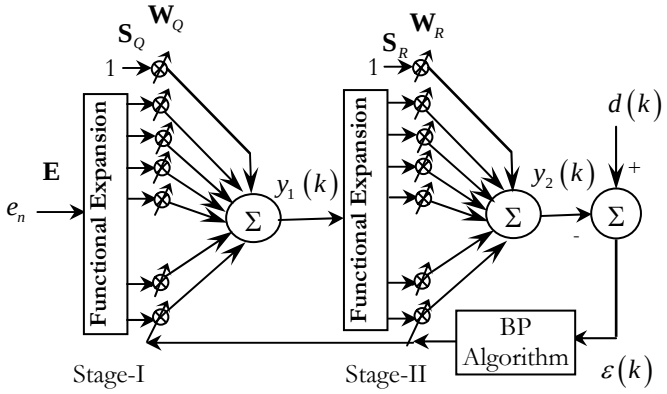


Fig. 5.8 Structure of CFLANN model

Mathematical Derivation of CFLANN

Let $\mathbf{E}$ be the input vector of size $N\times1$ which represents N number of elements; the n^{th} element is given by:

$$\mathbf{E}(n) = e_n, 1 \le n \le N \tag{5.7}$$

Each element undergoes nonlinear expansion to form M_1 elements such that the resultant matrix has the dimension of $N \times M_1$.

The trigonometric expansions of e_n are computed as

$$s_q = \begin{cases} e_n & \text{for } q = 1 \\ \sin\left(m\pi e_n\right) & \text{for } q = 2, 4, \ldots, M_1 \\ \cos\left(m\pi e_n\right) & \text{for } q = 3, 5, \ldots, M_1 + 1 \end{cases} \tag{5.8}$$

where $m = 1, 2, \cdots, M_1/2$. In the proposed expansion M_1 is always an even number. For example, if $n = 1$ and $M_1 = 6$ we have $m = 1, 2, 3,$ and $q = 1, 2, \ldots, 7.$ Then the expanded elements are

$$\begin{aligned} s_1 &= e_1, \\ s_2 &= \sin\left(\pi e_1\right), \\ s_3 &= \cos\left(\pi e_1\right), \\ s_4 &= \sin\left(2\pi e_1\right), \\ s_5 &= \cos\left(2\pi e_1\right), \\ s_6 &= \sin\left(3\pi e_1\right), \\ s_7 &= \cos\left(3\pi e_1\right). \end{aligned}$$

In matrix notation the expanded elements of the input vector **E**, is denoted by $\mathbf{S}_Q$ of size $N \times (M_1+1)$.

The bias input is unity. So an extra unity value is padded with the $\mathbf{S}_Q$ matrix and the dimension of the $\mathbf{S}_Q$ matrix becomes $N \times Q$, where $Q = \left(M_1 + 2\right)$.

Let the weight vector of stage-I be represented by $\mathbf{W}_Q$ having Q elements. The output of stage-I, y_1 is given as

$$y_1 = \sum_{q=1}^{Q} s_q \cdot w_q \tag{5.9}$$

In matrix notation the output can be,

$$\mathbf{Y}_1 = \mathbf{S}_Q \cdot \mathbf{W}_Q^{\mathrm{T}} \tag{5.10}$$

In the ordinary FLANN model, the final output y_1 is compared with the desired response. The weights, w_q are updated by using least mean square (LMS) algorithm [5.11].

However, in case of CFLANN, the output of stage-I, y_1 is further expanded nonlinearly to M_2 elements each. Hence the functional expansion results

$$s_r = \begin{cases} y_1 & \text{for } r = 1 \\ \sin\left(l\pi y_1\right) & \text{for } r = 2,4,\ldots,M_2 \\ \cos\left(l\pi y_1\right) & \text{for } r = 3,5,\ldots,M_2+1 \end{cases} \tag{5.11}$$

where $l = 1,2,\cdots,M_2/2.$ Here M_2 is an even number. The matrix form of the nonlinearly expanded elements is represented by $\mathbf{S}_R$ having dimension $N\times(M_2+1)$.

In the same way as stage-I, the bias input is included to form $\mathbf{S}_R$ with dimension $N\times R$, where $R = \left(M_2 + 2\right)$. Let $\mathbf{W}_R$ be the weight vector of stage-II with R elements. The output of IInd stage is given by

$$y_2 = \sum_{r=1}^{R} s_r \cdot w_r \tag{5.12}$$

In matrix notation the final output is obtained as

$$\mathbf{Y}_2 = \mathbf{S}_R \cdot \mathbf{W}_R^{\mathrm{T}} \tag{5.13}$$

At k^{th} iteration the error signal $\varepsilon\left(k\right)$ can be computed as

$$\varepsilon\left(k\right) = d\left(k\right) - y_2\left(k\right) \tag{5.14}$$

where $d(k)$ is the desired signal which is same as the control signal given to the displacement actuator.

Let $\xi(k)$ denote the cost function at iteration *k* and is given by

$$\xi(k) = \frac{1}{2}\sum_{j=1}^{J} \varepsilon_j^2(k) \tag{5.15}$$

where *J* is the number of nodes at the final output layer.

The weight vectors (of both stage-I and stage-II) can be updated by back-propagation (BP) algorithm, [5.12] as

$$w_r(k+1) = w_r(k) + \Delta w_r(k) \tag{5.16}$$

$$w_q(k+1) = w_q(k) + \Delta w_q(k) \tag{5.17}$$

where,

$$\Delta w_r(k) = -\eta \cdot \frac{\partial \xi(k)}{\partial w_r(k)} \tag{5.18}$$

where η is the learning-rate parameter *(0 ≤ η ≤ 1)*. The minus sign accounts for gradient descent in weight space. (5.18) can be solved to yield

$$\Delta w_r(k) = \eta \cdot \delta_r(k) \cdot s_r \tag{5.19}$$

where $\delta_r(k)$ is the local gradient and is defined as,

$$\begin{aligned} \delta_r(k) &= -\frac{\partial \xi(k)}{\partial y_2(k)} \\ &= \varepsilon(k) \end{aligned} \tag{5.20}$$

Similarly the change in weights of stage-I is given by

$$\Delta w_q(k) = -\eta \cdot \frac{\partial \xi(k)}{\partial w_q(k)} \tag{5.21}$$

$$\Delta w_q(k) = \eta \cdot \delta_q(k) \cdot s_q \tag{5.22}$$

where $\delta_q(k)$ is the local gradient of stage-I and is defined as

$$\begin{aligned} \delta_q(k) &= -\frac{\partial \xi(k)}{\partial y_1(k)} \\ &= \varepsilon(k) \cdot w_j \cdot \varphi'\left(y_1(k)\right) \end{aligned} \tag{5.23}$$

where $s_r = \varphi\left(y_1(k)\right)$ and is obtained by functionally expanding $y_1(k)$. $\varphi'\left(y_1(k)\right)$ is the first order derivative of $\varphi\left(y_1(k)\right)$.

Simulation studies are carried out using the same experimental data of Table: 5.1, but employing the CFLANN model developed in this Chapter. In this case 12 trigonometric expansions are taken in stage-I and 4 expansions in stage-II respectively and the *PEL* is found to be 3.41×10^{-15} % [5.15]. The plot of CFLANN model is shown in Fig. 5.9.

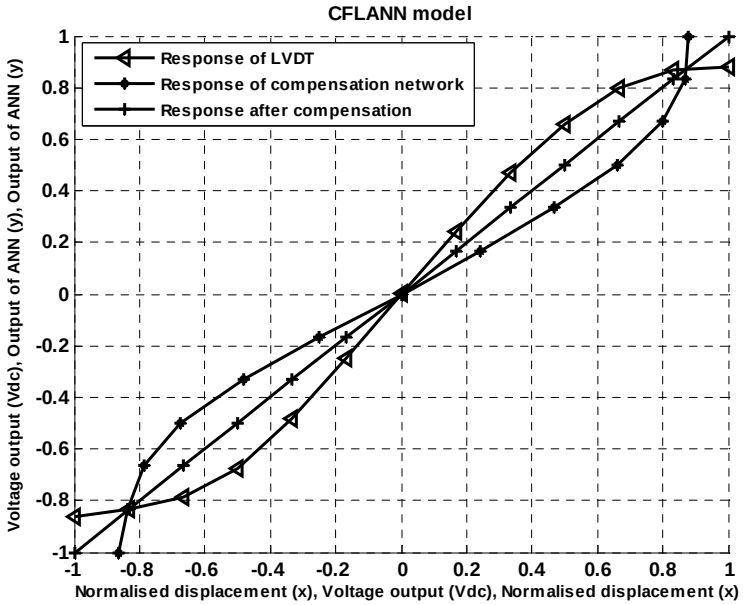


Fig. 5.9 Plots of response of the CFLANN model,
Nonlinearly mapped inputs = 12 (stage-I), 4 (stage-II)

5.4 Computational Complexity

In this section the computational complexity of MLP, FLANN and CFLANN models is evaluated and compared. In case of MLP, the structure chosen is 1-H1-H2-1, i.e. number of input, hidden layer-1, hidden layer-2, and output nodes are 1, H1, H2, 1, respectively. Three basic computations, i.e. addition, multiplication and computation of tanh(.) are involved in forward path of the MLP; whereas in FLANN and CFLANN, along with the additions and the multiplications, some extra computations are required for calculating sin(.) and cos(.) functions. In FLANN, the order chosen is P. The total number of elements after P numbers of trigonometric expansion is M=2P+1. The computational complexity involved during the training period is not considered as this operation is performed offline. Table 5.2 presents the operation units during normal operation of various ANN models.

TABLE-5.2: COMPUTATIONAL COMPLEXITY

Network	Addition	Multiplication	tanh(.)	sin(.)/cos(.)	Percentage of Linearity
MLP (1-100-50-1)	5150 (100+5000+50)	5301 (200+5050+51)	151	---	**3.1120%**
FLANN (1-22-1)	21	22	---	20 (10+10)	**0.0033%**
FLANN (1-52-1)	51	52	---	50 (25+25)	**5.12×10^{-150}%**
CFLANN (1-14-6-1)	18	20	---	16 (12+4)	**3.41×10^{-150}%**

From Table-5.2, it is evident that the CFLANN model drastically reduces the computational complexity. It also offers better linearity compared to MLP and FLANN models (1-22-1). The 1-52-1 FLANN model though offers almost similar linearity as CFLANN; its complexity is much higher.

5.5 Summary and Discussion

Different efficient ANN based nonlinearity compensators for LVDT are developed in this Chapter. The nonlinearity compensation capability of MLP is poor; the *PEL* is about 3.1120 %. The high order FLANN based nonlinearity compensator improves the linearity quite appreciably but involves more computations. To reduce the computational complexity of the FLANN model, the CFLANN structure is proposed here. With reduced number of expansions at each stage, the CFLANN model provides better performance compared to others. In Table-

5.2, it is noted that the CFLANN (1-14-6-1) model employs 18 additions, 20 multiplications and 16 numbers of sin(.)/cos(.) expansions as compared to 51 additions, 52 multiplications and 50 numbers of sin(.)/cos(.) expansions needed for FLANN model. Thus, considering performance and computational complexity, it is observed that the proposed 2-stage FLANN model is preferred one compared to other models studied in this Chapter.

REFERENCES

[5.1] Hermann K.P. Neubert, *"Instrument Transducers: An Introduction to their performance & design"*, 2nd edition, Oxford University Press.

[5.2] John P. Bentley, *"Principles of Measurement Systems"*, 3rd edition, Pearson Education.

[5.3] Y. Kano, S. Hasebe, and H. Miyaji, "New linear variable differential transformer with square coils," *IEEE Transactions on Magnetics*, Vol. 26, Issue:5, pp.2020 – 2022, Sep 1990.

[5.4] S.C. Saxena, and S.B.L. Sexena, "A self-compensated smart LVDT transducer," *IEEE Transactions on Instrumentation and Measurement,* Vol. 38, Issue: 3, pp.748 – 753, June 1989.

[5.5] G. Y. Tian, Z, X. Zhao, R. W. Baines, N. Zhang, "Computational algorithms foe linear variable differential transformers (LVDTs)" *IEE Proc.-Sci. Measurement Technology*, Vol. 144, No. 4,pp 189-192, July 1997.

[5.6] D. Crescini, A. Flammini, D. Marioli, and A. Taroni, "Application of an FFT-based algorithm to signal processing of LVDT position sensors" *IEEE Transactions on Instrumentation and Measurement*, Vol. 47, No. 5, pp. 1119-1123 Oct. 1998.

[5.7] R. M. Ford, R. S. Weissbach, and D. R. Loker, "A novel DSP-based LVDT signal conditioner" *IEEE Transactions on Instrumentation and Measurement*, Vol. 50, No. 3, pp. 768-774, June 2001.

[5.8] J.C. Patra, A. C. Kot, and G. Panda, "An intelligent pressure sensor using neural networks," *IEEE Transactions on Instrumentation and Measurement,* vol.49, No.4, pp.829-834, Aug.2000.

[5.9] J. C. Patra, G. Panda, and R. Baliarsingh, "Artificial neural network-based nonlinearity estimation of pressure sensors", *IEEE Transactions on Instrumentation and Measurement*, vol. 43, No. 6, pp.874-881, Dec. 1994.

[5.10] J.C. Patra, and R.N. Pal, "A functional link artificial neural network for adaptive channel equalization", *Signal Processing* 43(1995) 181-195, vol.43, no.2, May 1995.

[5.11] B. Widrow and S.D. Sterns, "*Adaptive Signal Processing"* Prentice-Hall, Inc. Engle-wood Cliffs, New Jersey, 1985.

[5.12] S. Haykin, "*Neural Networks: A comprehensive foundation*" 2nd Edition, Pearson Education Asia, 2002.

[5.13] **S.K. Mishra**, G. Panda, D.P. Das, S.K. Pattanaik and M.R. Meher, "A Novel Method of Designing LVDT using Artificial Neural Network," *Proceedings of IEEE, ICISIP-2005*, pp.223-227, Jan, 2005.

[5.14] G. Panda, **S.K. Mishra**, and D.P. Das, "A novel method of extending the linearity range of LVDT using artificial neural network" (paper ID: SNA-D-05-00679) submitted (under final revision) to *Sensors & Actuators: A. Physical, Elsevier*, on Nov, 2005.

[5.15] **S.K. Mishra** and G. Panda, "A Novel Method for Designing LVDT and its Comparison with Conventional Design",(paper ID: 6014) accepted in *IEEE Sensors Applications Symposium (SAS-2006)* held on Houston Marriott West Loop by the Galleria, Houston, Texas, February 7-9, 2006.

Chapter

6

Adaptive Nonlinear Compensation of Control Valve Actuator using Soft Computing Techniques

THE CONTROL VALVE ACTUATORS are the most common actuators in the process industries. They are found in process plants, manipulating flows to maintain controlled variable at their set points. A control valve acts as a variable restriction in a process pipe. By changing its opening it changes the resistance to flow and, thus, the flow itself [6.1]. The equal percentage control valve actuator devices have nonlinearities in their inherent flow characteristics. But when it is put in a pipe line, it acts as nearly linear control valve, called installed characteristics. The linearity depends on the pressure difference across the valve. As the pressure difference across the value varies, it is always required to compensate to get very accurate linear characteristics. Some of other fundamental actuator nonlinearities include friction, deadzone, saturation, backlash and hysteresis. Amongst these the friction, the deadzone and the saturation are static nonlinearities, whereas the backlash and the hysteresis are dynamic

nonlinearities. Various attempts have been made by many researchers to compensate the *static* and *dynamic nonlinearities* present in the actuators [6.3-6.10]. The motivation of this Chapter is to develop the efficient adaptive inverse models with less computational complexity to compensate the nonlinearity effect of the equal percentage control valve actuator characteristics.

The rest of the Chapter is organized as follows. In Section 6.1, the operation of different control valve characteristics is presented. Section 6.2 develops the proposed adaptive inverse algorithms required to develop the intelligent model and their simulation studies. Finally, the summary and discussion are presented in Section 6.3.

6.1 Control Valve Actuator

The control valve is essentially a variable resistance to the flow of a fluid, in which the resistance and therefore the flow, can be changed by a signal from a process controller. The cross sectional view of a control valve is shown in Fig. 6.1. The function of a control valve is to vary the flow of fluid through the valve by means of a change of pressure to the valve top.

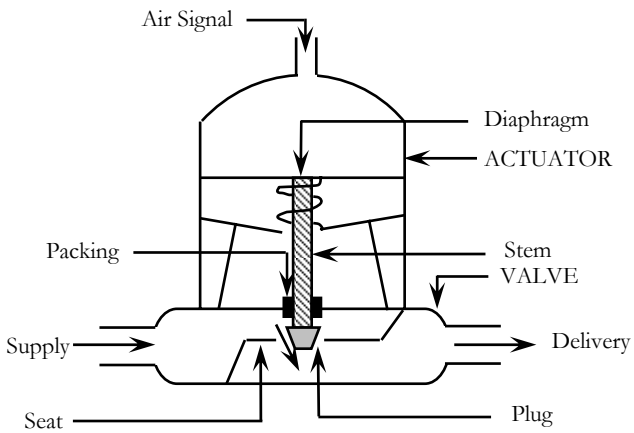


Fig. 6.1 Cross-sectional view of a control valve actuator

The relation between the flow through the valve and the valve stem position (or lift) is called the valve characteristics. It may be differentiated as *inherent flow characteristics* and *installed flow characteristics*. The *inherent flow characteristics* refers to the characteristics observed with constant pressure drop across the valve. Where as the *installed flow characteristics* refers to the characteristic observed when the valve is in service with varying pressure drop and other changes in the system. The relation between the flow through the valve and the valve stem position (or lift) is called the valve characteristics [6.1, 6.2].

In general, the flow through a control valve for a specific fluid at a given temperature can be expressed as

$$q = f_1(L, p_0, p_1) \tag{6.1}$$

where q = volumetric flow rate,

L = valve stem position (or lift)

p_0 = upstream pressure

p_1 = downstream pressure

The *inherent valve characteristics* is determined for fixed values of p_0 and p_1, for which (6.1) becomes

$$q = f_2(L) \tag{6.2}$$

Let, $m = q/q_{\max}$ and $x = L/L_{\max}$, whe re $q_{\max}$ is the flow when the valve stem is at its maximum lift $L_{\max}$ (valve is full open), x is the fraction of maximum lift, and m is the fraction of maximum flow.

Hence (6.2) can be rewritten as

$$m = q/q_{\max} = f(L/L_{\max})$$

or, $$m = f(x) \tag{6.3}$$

The sensitivity may be represented as

$$s = \frac{dm}{dx} \tag{6.4}$$

In terms of valve characteristics, valves can be divided into three types: decreasing sensitivity, linear and increasing sensitivity. For the decreasing sensitivity type, the sensitivity (slope) decreases with m and it is known as "quick opening flow characteristics". For the linear flow characteristics, the sensitivity is constant and the characteristic is a straight line. For the increasing type, the sensitivity increases with flow and it is also known as "equal percentage valve".

Mathematically the linear flow characteristics can be represented as

$$\frac{dm}{dx} = \alpha \tag{6.5}$$

where α is a constant. And assuming that the valve is shut tight when the lift is at lowest position, we have $m = 0$ at $x = 0$. For a single-seated valve that is not badly worn, the valve can be shut off for $x = 0$.

Integrating (6.5) and introducing the limits $m = 0$ at $x = 0$ and $m = 1$ at $x = 1$ gives

$$\int_0^1 dm = \int_0^1 \alpha dx \tag{6.6}$$

By solving this we get $\alpha = 1$.

For $\alpha = 1$, (6.5) can now be integrated to give

$$m = x \quad \text{(linear valve)} \tag{6.7}$$

Hence the linear type valves are commonly used in liquid level loops and in other processes where the pressure drop across the valve is fairly constant.

For the equal percentage valve, the slope can be

$$\frac{dm}{dx} = \beta m \tag{6.8}$$

where β is constant. Integration of (6.8) gives

$$\int_{m_0}^{m} \frac{dm}{m} = \int_0^x \beta dx \tag{6.9}$$

or, $$\ln \frac{m}{m_0} = \beta x \tag{6.10}$$

where m_0 is the flow at $x = 0$.

The basis for calling the valve characteristics equal percentage can be by rearranging (6.8) in the form

$$\frac{dm}{m} = \beta dx \quad \text{and} \quad \frac{\Delta m}{m} = \beta \Delta x \tag{6.11}$$

in this form it can be seen that an equal fractional (or percentage) change in flow $(\Delta m/m)$ occurs for a specified increment of change in stem position (Δx), regardless of where the change in stem position occurs along the change the characteristic curve.

The term β can be expressed in terms of m_0 by inserting $m = 1 \text{ at } x = 1$ into (6.10). The result is

$$\beta = \ln\left(\frac{1}{m_0}\right)$$

Then (6.9) for m gives

$$m = m_0 e^{\beta x} \quad \text{(equal percentage valve)} \tag{6.12}$$

m_0 is the flow at $x = 0$. m_0 cannot be zero since there may be some leakage when the stem is at its lowest position. For some valves, especially large ones, the valve manufacturer intentionally allows some leakage at minimum lift $(x = 0)$ to prevent binding and wearing of the plug and seat surfaces.

The Rangeability (R) is defined as the ratio of maximum flow to minimum controllable flow over which the valve characteristic is followed,

$$R = \frac{m_{\max}}{m_{\min,\text{controllable}}} \tag{6.13}$$

The main constraint in equal percentage valve is low rangeability. The rangeability mainly depends on the term m_0. For example, if $m_0 = 0.02$, the rangeability is 50. The application of

these valves requires in services where large variations in pressure drop are expected or in services where a small percentage of the total system pressure drop is to be taken.

6.2 Adaptive Inverse Models

The practical setup of the interfacing of the inverse model with the control valve actuator is shown in Fig. 6.2. This figure depicts that the adaptive inverse model is connected in series with the actuator to compensate the associated varying nonlinearity of the actuator.

In this section we have employed three types of inverse models (LMS, MLP and FLANN) for nonlinearity compensation of the control valve actuator and their performance in terms of convergence speed, MSE level, accuracy and computational complexity has been compared. Brief discussion of each network and its mathematical derivation is mentioned in Chapter 2. The simulation studies of all types of adaptive inverse model are dealt in this section.

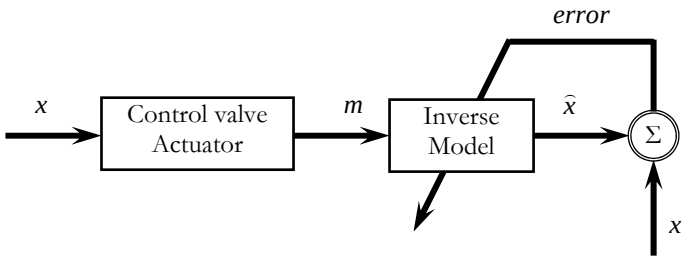


Fig. 6.2 Set-up for compensating nonlinearity of control valve actuator

6.2.1 LMS based Inverse Model

The detail of the LMS algorithm [6.11] is already reported in Chapter 2. The input flow x varies from 1 to 15 psig with an increment of 1 psig. Accordingly the output of the control valve is calculated by (12). The normalized output of the control valve serves as the input to the inverse model and the output of that is compared with the normalized pressure. In this case, the order of the tap is 20. The learning rate of the updation algorithm is taken as 0.1. To make the model effective, 5000 iterations are used to train the inverse model. The weights of the network are frozen after updation. The *PEL* is achieved as 4.48×10^{-15} %. The overall response of the model is shown in Fig. 6.3.

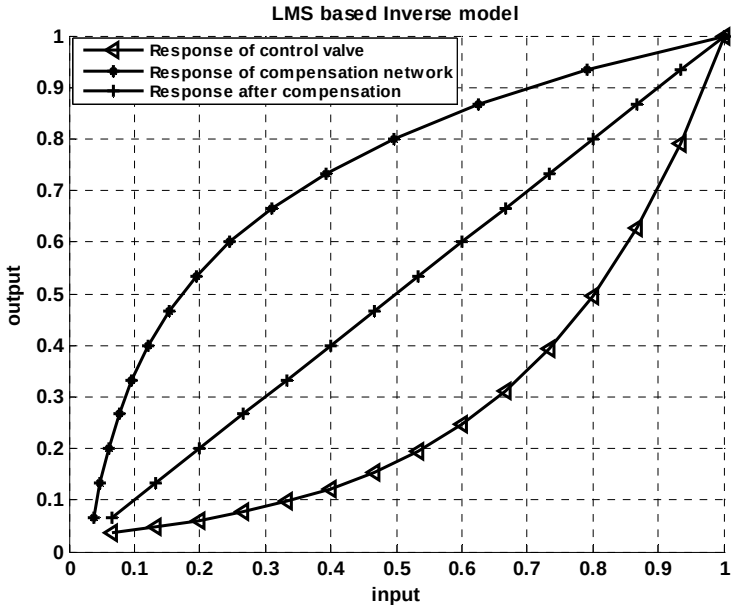


Fig. 6.3 Plots of response of the LMS based inverse model

6.2.2 MLP based Inverse Model

In case of the MLP, (1-50-30-1) network has been taken for the development of the inverse model of the actuator. Each hidden and the output layer consists of a nonlinear activation function of tanh(.) type [6.12]. The weights of the network are updated by using the back propagation algorithm. The learning rate and the momentum parameter of the algorithm are chosen as 0.02 and 1 respectively. The number of iterations is chosen as 10000 to effectively train the network. The response of the actuator along with the response of the compensation network is plotted in Fig. 6.4.

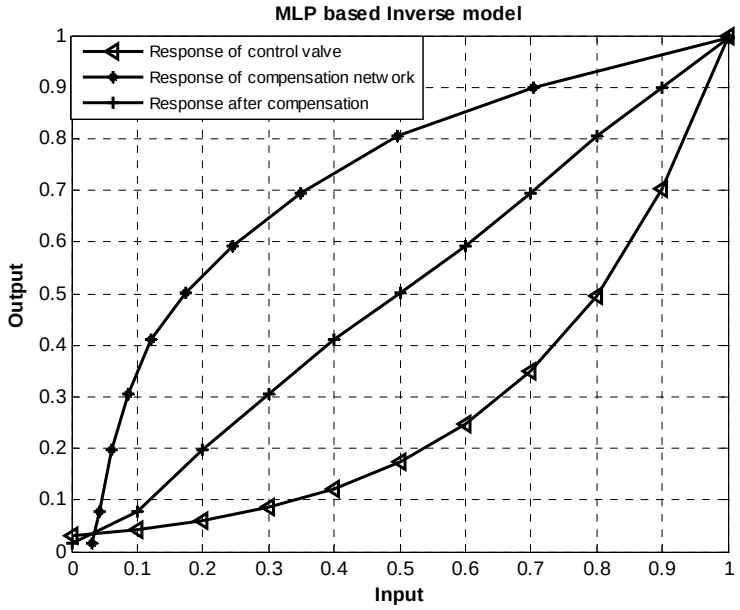


Fig. 6.4 Plots of response of the MLP based inverse model

Once the training is complete, the weights of the network are frozen in the memory. In this case, the *PEL* is obtained as 0.3106 %.

6.2.3 FLANN based Inverse Model

In this case, the output of the control valve actuator is functionally expanded into 50 elements. The FLANN structure is formed as 1-52-1, including the original input and the bias term. There are 52 weight elements in the network, and these are updated by employing the LMS algorithm. The learning rate is taken as 0.01. For successfully training the network, 10000 iterations are chosen. In this case the *PEL* is 7.70×10^{-5} %. Once the training is over, the coefficients are stored and frozen in the memory. For testing phase, the inverse model is connected in series with the control valve. The change of output of the control valve with respect to the input pressure is fed to this inverse model. The FLANN model output is computed and compared with the actual output to verify the effectiveness of the model [6.13]. The plot of the FLANN based inverse model is shown in Fig. 6.5.

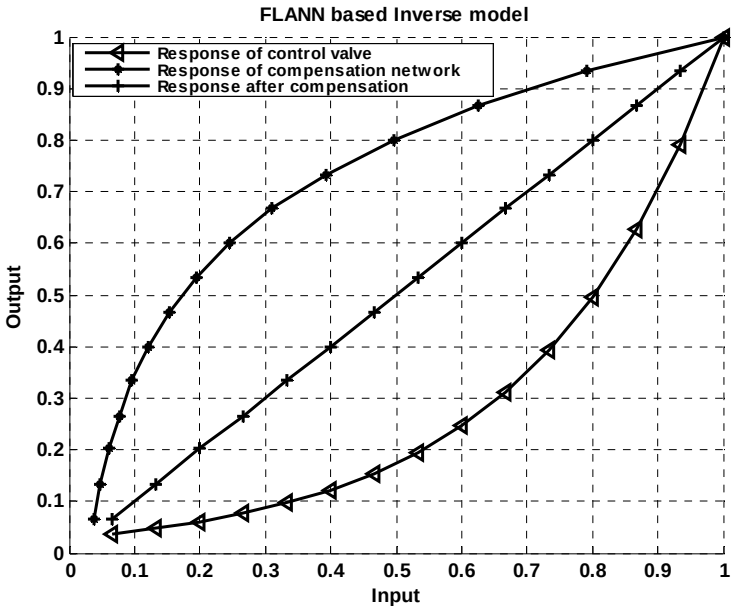


Fig. 6.5 Plots of response of the FLANN based inverse model

6.3 Summary and Discussion

In this Chapter, three adaptive inverse models have been proposed. A comparative study between of these three models has been carried out through simulation study. It is observed that the LMS based inverse model offers better linearity performance compared to others. The LMS

based inverse model is simple to implement and offers improved performance compared to others. This is because the actuator offers less nonlinearity which does not require MLP and FLANN structures.

REFERENCES

[6.1] D. R. Coughanowr, "*Process Systems Analysis and Control*", 2nd edition, McGraw-Hill international editions, 1991.

[6.2] G. Tao and P. V. Kokotovic, "*Adaptive Control of Systems with Actuator and Sensor Nonlinearities"*, John Wiley & Sons, New York, 1996.

[6.3] B. Armstrong-Htlouvry, P. Dupont, and C. Canudas de Wit, "A survey of models, analysis tools and compensation methods for the control of machines with fiction," *Automatica*, vol.30, no.7, pp.1083-1138, 1994.

[6.4] C. Canudas de Wit, P. Noel, A. Aubin, and B. Brogliato, "Adaptive friction compensation in robot manipulators: Low velocities," *Int. J. Robot. Res.,* vol.10, no.3, Jun. 1991.

[6.5] S. W. Lee and J. H. Kim, "Robust adaptive stick-slip fiction compensation," *IEEE Trans. Ind. Electron.*, vol. 42, no.5, pp. 474-479, Oct. 1995.

[6.6] Y. H. Kim and F. L. Lewis, "Reinforcement adaptive learning neural-net-based friction compensation control for high speed and precision," *IEEE Trans. Control Sysi. Technol.*, vol. 8, no. l, pp.118-126, Jan. 2000.

[6.7] C. A. Desoer and S. M. Shahruz, "Stability of dithered nonlinear systems with backlash or hysteresis," *Int. J. Contr.*, vol. 43, no. 4, pp. 1045-1060, 1986.

[6.8] R. R. Selmic, F. L. Lewis, "Deadzone Compensation in Nonlinear Systems using Neural Network", *Proc. of the 37th IEEE conference of Decision & control*, Dec, 1998.

[6.9] Y. D. Song, T. L. Mitchell, and H. Y. Lai, "Control of a class of nonlinear uncertain systems via compensated inverse dynamics approach," *IEEE Trans. Automat. Control*, vol. 39, no. 9, pp. 1866-1871, Sep. 1994.

[6.10] R. R. Selmic, V. V. Phoha and F. L. Lewis, "Intelligent control of Actuator Nonlinearities", *Proc. of the 42nd IEEE Conf. on Decis. & Contr.,* Maui, Hawaii USA, Dec, 2003.

[6.11] B. Widrow, S. D. Stearns, "*Adaptive Signal Processing*", Pearson Education Asia, 2002.

[6.12] S. Haykin, "*Neural Networks: A Comprehensive Foundation*", 2nd edition, Pearson Education Asia, 2002.

[6.13] **S.K.Mishra,** G.Panda, T.K.Dan and S.K.Pattanaik, "Artificial Neural Network based Nonlinearity Estimation of Control Valve Actuators", *Proceeding of National Conference on Soft Computing Techniques for Engineering Applications (SCT2006),* pp.223-232, 24th-26th Mar, 2006, at NIT, Rourkela.

Chapter

7

Conclusions

THE overall conclusions of the thesis are the following:

(1) It investigates on the nonlinearity issues relating to three typical sensors:

(i) The CPS

(ii) The LVDT

(iii) The control valve actuators.

(2) The nonlinearity problem gives rise to the following difficulties:

(i) Non accuracy in measurement

(ii) Limitation of dynamic range (linearity region)

(iii) Full potentiality of the sensor cannot be utilized.

(3) The nonlinearity problem arises due to:

(i) Environmental changes such as change in temperature, humidity and atmospheric pressure

(ii) Aging

(iii) Constructional limitations.

(4) In this thesis adaptive and intelligent methods for compensation of nonlinearities have been proposed and have been applied to three typical sensors.

(5) These methods are based on the following structures:

(i) MLP

(ii) FLANN

(iii) CFLANN

(iv) RBFNN

The learning algorithms employed in the thesis are:

(i) LMS algorithm in FLANN

(ii) BP algorithm in MLP and CFLANN

(iii) RBF learning algorithm

(6) Basically nonlinear compensation has been achieved through

(i) Direct modeling of the sensors

(ii) Inverse modeling of the sensors

(7) Exhaustive simulation studies of various methods show that MLP and RBFNN structures provide improved linearity compensation performance but involves more computations and tedious to implement. However, the proposed FLANN and CFLANN based solutions offer quite satisfactory performance and involve low complexity.

(8) The thesis also presents the fixed-point implementation of the FLANN based direct and inverse models of the sensors.

7.1 Scope for Further Research Work

Many further research work may be carried out on the same and related topics.

(1) For developing the inverse model, supervised learning has been used. It has employed training data. In many situations the training data is not available. In absence of such data, training can be carried out by employing blind techniques which are known as unsupervised method. Investigation is needed to develop nonlinear compensator using blind techniques such as higher order statistics (HOS).

(2) The investigation made in this thesis can also be extended to other types of sensors and instruments.

(3) Practical implementation of the inverse model as a plug-in-module, cascading it with physical sensors and using the combined ones for real-time applications is very important. Further research work can be carried out in this direction.

(4) FPGA implementation of the proposed nonlinearity compensators can be carried out as further research work.

Publications out of the Research Work

[P1] G. Panda, D.P. Das, **S.K. Mishra**, S.K. Panda, S. Meher and K.K. Mahapatra, "Development of an Efficient Intelligent Pressure Sensor using RBF based ANN", *Proceedings of International Conference on Intelligent Signal Processing & Robotics (ISPR-2004)*, (ITS2 in CD), 20th-23rd Feb, 2004, IIIT, Allahabad.

[P2] **S.K. Mishra**, G. Panda, D.P. Das, S.K. Pattanaik and M.R. Meher, "Performance Evaluation of Fixed-point Inverse FLANN model of Intelligent Pressure Sensors," *Proceedings of NCC-2005*, pp.647-650, 28th-30th Jan. 2005, IIT, Kharagpur.

[P3] **S.K. Mishra**, G. Panda, D.P. Das, S.K. Pattanaik and M.R. Meher, "A Novel Method of Designing LVDT using Artificial Neural Network," *Proceedings of IEEE, ICISIP-2005*, pp.223-227, Jan, 2005.

[P4] **S.K.Mishra** and G.Panda, "A New Methodology for Design of LVDT and Comparison with Conventional Design",(paper ID: 520-133) accepted in *IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2006),* held on 15th Feb, 2006, in Innsbruck, Austria.

[P5] **S.K. Mishra** and G. Panda, "A Novel Method for Designing LVDT and its Comparison with Conventional Design",(paper ID: 6014) accepted in *IEEE Sensors Applications Symposium (SAS-2006)* held on Houston Marriott West Loop by the Galleria, Houston, Texas, February 7-9, 2006.

[P6] **S.K.Mishra**, G.Panda, T.K. Dan and S.K. Pattanaik, "Artificial Neural Network based Nonlinearity Estimation of Control Valve Actuators", *Proceeding of National Conference on Soft Computing Techniques for Engineering Applications (SCT2006)*, pp.223-232, 24th-26th Mar, 2006, at NIT, Rourkela.

[P7] G. Panda, **S.K. Mishra**, and D.P. Das, "A novel method of extending the linearity range of LVDT using artificial neural network" (paper ID: SNA-D-05-00679) submitted (under final revision) to *Sensors & Actuators: A. Physical, Elsevier*, on Nov, 2005.

[P8] G.Panda and **S.K.Mishra**, "Development of an Intelligent LVDT using CFLANN Structure", (paper ID: IM-8409) communicated to *IEEE Transaction on Instrumentation & Measurement*, on Jan, 2006.

[P9] G.Panda and **S.K.Mishra**, "Performance Evaluation of Fixed-point FLANN model of Intelligent Pressure Sensors", Communicated to *IETE Journal of Research*, on Oct, 2005.