Adaptive Network based Fuzzy Inference System (ANFIS) as a Tool for System Identification with Special Emphasis on Training Data Minimization

MRINAL BURAGOHAIN

Adaptive Network based Fuzzy Inference System (ANFIS) as a Tool for System Identification with Special Emphasis on Training Data Minimization

A

Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

by

MRINAL BURAGOHAIN

Department of Electronics and Communication Engineering

Indian Institute of Technology Guwahati

Guwahati - 781 039, INDIA

July, 2008

TH-801_02610204

Adaptive Network Based Fuzzy Inference System (ANFIS) as a Tool for System Identification with Special Emphasis on Training Data Minimization

A

Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

by

MRINAL BURAGOHAIN

Department of Electronics and Communication Engineering
Indian Institute of Technology Guwahati
Guwahati - 781 039, INDIA

July, 2008

TH-801_02610204

Certificate

This is to certify that the thesis entitled "**Adaptive Network based Fuzzy Inference System (ANFIS) as a Tool for System Identification with Special Emphasis on Training Data Minimization**", submitted by Mrinal Buragohain, a research scholar in the *Department of Electronics and Communication Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the Institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:
Guwahati

Dr. Chitralekha Mahanta
Associate Professor
Dept. of Electronics and Communication Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039
India

TH-801_02610204

THIS WORK IS DEDICATED TO

MY PARENTS

TH-801_02610204

Acknowledgements

I feel it is a great privilege to express my deepest and most sincere gratitude to my supervisor, Dr. Chitralekha Mahanta for her suggestions, constant encouragement and support during the course of the thesis work. I am also grateful to the other members of my doctoral committee, namely Prof. P.K. Bora, Prof. S. Majhi and Dr. J. S. Sahambi for their valuable comments on my work. I take this opportunity to thank Prof. A. K. Gogoi and Prof. S. Dandapat, the former heads of the department and the present HOD, ECE for their kindness in allowing me to use various computing facilities of the department. I would also like to thank all the other faculty members of the department for their encouragement and help.

I am also grateful to all the members of the Research and technical staff of the department, namely Jharana Rani Rabha, L. N. Sharma, Pranab Kr. Goswami, Sanjib Das, Sidananda Sonowal, Usha Bharali and Utpal Kumar Sharma without whose help I could not have completed this assignment. I thank all my fellow research scholars and M. Tech students for their cooperation. My thanks also go out to all my friends, namely Ahmad Ali, Amrita Ganguly, Atanu Kr Dutta, D. Senthil, Diganta Kr Gogoi, Manas Kamal Bhuyan, P. Vinod, Prabin Kr Padhy, S. K. Mahesh, Senthil Raja, Utkal Mehta and others who have rendered their help at different times which made my stay in this institute a memorable period of my life.

I also acknowledge the endurance and perseverance of my wife Meghali for the last few years without whose sacrifice, love, encouragement and assistance, I would not have been able to complete my research.

Finally, I also express my heartfelt gratitude to all my family members for their unbounded love, encouragement and assistance throughout. I am deeply indebted to them for their countless sacrifices and support without which my research work would not have finished.

(Mrinal Buragohain)

TH-801_02610204

Abstract

Nearly two decades back nonlinear system identification consisted of several ad-hoc approaches which were restricted to a very limited class of systems. However, with the advent of the various soft computing methodologies like neural networks, the fuzzy logic and the genetic algorithm combined with modern structure optimization techniques, a wider class of systems can be handled at present. Complex systems may be of diverse characteristics and nature. These systems may be linear or nonlinear, continuous or discrete, time varying or time invariant, static or dynamic, short term or long term, central or distributed, predictable or unpredictable, ill or well defined. System outputs may be measurable or unmeasurable. Models of real systems are of fundamental importance in virtually all disciplines and hence there is a strong demand for advanced modeling, identification and controlling schemes. This is because models help in system analysis which in turn help to get a better understanding of the system for predicting or simulating a system's behavior. Also, system models facilitate application and validation of advanced techniques for controller design. Development of new processes and analysis of the existing ones along with their optimization, supervision, fault detection, and component diagnosis are all based on the models of the systems. As most of the real world systems are nonlinear in nature, an endeavor is made for modeling a nonlinear system in the present work. A linear system is considered to be a special case of the nonlinear system. The challenges involved in modeling, identification and control of a nonlinear system are too many and attempt has been made to tackle them by applying various soft computing methodologies. In most of the conventional soft computing methods the system modelling results are dependent on the number of training data used. It has been found that the modeling results improve as the number of training data increases. But in many complex systems the number of available training data are less and the generation of new data is also not cost effective. In such a scenario the system has to be modelled with the available data. The proposed modeling scheme has been devised keeping such a possibility in mind. The results obtained by applying this proposed model are compared with the results obtained by using various statistical and genetic algorithm based fuzzy models and finally the relative merits and demerits involved with the respective models are discussed.

The work embodied in the present thesis is concerned with optimal design of the conventionally existing soft computing based system models. The statistics based Full factorial design (FFD) and the V-fold cross validation technique are applied to augment a conventional neuro-fuzzy technique and the following observations are noted :

TH-801_02610204

- The results obtained by applying the proposed technique are comparable and in some cases superior to those obtained by using the conventional neuro-fuzzy model.
- Comparable or superior results are obtained with this proposed model even though the number of data pairs used for system modeling here are less as compared to that used in the conventional methods.
- It resulted in reduction of the number of computations involved. As the experiments were performed by using reduced number of specifically chosen data, the number of computations required to be performed also came down.

TH-801_02610204

Contents

List of Figures iii

List of Tables v

Nomenclature vi

Mathematical Notations viii

1 Introduction 1
- 1.1 Aim of the research . . . 3
- 1.2 Contribution of the thesis . . . 3
- 1.3 Organization of the thesis . . . 4

2 Soft Computing based Techniques for System Identification 6
- 2.1 Neural Network based Algorithms . . . 6
- 2.2 Fuzzy Logic based Algorithms . . . 8
  - 2.2.1 Analysis with Fuzzy Inference System . . . 9
  - 2.2.2 Types of Fuzzy System . . . 10
- 2.3 Genetic Algorithm . . . 12
- 2.4 Neuro Fuzzy Approach . . . 15
  - 2.4.1 Adaptive Network based Fuzzy Inference System . . . 18
- 2.5 Forecasting Methodologies . . . 19
- 2.6 Research gap . . . 22

3 Research Objectives and Methodology 24
- 3.1 Objectives of the present work . . . 24
- 3.2 Methodology of the present work . . . 25
  - 3.2.1 ANFIS based methodology for modeling . . . 25
  - 3.2.2 Full factorial design based methodology for optimal data selection . . . 26
  - 3.2.3 V-fold based methodology for optimal data selection . . . 26
  - 3.2.4 Full factorial design combined with V-fold based methodology for optimal data selection . . . 27
  - 3.2.5 Model building with statistical approaches for performance study . . . 27
  - 3.2.6 Genetic algorithm based fuzzy methodology . . . 27
  - 3.2.7 Controller with the FFD-V-fold based methodology . . . 27

TH-801_02610204

4 Adaptive Network based Fuzzy Inference System **28**
4.1 ANFIS structure 28
4.2 Learning Algorithm 30
4.3 Derivation of the Initial Fuzzy Model 31
4.3.1 Extracting the initial fuzzy model 31
4.3.2 Selection of input variables and final fuzzy model 35
4.3.3 Optimization of the initial fuzzy model 37
4.4 Experimental Data 37
4.5 Experimental Results and Discussion 38
4.5.1 ANFIS Model for Gas Furnace 39
4.5.2 ANFIS Model for Thermal Power Plant 41
4.6 Conclusions 51

5 Optimal Data based ANFIS Modeling **54**
5.1 Full Factorial Design Based ANFIS Modeling 54
5.2 V-Fold Technique Based ANFIS Modelling 58
5.3 FFD-V-fold Based ANFIS Modeling 59
5.4 Experimental Results and Discussion 59
5.4.1 Modeling results with Full factorial design based ANFIS model 60
5.4.2 V-fold technique based ANFIS model 67
5.4.3 Modeling results with FFD-V-fold technique based ANFIS model 68
5.5 Conclusion 75

6 Statistical Models and Genetic Algorithm Based Fuzzy Model **76**
6.1 Statistical Models 76
6.1.1 Auto Regressive with Exogenous Input (ARX) Model 77
6.1.2 Auto Regressive Moving Average with Exogenous Input (ARMAX) Model . . 77
6.1.3 Auto Regressive Integrated Moving Average with Exogenous Input (ARIMAX) Model 77
6.2 Genetic Algorithm Based Fuzzy Model 78
6.3 Experimental Results and Discussion 78
6.3.1 Statistical models 79
6.3.2 GA based fuzzy model 82
6.4 Controller Using FFD-V-fold based ANFIS Model 88
6.4.1 Controller Design 90
6.4.2 Experimental Results and Discussion 92
6.5 Conclusion 93

7 Conclusions and Future Work **98**
7.1 Summary of the thesis 98
7.2 Discussion 99
7.3 Conclusion 100
7.4 Future Work 101

A Sample Power Plant Data **103**

Bibliography **110**

TH-801_02610204

List of Figures

2.1 Crossover operation . . . 14
2.2 Mutation operation . . . 14
2.3 GA algorithm flowchart . . . 15

4.1 Type-3 ANFIS Structure . . . 29
4.2 Grid partitioned fuzzy subspaces for a type-3, 2-input ANFIS . . . 35
4.3 Variable Selection Process for a four input initial model . . . 36
4.4 Rules for the grid partition based ANFIS model when trained with entire data set for the gas furnace example . . . 40
4.5 Rules for the subtractive clustering based ANFIS model when trained with entire data set for the gas furnace example . . . 40
4.6 Actual and ANFIS model predicted output with Box and Jenkins gas furnace data . . . 41
4.7 ANFIS model prediction error for Box and Jenkins gas furnace data . . . 43
4.8 RMSE of the ANFIS model during training for Box and Jenkins gas furnace data . . . 44
4.9 Output surface of the data for Box and Jenkins gas furnace example . . . 44
4.10 Rules for the ANFIS model when trained with half of the data set for the gas furnace example . . . 45
4.11 Actual and ANFIS model predicted output with Box and Jenkins gas furnace data . . . 46
4.12 Rules for the ANFIS model when trained with half of the data set-I for the thermal power plant example . . . 47
4.13 Actual and ANFIS model predicted output with the data set-I for thermal power plant . 47
4.14 Rules for the ANFIS model when trained with half of the data set-II for the thermal power plant example . . . 48
4.15 Actual and ANFIS model predicted output with the data set-II for thermal power plant 48
4.16 Rules for the ANFIS model when trained with half of the data set-III for the thermal power plant example . . . 49
4.17 Actual and ANFIS model predicted output with the data set-III for thermal power plant 49
4.18 Rules for the ANFIS model when trained with half of the data set-IV for the thermal power plant example . . . 50
4.19 Actual and ANFIS model predicted output with the data set-IV for thermal power plant 50
4.20 Rules for the ANFIS model when trained with half of the data set-V for the thermal power plant example . . . 51
4.21 Actual and ANFIS model predicted output with the data set-V for thermal power plant 52

5.1 Selection of training data in a 2^3 factorial design . . . 56
5.2 Flow Chart for the FFD-V-fold Technique . . . 60

5.3 Actual and FFD-V-fold based ANFIS model predicted output with the Box and Jenkins gas furnace data . . . 72
5.4 Actual and FFD-V-fold based ANFIS model predicted output with data set-I of thermal power plant . . . 72
5.5 Actual and FFD-V-fold based ANFIS model predicted output with data set-II of thermal power plant . . . 73
5.6 Actual and FFD-V-fold based ANFIS model predicted output with data set-III of thermal power plant . . . 73
5.7 Actual and FFD-V-fold based ANFIS model predicted output with data set-IV of thermal power plant . . . 74
5.8 Actual and FFD-V-fold based ANFIS model predicted output with data set-V of thermal power plant . . . 74

6.1 Flow Chart for the GA based Fuzzy model . . . 79
6.2 Actual and ARIMAX model predicted output with Box and Jenkins gas furnace data . . . 81
6.3 Actual and ARIMAX model predicted output with data set-I of thermal power plant . . . 81
6.4 Actual and ARIMAX model predicted output with data set-II of thermal power plant . . . 82
6.5 Actual and ARIMAX model predicted output with data set-III of thermal power plant . . . 83
6.6 Actual and ARIMAX model predicted output with data set-IV of thermal power plant . . . 83
6.7 Actual and ARIMAX model predicted output with data set-V of thermal power plant . . . 84
6.8 Actual and GA based fuzzy model predicted output with Box and Jenkins gas furnace data . . . 85
6.9 Actual and GA based fuzzy model predicted output with data set-I of thermal power plant 85
6.10 Actual and GA based fuzzy model predicted output with data set-II of thermal power plant . . . 86
6.11 Actual and GA based fuzzy model predicted output with data set-III of thermal power plant . . . 87
6.12 Actual and GA based fuzzy model predicted output with data set-IV of thermal power plant . . . 87
6.13 Actual and GA based fuzzy model predicted output with data set-V of thermal power plant . . . 88
6.14 Training phase of the inverse controller . . . 90
6.15 Application phase of the inverse controller . . . 90
6.16 Inverse controller with fuzzy integrator block . . . 91
6.17 Desired output and the actual output of the plant with the controller . . . 93
6.18 Prediction error of the controller . . . 94
6.19 Output of the controller . . . 94
6.20 Desired and actual output of the plant with controller in presence of disturbance . . . 95
6.21 Desired and actual output of the plant with controller having plant parameter variations 95
6.22 Prediction error of the controller with plant parameter variations . . . 96
6.23 Output of the controller with plant parameter variations . . . 96
6.24 Desired and actual output of the controller based plant with plant parameter variations along with disturbance . . . 97

TH-801_02610204

List of Tables

4.1 Comparison of various models derived for the Box and Jenkins gas furnace data. The first 7 rows are excerpted from a table in [76] 42
4.2 Modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using ANFIS 52

5.1 Modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using FFD based ANFIS 66
5.2 Modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using V-fold based ANFIS model 68
5.3 Modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using FFD-V-fold based ANFIS 70
5.4 Comparison of modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using various ANFIS models 71

6.1 Results obtained for Statistical models using the Box and Jenkins gas furnace and the thermal power plant data 80
6.2 Results obtained for the GA based fuzzy model using the Box and Jenkins gas furnace and the thermal power plant data 89
6.3 Comparison of the FFD-V-fold based ANFIS, GA based fuzzy and the statistical ARIMAX model obtained for the Box and Jenkins gas furnace and the thermal power plant data 89

A.1 Thermal power plant data 104

TH-801_02610204

Nomenclature

AI	Artificial Intelligence
ANFIS	Adaptive network based fuzzy inference system
ANN	Artificial neural network
AR	Auto regressive
ARIMAX	Auto regressive integrated moving average with exogenous input
ARMAX	Auto regressive moving average with exogenous input
ARX	Auto regressive with exogenous input
BD	Bad data
BP	Back propagation
DAS	Data acquisition system
DDEC	Data driven echo canceller
EOF	Effect of factor
FFD	Full factorial design
FIS	Fuzzy inference system
FL	Fuzzy logic
FLC	Fuzzy logic controller
GA	Genetic algorithm

LP	Linear Programming
LSE	Least square estimation
MA	Moving average
MIMO	Multi input multi output
MLP	Multi layer perceptron
MSE	Mean square error
NC	Neuro computing
RBFN	Radial basis function network
RHONN	Recurrent high order neural network
RMEEI	Recursive measurement error estimation identification
RMSE	Root mean square error
SISO	Single input single output
SMT	Statistical machine translation
SOFC	Self organizing fuzzy controller

Indian Institute of Technology Guwahati

Mathematical Notations

μ Membership value

r_a Radius of the cluster

r_b Radius defining the neighborhood of the cluster

α Constant whose value depends on r_a

β Constant whose value depends on r_b

Chapter 1

Introduction

Most of the present day systems are large and may be considered to be complex in nature. Electrical power, chemical, water treatment and similar large-scale industrial plants are all complex in nature. Complex systems may be linear or nonlinear, continuous or discrete, time varying or time invariant, static or dynamic, short term or long term, central or distributed, predictable or unpredictable, ill or well defined. Also, system outputs may be measurable or unmeasurable. They may consist of many interconnected systems, sub-processes or components. The processes involved in the complex systems may possess widely varying properties. In large scale systems, every part performs a desired function and the overall system works satisfactorily only if all the different parts work in tandem for what they are designed for. Modeling of complex systems is of fundamental importance in almost all fields. This is because models facilitate better understanding of the system and so help in system analysis. So prediction and simulation of the system's behavior are then possible. System model also helps to design new processes and analyze the existing ones. The design, optimization and supervision of controllers, fault detection and faulty component diagnosis are all based on the system model. This is because for the improvement of the system's performance, it is required to model the system correctly so that the model parameters can be tuned to get the required system response. It is because of this fact that in the last few decades, modeling of large scale, complex systems has been a special topic of interest among the researchers of various disciplines worldwide [1]. Most of the real world systems are ill defined in nature and hence difficult to model. Generally the performance of the system is dependent on the accuracy of the model. Therefore it is of utmost importance to build a model which correctly reflects the behavior of the system under consideration. The functioning of complex large-scale systems also involves numerous tradeoff problems like cost and accuracy [2]. Hence, there is a strong demand for developing advanced methods of system modeling and identification techniques. The conventional

TH-801_02610204

methods that have been used for system modeling rely heavily on the mathematical tools which require precise knowledge about the involved physical processes. In systems where the mathematical model is not available, it is not possible to use the conventional methods for its analysis. In such cases, soft computing based modeling [3] approaches provide a viable alternative for identification of the system from the available data. The concept of soft computing [4] began to materialize near about the time when Lotfi Zadeh was working on soft analysis of data and fuzzy logic. This gave birth to the intelligent systems. Nearly four decades later, the intelligent system became a reality. However, initially the technology needed for building systems that possess Artificial intelligence (AI) was not available. Instead only predicate logic and symbol manipulation techniques formed the core of the traditional AI. These techniques could not be used for building machines which could be called intelligent from the point of view of real world application. But today the requisite hardware, software and sensor technology are available for building intelligent systems. In addition to these, computational tools are available now which are far more effective for conception and design of intelligent systems. These tools are derived from a collection of methodologies called soft computing. Unlike hard computing the essence of soft computing is aimed at accommodating the prevalent imprecision of the real world. Therefore soft computing helps in exploiting the tolerance for imprecision, uncertainty and partial truth so that tractability, robustness, low solution cost and better rapport with reality can be achieved. Hence the human mind can be considered to be a role model for soft computing. Rather than a single technique, soft computing may be considered to be comprising of different methodologies with Neuro-computing (NC), the Fuzzy logic (FL) and the Genetic algorithm (GA) as the principal partners. Therefore in soft computing based system identification, instead of a single standard method, a collection of techniques has been put forward as possible solutions to the identification problem. They can be broadly grouped as neural network based algorithm, fuzzy logic based algorithm and the genetic algorithm. The neural network has the inherent advantage of being able to adapt itself and also in its learning capabilities. Similarly the salient feature that is associated with the fuzzy logic is the distinct ability to take into account the prevailing uncertainty and imprecision of real systems with the help of the fuzzy if-then rules. In order to exploit the advantage of the self adaptability and learning capability of the neural network and the capability of the fuzzy system to take into account of the prevailing uncertainty and imprecision of real systems with the help of the fuzzy if-then rules, an integrated forecasting approach comprising of both the fuzzy logic and the neural network has been considered. This hybrid system is called the Adaptive network based fuzzy inference system (ANFIS). Here the fuzzy system with its expert knowledge stands as a front end preprocessor for the neural network input and output layers. Based

TH-801_02610204

on the historical data, the neural network learning algorithms are used to determine the parameters of the expert knowledge based fuzzy system. The use of this hybrid system ANFIS helps to complement the weakness of the respective systems.

1.1 Aim of the research

The main research objective that has been kept in mind while initiating and furthering the present work is the modeling of a real world system with the help of the soft computing technique. In the quest for developing a model for a system based on its available input output data, it has been observed that in the conventional modeling approach the results depend on the mathematical model of the system and its accuracy. In cases where the mathematical model is not available the system analysis becomes very difficult. It is in this context that the soft computing approach can provide a viable alternative. The prime inherent advantage associated with the soft computing techniques of not requiring a mathematical model has been a motivating factor for consideration in our present work. Motivated by this advantageous feature of soft computing based system identification, the present work focuses on building a model for an ill defined real world system based on its available record of input-output data using ANFIS. An endeavor is made to extract previously unknown information from the available time-series data so that an accurate model can be built. Once built, the model can be used to predict or forecast future values. The present research concentrates on the difficult situation when the available input-output data for a system are very less and generation of real time data is also prohibitively expensive. In such cases, it is really a challenging task to build a faithful model for the system using soft computing based data driven identification technique like ANFIS. This research attempts to find a solution to this problem of modeling an ill defined real world system faithfully in the situation when available input-output data is scanty.

1.2 Contribution of the thesis

The major contributions of this thesis can be outlined as follows:

- Building an ANFIS model for real world systems like gas furnace and thermal power plant

- Proposal of a novel technique for training the neural network in the ANFIS model by optimally

TH-801_02610204

selecting the available data using the Full factorial design (FFD) and the V-fold cross validation method

- Proposal of a hybrid method for optimal data selection incorporating both the above techniques of full factorial design and V-fold cross validation
- Proposal of a GA based fuzzy model
- Design of an ANFIS based controller to study the performance of the proposed ANFIS model

1.3 Organization of the thesis

The thesis records a detailed account of the use of the soft computing techniques for system identification. The organization of the thesis is given below.

Chapter 2: A brief overview of the system identification techniques based on soft computing methodology is presented in this chapter. Available literature about model based forecasting techniques is also reviewed in this chapter.

Chapter 3: In this chapter the objectives and methodology used in the research are discussed.

Chapter 4: In this chapter the ANFIS architecture is discussed in details. This chapter is devoted to the modeling of systems based on ANFIS.

Chapter 5: In this chapter the use of two new techniques proposed in the present research for optimal data selection in training the ANFIS model is explained. These two methods are the Full factorial design (FFD) and the V-fold technique by using which the number of data required for training the ANFIS was drastically reduced. Also another hybrid model of ANFIS combining these two techniques has been proposed.

Chapter 6: In this chapter the proposed GA based fuzzy model for system identification is discussed. This chapter also gives a detailed account of the statistical models namely ARX, ARMAX and ARIMAX which are applied to the system under consideration. The results obtained from the proposed soft computing based model are compared with those obtained by using these conventional statistical models as well as GA based models.

TH-801_02610204

Chapter 7: The last chapter presents a brief discussion about the research work and also draws conclusions by analyzing the results obtained in this thesis. Also a few areas for further investigation are suggested in this chapter.

TH-801_02610204

Chapter 2

Soft Computing based Techniques for System Identification

Contrary to the conventional hard computing techniques, the prime inherent advantage associated with the soft computing techniques is the non requirement of a mathematical model and hence are becoming increasingly popular as system identification methodology. Three powerful soft computing techniques which are very popular are the Neural network, the Fuzzy logic and the Genetic algorithm (GA). A brief overview of available methods using these techniques for identification and control of linear as well as nonlinear dynamical systems is presented in this chapter. These soft computing based approaches are reviewed thoroughly in this chapter since this research focuses on utilizing soft computing as a tool for system identification for the purpose of modeling an ill defined real world system so as to forecast future values.

2.1 Neural Network based Algorithms

In addition to being the source of natural intelligence, the human brain can process incomplete information obtained by perception at a very rapid rate. Inspired by this biological property of the nervous systems and the brain, researchers attempted to model the human brain resulting in the evolution of the neural network. Here the brain has been modeled as a continuous time nonlinear dynamic system with a connection architecture. In this architecture the neurons or the processing units which are interconnected by weights are expected to mimic the human brain. This gives the neural network the capability for learning and adaptation by adjusting the interconnection between the layers. The most important

TH-801_02610204

characteristics of the neural network are:

- Presence of a large number of simple units
- Presence of a large number of highly parallel units
- Presence of strongly connected units
- Robustness against the failure of single units
- Learning from data

The network of any system will be considered to be an artificial neural network if the same basis function is used throughout the network. Here the nodes of the system are called the neurons. The layers of a neural network architecture may be subdivided into three principal groups:

- Depending upon the number of inputs all the input neurons together constitute the input layer
- Similarly all the output neurons together constitute the output layer
- All the neurons in any intermediate layer form the hidden layer. There may be more than one hidden layer in a neural network

Some of the popular neural network architectures are the Multi-layer perceptron (MLP) network, Radial basis function network (RBFN) and the neuro-fuzzy network. The pivotal contribution of neural networks is a methodology for identification, learning and adaptation.
The early works on neural networks include those of McCulloch and Pitts [5], Hebb [6], Rosenblatt [7], Widrow and Hoff [8], Minsky and Papert [9], Hopfield [10], Parker [11], Rumelhart and McClelland [12], Carpenter and Grossberg [13] and Kohonen [14]. Narendra and Parthasarathy [15] demonstrated the use of neural network for the identification and control of nonlinear dynamical systems. Polycarpo and Ioannou [16] proposed the general formulation for modelling, identification and control of a nonlinear dynamical system. Another development was the design and analysis based on Lyapunov stability theory [16, 17]. Pham and Liu [18] proposed the use of recurrent neural networks for the identification of linear and nonlinear dynamic systems. The interest in the use of neural networks for modeling and identification of static and dynamical complex system on the basis of the input-output data pairs was a new development. Kosmatopoulos and Christodoulou [19] proposed an algorithm for identification of nonlinear systems using Recurrent high order neural network (RHONN) based on the

TH-801_02610204

extension of Hopfield [20] and Cohen-Grossberg [21] works. Neural networks with Radial basis function (RBF) are used due to their excellent classification property. Sanner and Slotine [17] presented an approach with a Gaussian radial-basis function adaptive dynamical system with unknown nonlinearities. Hong and Xinkuo [22] proposed a neural network approach by combining the equivalence between RBF and the Fuzzy inference system (FIS) for identification of a nonlinear system. Ahmad et al. [23] presented a nonlinear Multi-input-multi-output (MIMO) system identification scheme which is based on the radial basis function network. Similarly Selmic and Lewis [24] presented a multi model identification scheme by using nonlinear system identification technique with the RBF based neural network. Azam and Valandham [25] presented a RBF based neural network which uses the log-sigmoid as the basis function for identification purpose. This function eliminates the risk of mathematical instabilities which are found while using Gaussian radial basis function based networks. Robert et al. [26] proposed a class of additive dynamic connectionist model for the identification of unknown dynamic systems with the help of two online parameter adaptation algorithms. One of these algorithms is based on gradient descent [27–29] technique and sensitivity analysis while the other is based on the variational calculus. To deal with the problem of time variation of disturbance and system parameters, Song and Soh [30] proposed an adaptive and robust identification algorithm. This method can overcome the disturbance problem by the selection of a robust adaptive dead zone scheme. The main drawback of the feed forward Multilayered perceptron (MLP) with Back-propagation (BP) algorithm is the requirement of intensive computation and the slow rate of error convergence. To remove this drawback, Patra and Chen [31, 32] proposed the Chebyschev functional link artificial neural network (C-FLANN) comprising of the Chebyschev polynomials. This method was found to have superior performance as compared to the MLP requiring less computation for the task of nonlinear system identification. Ren et al. [33] proposed an algorithm for the online identification and control of a class of continuous time higher order nonlinear system using dynamic neural network.

2.2 Fuzzy Logic based Algorithms

The pivotal contribution of fuzzy logic is a methodology for computing with words which can deal with imprecision and granularity. The human brain can interpret and process imprecise and incomplete sensor information which are received from the perceptive organs. Analogously the fuzzy set theory can also provide a systematic approach to deal with such information linguistically. It can also perform numerical computation by using membership function for the stipulated linguistic labels. The Fuzzy

TH-801_02610204

inference system (FIS) is based on the concepts of fuzzy set theory, fuzzy if-then rules and fuzzy reasoning. The framing of the fuzzy if-then rules forms the key component in FIS. FIS is a very popular technique and has been widely applied in different fields like data classification, automatic control, expert system, decision making, robotics, time series analysis, pattern classification, system identification etc. The basic structure of a fuzzy inference system consists of three principal components viz a rule base comprising of the selected fuzzy rules, a database defining the membership functions of the fuzzy rules, and a reasoning mechanism which performs a fuzzy reasoning inference with respect to the rules so as to derive a reasonable output or conclusion.

2.2.1 Analysis with Fuzzy Inference System

For the analysis of a fuzzy system whose inputs and outputs are described by linguistic variables, the following steps have to be carried out:

- Fuzzification :-The linguistic variables of the fuzzy rules are expressed in the form of fuzzy sets where these variables are defined in terms of degree of their associated membership functions. This method of calculating the degree of belongingness of the crisp input in the fuzzy set is called the fuzzification. The membership functions may be triangular, trapezoidal, gaussian or bell shaped. As the information about the degree of the membership is used for further processing, considerable amount of information may be lost during the course of fuzzification. This is because the procedure can be seen as a nonlinear transformation of the inputs. For example in the case of triangular or trapezoidal membership functions information is lost in the regions of membership functions where the slope is zero, as at these points the membership functions are not differentiable. Therefore fuzzy systems having triangular or trapezoidal membership function can encounter problems of learning from data. Smoother membership functions like gaussian or bell function may be used to overcome this difficulty.

- Aggregation :-After the degree of each linguistic statement is evaluated, they are combined by logical operators such as AND and OR. The conjunction of these linguistic statements is carried out by logical t-norm and the t-conorm operator to a large number of linguistic statements. Max and Min operators are used for classification task. For the purpose of approximation and identification the product and algebraic product operators are better suited due to their smoothness and differentiability. Similarly the bounded sum and difference operators offer several advantages to some neuro-fuzzy learning schemes.

TH-801_02610204

- Activation :-Here the degree of rule fulfilment is used to calculate the output activations of the rules.

- Accumulation :-In this step the output activations of all the rules are joined together to give rise to the fuzzy output of the system.

- Defuzzification :-If a crisp value of the system is required, the final fuzzy output has to be defuzzified. This can be done by different methods like center of gravity, bisector of area, mean of maximum (mom), smallest (absolute) of maximum (som) and largest (absolute) of maximum (lom).

2.2.2 Types of Fuzzy System

A fuzzy system may be of three principal types, namely:

- Mamdani fuzzy system :-This type of system is also known as the linguistic fuzzy system.

- Singleton Fuzzy system :-The complexity of defuzzification of a linguistic fuzzy system can be simplified by restricting the output to a singleton membership function. Since no integration has to be carried out numerically, this results in reducing the computational demand for the evaluation and learning of the fuzzy system. Therefore a singleton fuzzy system is most widely applied in industry.

- Takagi-Sugeno Fuzzy system :- This system may be considered to be an extension of the singleton fuzzy system. Here the function f is not a fuzzy set. But the premise of a Takagi-Sugeno fuzzy system [34,35] is linguistically interpretable. For a dynamic process modelling the Takagi-Sugeno models possess an excellent interpretation. A singleton fuzzy system can be recovered from a Takagi-Sugeno fuzzy system if the function f is chosen to be a constant. As the constant can be seen as a zeroth order Taylor series expansion of the function f, it is also called the zeroth order Takagi-Sugeno fuzzy system. However, in most of the applications, the first order Takagi-Sugeno fuzzy system is more common.

Since the introduction of the fuzzy logic concept by Zadeh [36, 37], research was continuing for application of fuzzy system theory for system identification [38–40]. This is because in many complex and ill-defined systems where precise mathematical models are difficult to build, their fuzzy models can be obtained easily which reflect the uncertainty of the system in a proper way. There are numerous

TH-801_02610204

applications of Fuzzy logic controller (FLC) in industrial processes [41–43]. The interpolation of the fuzzy rule base provided by human experts governs the performance of a FLC. Self-organizing fuzzy controller (SOFC) was proposed by Mamdani and Procyk [44] where the fuzzy rule base is modified by evaluating the system performance that is expressed linguistically. Park and Lee [45] proposed an SOFC where the rules are generated by input-output data. Here the rules get updated by self learning procedure. Graham and Newell [46] presented another approach where the fuzzy model of the process is identified online and the control input is calculated based on the identified model of the process and a heuristically determined performance measure. Here the system is considered to be linear. It has been observed that a lot of researchers are paying attention to the fuzzy relational model. This is because the fuzzy relational equations reflect the fuzzy nature of the system effectively. These relations are universal approximators which can perform nonlinear mappings between input-output variables which are treated as fuzzy sets. There are two aspects of fuzzy relational equation method. One aspect is based on the resolution of composite fuzzy relational equation [47, 48] and the other is the linguistic model identification as proposed by Tong [49]. But as Tong's model could not be extended to a higher dimensional system, Li and Liu [50] proposed an adaptive mechanism based on the decision table form of simple linguistic models. Pedrycz [48] proposed another fuzzy compositional rule based system identification algorithm which uses the concept of referential fuzzy set and Zadeh's conditional possibility distribution. Babu and Sachidanand [51] presented another method for the identification of a nonlinear system based on the fuzzy relational model. Here the input is considered to be linear and the output is considered to be nonlinear in nature. Lee et. al. [52] proposed a combined approach to fuzzy model identification which combines the linguistic approach and the numerical resolution of fuzzy relational equation. Moore and Harris [53] proposed an indirect adaptive fuzzy control method which uses a first order fuzzy model for the identification of the plant. Xu [54] showed the application of the fuzzy system for the identification of a nonlinear system. Up till now, there are three different kinds of fuzzy models which have been in use from the fuzzy control viewpoint. In the first class the fuzzy basis function approximation [55] has been used, which may be considered as a mapping between the input and the output space. This model suffers from the disadvantage that some important dynamical behavior of the system can not be represented. In the second type of fuzzy model as described by Takana and Sugeno [56], global function approximation can be achieved from a set of local linear equations. But this model suffers from the disadvantage that it cannot be used for controller design. The third model is called the fuzzy dynamic model [57] which requires accurate determination of the upper bound of the local model. Huaguang and Yongbing [58] proposed a fuzzy hyperbolic model for

TH-801_02610204

a class of complex systems which is difficult to model. Sugeno and Yasukawa [59] presented a general approach for qualitative modeling based on fuzzy logic. Cipriano and Montoya [60] proposed a fuzzy model for the identification of nonlinear systems. Similarly Gaweda and Zurada [61] presented a fuzzy identification system with relational input partition. Saez and Cipriano [62] proposed the application of fuzzy models for the control of a combined cycle power plant boiler. Elshafei and Karray [63] proposed a fuzzy model for the identification of a class of black-box type nonlinear systems. Similarly Saez and Cipriano [64] proposed a fuzzy model for the representation of the nonlinearity of a process. Again Chen and Linkens [65] proposed a method for the fuzzy modeling from numerical data. Flores et al. [66] applied fuzzy logic for controlling a solar power plant.

The fuzzy if-then rules contain the structured knowledge representation of the fuzzy inference system. But this does not provide the adaptive capability to the fuzzy inference system for dealing with the changing external environment which is found in a neural network.

2.3 Genetic Algorithm

Genetic Algorithm was envisaged by John Holland [67] at the university of Michigan, in the 1970s. It is based on the Darwinian evolution theory of survival of the fittest which states that the fitter and stronger individuals in a population have a higher chance of creating offsprings for the next generation by random mutation and natural selection. It can be implemented as an optimization search procedure which uses the principles of genetics and natural selection by modelling possible solutions to a search problem as strings of zeroes and ones. Each point in a parameter or solution space can be encoded by Genetic algorithm (GA) in a binary bit string, which is called the chromosome. For example, if a particular point $(8, 5, 7)$ in a three dimensional space, is to be transformed, it can be represented by a concatenated binary string of $1000, 0101, 0111$. Here each coordinate is encoded by a gene which is represented by four binary bits. Encoding plays a key role in determining the GA's performance as it helps to translate problem specific knowledge directly to the GA framework. After the creation of a generation comprising of these points, each of these points is also assigned a fitness value which is given by the evaluated value of the objective function at that point. Generally positive values of the objective function are preferred. So if the fitness value is not positive some kind of monotonical scaling and/or translation can be done [68]. The problem can also be sometimes tackled by calculating the fitness value in terms of the ranking of the members in a population. A collection of these points is kept

TH-801_02610204

as a gene pool or population which can be repeatedly evolved towards a better fitness value in the next successive generations, by upgrading the entire population. After the completion of the encoding and the fitness evaluation steps, the GA constructs a new population in each successive generation by using three basic genetic operators. They are:

- Selection:- The selection operation determines the parents which will be participating in mating to produce offsprings for the next generation. This is analogous to the survival of the fittest in the natural selection. Starting from a possible solution strings, pairs of individuals from the current population are allowed to mate to produce offsprings for the next generation. This selection procedure is based on the strategy of survival of the fittest. Generally the selection of members for mating are based on their selection probability which is proportional to their fitness value. Usually the roulette wheel selection strategy is used where depending upon the string's relative fitness size a string is assigned a slot in the simulated wheel. The selection strategy results in the replacement of members with below-average fitness value with members having above-average fitness value, to take part in the mating process.

- Crossover:-The crossover operation is carried out for the generation of new chromosomes which will be able to retain the good features from the previous generation. Selected pairs of parents having a probability equal to a given crossover rate are considered for this operation. In this operation genetic materials (i.e bit-values) between two parent strings are swapped so that generated offsprings represented by highly fit strings can have a greater probability to be selected in the subsequent generation. A single point crossover is the most basic operator. In a single point crossover along a randomly selected bit position the genetic materials of the two parent chromosome strings cross over. Similarly for a two-point crossover, the genetic materials between two randomly selected crossover points in two chromosome strings are interchanged to generate two children. The effect of crossover is analogous to mating in the evolutionary process in which the parents pass segments of their own chromosomes to their children. These children can outperform their parents if they inherit good genes from them. This operation is shown in Fig. 2.1.

- Mutation:- A certain pool of population may not be able to solve a particular problem if it does not contain all the encoded information. To rectify this problem the mutation operator is used so that spontaneously generated new chromosomes can be added to the existing pool. This operation alters a few more selected bit values in randomly selected strings, with a probability equal to a

TH-801_02610204

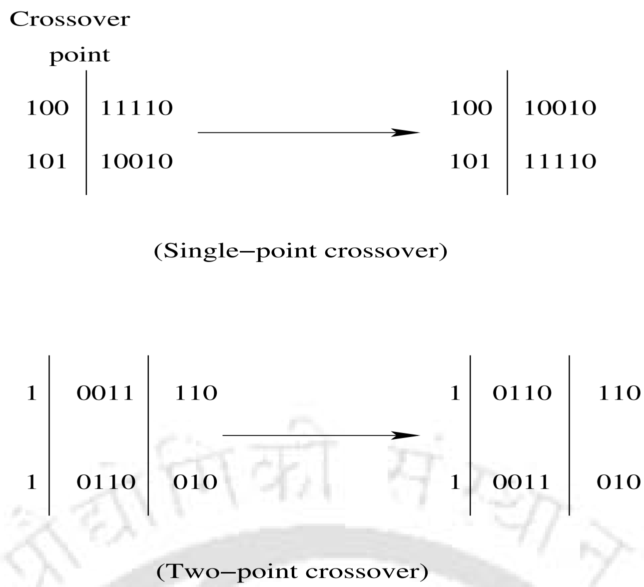

Figure 2.1: Crossover operation

very low mutation rate, after the crossover operation. The mutation rate is kept low so that good chromosomes obtained from crossover operation are not lost. The mutation operator can prevent any single bit from converging to a value throughout the entire population. In addition to this the population is prevented from converging or getting stagnant in any local optima. This operation enhances the ability of the GA to find a solution which is near optimal by searching the entire solution space for the best solution. This operation is shown in Fig. 2.2.

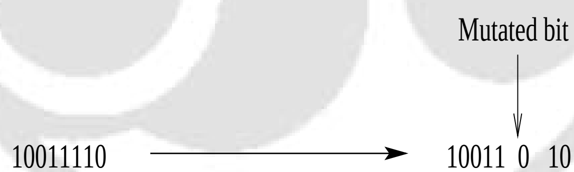

Figure 2.2: Mutation operation

The pivotal contribution of genetic algorithm is a methodology for systematized random search and optimization. Researchers attempt to simulate complex biological evolutionary processes to discover how evolution can propel living beings towards a higher level of intelligence which has resulted in the concept of the genetic algorithm. The flow chart for this algorithm which is based on the evolutionary principle of natural selection has been shown in Fig. 2.3.

Kumon et al. [69] proposed the use of genetic algorithm for nonlinear system identification. Similarly Akramizadeh et al. [70] proposed the use of genetic algorithm for the identification of a nonlinear

TH-801_02610204

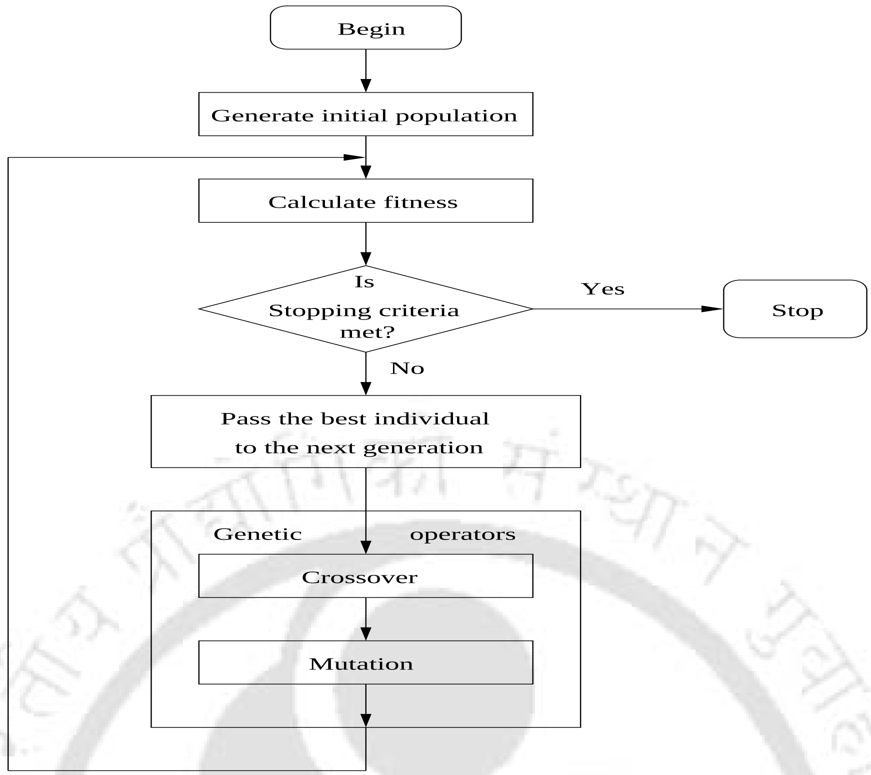

Figure 2.3: GA algorithm flowchart

Hammerstein model. Juang [71] presented an algorithm which was based on recurrent neural network with genetic algorithm for nonlinear system identification. Kim and Lee [72] proposed the application of multi-objective fuzzy optimization for the parameter identification of a nonlinear system. Vazquez et. al. [73] used genetic programming for the structure identification of nonlinear dynamic systems.

2.4 Neuro Fuzzy Approach

Fuzzy logic, neural network and genetic algorithm are complementary rather than competitive for system identification. Therefore it is advantageous to use these techniques in combination amongst themselves rather than exclusively. This gives rise to what is called the hybrid intelligent systems. One of the popular combinations that has been used extensively is the neuro fuzzy hybrid system. The essential part of neuro-fuzzy modelling comes from a common framework called adaptive network which unifies the neural network and the fuzzy model. In this resultant hybrid intelligent system, the neural network has the ability to recognize patterns and adapt themselves to cope with changing environment. On the other hand the fuzzy inference system incorporates human knowledge and performs inferencing and decision making. The modeling by neuro-fuzzy method is concerned with model extraction from

TH-801_02610204

numerical data which represents the dynamic behavior of the system. System modelling based on this methodology can serve two purposes:

- The behavior of the system under consideration can be predicted from the derived model
- The derived model can be used for the design of a controller

The important steps of the neuro-fuzzy modelling approach are:

- Fuzzification of the input physical variables
- Computation of the degree of satisfaction for the available linguistic terms
- Conjunction of the premise and the fuzzy inferred parameters
- Defuzzification of the output

All the above mentioned steps are realized in sequentially arranged layers of the neural network which has an architecture to adjust the weights in the form of the parameters of the extracted rules. Using the neuro-fuzzy modelling as a backbone, the characteristics of soft computing may be classified as follows :-

- Human expertise:- For solving practical problems, soft computing can be used to utilize the human expertise in the form of conventional knowledge representation as well as fuzzy if-then rules.
- Biologically inspired computing models:- Inspired by the biological neurons, the artificial neural network can be used in soft computing methodology to deal with many practical problems like pattern recognition, nonlinear regression, perception and classification.
- Innovative approach:- Soft computing methodology applies innovative optimization techniques imbibed from various sources. They are genetic algorithm as motivated from the evolution and selection process, simulated annealing which is inspired from thermodynamics, the random search method and the simplex downhill method.
- Numerical computation:- Unlike symbolic artificial intelligence, soft computing mainly relies on numerical computation.

TH-801_02610204

- New application domains:- Because of being software based, the soft computing method is increasingly finding applications in new domains like adaptive signal processing, adaptive control, nonlinear system identification etc.

- Model free learning:- In the absence of system models, the fuzzy inference system and the neural network have the ability to construct models from the available sampled data.

- Intensive computation:- Neuro-fuzzy and soft computing methods rely heavily on the high speed number crunching computation to find rules or regularity in data sets, rather than too much background knowledge on the problem being solved.

- Fault tolerance:- The deletion of a neuron in a neural network or a rule in a fuzzy inference system does not stop these models to perform because of their parallel and redundant architecture. However, the performance degrades gradually.

- Goal driven characteristics:- Soft computing and neuro-fuzzy systems are goal oriented. However, a little deviation from their assigned goal will not deter them from achieving their goal finally because of their domain specific knowledge.

- Real world applications:- Because of the complexity and uncertainty involved with the real world problems, the conventional approaches of problem solving require a mathematical model or a detailed description of the problem which is being solved. But with the integrated approach of soft computing, it usually utilizes the specific techniques within the subtasks to constitute a satisfactory solution to the real world problems.

Jang and Sun [74] discussed the problems of neuro-fuzzy modeling and also the direction for its future use. Lin and Cunningham III [75] presented a method where the network was viewed as a fuzzy model which gave insight into the real system and also provided a method to simplify the neural network. Jang [76] proposed a simple method for the selection of inputs for the neuro-fuzzy model in identifying a nonlinear system. Similarly Chiu [77] proposed another method for the selection of inputs of the neuro-fuzzy model built for nonlinear system identification. Denai et al. [78,79] showed the application of the neuro-fuzzy method for the modeling of nonlinear systems. Ishibuchi et al. [80] proposed a method where the neural network is trained by utilizing the numerical data and also human expert knowledge that is represented by the fuzzy if and then rules. Juang and Lin [81] proposed a method for the identification of a dynamic system with the help of a Takagi-Sugeno-Kang (TSK) type fuzzy

TH-801_02610204

rule based model which also possesses the learning ability of the neural network. Sanchez et al. [82] proposed a method for the identification of a nonlinear system using a fast and stable neuro-fuzzy method having error minimization. Li et al. [83] presented a neuro-fuzzy method for the identification of a nonlinear system where in the first step the structure identification task is accomplished and in the next step the parameter identification is carried out. Kawaji and Chen [84] also proposed a soft computing based approach for the identification of a nonlinear system. Wu et al. [85] proposed an approach of generating the fuzzy rules using a generalized dynamic fuzzy neural network which is built on the ellipsoidal basis function. Gao and Joo [86] proposed a robust adaptive fuzzy neural model for the identification of a certain class of multi input-multi output (MIMO) systems. This model has a fast online learning capability where the fuzzy rules are generated or deleted automatically. Panella and Galeo [87] proposed a clustering approach which is applied to a joint input-output space for the neuro-fuzzy modelling of nonlinear systems. Thangavel et al. [88] proposed the use of an intelligent controller for reactive power control. Abraham and Nath [89] used the neuro-fuzzy approach for modeling the electricity demand in Victoria. Kamia et al. [90] discussed the use of soft computing technique for the modelling of large scale plants.

2.4.1 Adaptive Network based Fuzzy Inference System

A neuro-fuzzy technique called Adaptive network based fuzzy inference system (ANFIS) [68, 74, 91] has been used as a prime tool in the present work. Adaptive network based fuzzy inference system (ANFIS) is a neuro fuzzy technique where the fusion is made between the neural network and the fuzzy inference system. In ANFIS the parameters can be estimated in such a way that both the Sugeno and Tsukamoto fuzzy models [92] are represented by the ANFIS architecture. Again with minor constraints the ANFIS model resembles the Radial basis function network (RBFN) functionally [93]. This ANFIS methodology comprises of a hybrid system of fuzzy logic and neural network technique. The fuzzy logic takes into account the imprecision and uncertainty of the system that is being modeled while the neural network gives it a sense of adaptability. Using this hybrid method, at first an initial fuzzy model along with its input variables are derived with the help of the rules extracted from the input output data of the system that is being modeled. Next the neural network is used to fine tune the rules of the initial fuzzy model to produce the final ANFIS model of the system. In this proposed work ANFIS is used as the backbone for the identification of real world systems. Detailed description of ANFIS is presented in Chapter 4.

TH-801_02610204

After building a model for the system based on ANFIS, the model can be used for forecasting future values using a suitable forecasting technique. A brief description of various forecasting methodologies available is presented below.

2.5 Forecasting Methodologies

Some of the data based time series forecasting methodologies that were carried out in the recent past are described in this section. Hard computing based methods like linear and nonlinear auto-regression have been traditionally used for system forecasting. But due to the highly nonlinear relationship between the determining factors and the demand forecast, it made the understanding of the system difficult. This made the auto regression approach extremely tedious and doubts also rose on the accuracy of the results.

The self adaptive data driven neural networks [94] scored a major point in the sense that it can approximate any arbitrary continuous function to any given accuracy [95] even with very little knowledge on the structural relationship between the demand forecast and the determining factors.

Fuzzy logic can be used to approximate any continuous function to a required degree of accuracy [96]. Both the neural network and the fuzzy systems have been found to have outperformed the auto-regression approach with higher forecasting accuracy [96]. It is because of this fact that these approaches have been well accepted in many practical utilities, like load forecasting [97, 98].

Studies also show that attempts were made to combine multiple neural networks with recursive least squares (RLS) algorithm to utilize the advantages of both the methods [97, 99]. In some utilities the different combination of fuzzy logic, neural networks and hard computing based techniques like statistical methods, fourier transform [100], bayesian curve fitting [101], Box-Jenkins Autoregressive integrated moving average with exogenous input(ARIMAX) method [102], extrapolation technique [103], wavelet transform [104], B-spline [105], state space [106] approach were also tried to checked if any implicit information that may be embedded in the available data can be extracted for use in the forecasting model. In certain cases it has been found that these approaches make the forecasting accuracy improve substantially. Applications were also found where the past knowledge of experienced forecasters were utilized to improve the forecasting accuracy [107].

In many cases it has been found that the presence of spurious data has a considerable effect on the forecasting accuracy [108]. Significant research work continued in the direction of identification, detection and estimation of bad data [109–111]. In order to overcome the drawbacks encountered by

TH-801_02610204

conventional learning methods like statistics based ones, the use of the evolutionary methods were also explored [105].

Traditionally, hard computing based methods like linear and nonlinear auto-regression methods have been applied to build forecasting models [112–114]. However, as the relationships between these demand forecasts and the determining factors are highly nonlinear, they are not properly understood. This made the auto regression approach extremely tedious and doubts also surfaced on the accuracy of the results.

On the other hand, even with a very little knowledge on the structural relationship between the demand forecast and the determining factors, the self adaptive data driven neural networks can approximate any arbitrary continuous function to any given accuracy [94]. Zhang et al. [115] studied the application of neural networks for forecasting in the electricity load consumption study.

Similarly the fuzzy logic can also be used to approximate any continuous function to a required degree of accuracy [95]. Performancewise both the neural network and the fuzzy systems can be put at the same level [96, 116, 117]. Both these methods have been found to have outperformed the auto-regression approach with higher forecasting accuracy [96] and hence have been well accepted in many practices by utilities for load forecasting [97, 98].

In addition to the soft computing methods, different hard computing methods were also found to have been applied for generating the different time series forecasting models.

Verleysen et al. [118] proposed the method of fractal projection forecasting for engineering applications. Amjady [107] used an approach in which he utilized the past knowledge of experienced forecasters to improve the forecasting accuracy, where the hard computing based Box-Jenkins Autoregressive integrated moving average with exogenous input(ARIMAX) method outperformed the stand alone neural network approach. Adya et al. [103] proposed the rule-based forecasting approach which is an expert system that uses features of time series alongwith weight extrapolation techniques. Gao and Tsoukalas [104] proposed the wavelet based forecasting method and used it for load identification and forecasting. Wang et al. [119] proposed a nonparametric smoothing technique to build a kernel projection forecasting model for a given small seasonal time series data. Meade [101] used the logistic Bayesian curve fitting model to study and investigate the effect of the assumed error structure on the forecasting accuracy. Saito and Abe [106] used the Kalman filter to derive a state space model for traffic forecasting. Wang et al. [119] proposed a nonparametric smoothing technique to build a kernel projection forecasting model for a given small seasonal time series data. Meade [101] used the logistic Bayesian curve fitting model to study and investigate the effect of the assumed error structure on the

TH-801_02610204

forecasting accuracy.

Work was also carried out combining different existing methodologies for generating the forecasting models. In many utilities the forecasting approaches are based on multiple neural networks being combined with Recursive least squares (RLS) algorithm [97, 99]. This approach helps in extracting the various load patterns which are implicitly embedded in the training data. Khotanzad et al. [97] explored such an approach. This multiple forecaster approach is also supported by many hard computing based forecasting applications and theories. This is because this approach makes the forecasting accuracy improve substantially [120]. Kim et al. [121] proposed the use of linear combination of a fuzzy system with a neural network for improving the forecasting accuracy. Abraham and Nath [122] discussed the use of Evolving fuzzy neural network (EFuNN) and the feed forward Artificial neural network (ANN) as well as a conventional statistical approach for modeling electricity demand. Morabito and Versaci [123] proposed the use of fuzzy neural networks for identification and forecasting the experimental urban air pollution data. Makiko and Yoshitsugu [100] proposed a new demand forecasting method using the neural network and Fourier transform.

In time series forecasting, spurious or Bad data (BD) affect the forecasting performance drastically. So, identification of bad data is a challenging task here. Milli et al. [108] gave a comparative assessment of the evaluating techniques for identifying Bad data (BD), i.e. data gathered from grossly erroneous measurements. These techniques are first classified, then explored and compared. Abur [110] proposed another algorithm for detecting the bad data using the measurement residuals of the measurements rejected by the Linear programming (LP) estimator. Then the bad measurements are identified and eliminated by estimating the measurement errors of the zero residual measurements. The residuals obtained from this second estimation step are made use of for this purpose. Long and Ling [111] proposed a new method of estimating the impulse response of a complex system based on its complex input and only the real part of its output data values. Zhang and Lo [109] proposed a recursive measurement error estimation and identification algorithm for identifying multiple interacting bad data in the power set in power system static state estimation. As an extension for further analysis of bad data, Zhang et al. [124] proposed an efficient bad data identification method for a state estimator which was implemented in real-time for a power system control centre in Northeast China. The proposed algorithm, called the Recursive measurement error estimation identification (RMEEI) is powerful and efficient having a high computational speed. Kandemir and Ramanujam [125] presented an abstraction, called data relation vectors, to improve the data access characteristics and memory layouts in regular computations, by defining a relation between the data elements accessed by close-by iterations and using this relation to

TH-801_02610204

guide a number of optimizations for array-based computations. Ahmed and Cruz [126] proposed the modified version of this algorithm and proved that it yields a least squares solution, which is comparable to the Recursive least squares (RLS) method and initialized it for a Data-driven echo canceller (DDEC). Erdogmus and Principe [127] investigated error-entropy-minimization in adaptive systems training, by minimization of a Csiszar distance measure between the densities of desired and system outputs. Didenko and Movchan [128] investigated the evaluation of the uncertainty of A/D conversion with reasonable accuracy by considering the minimum number of parameters of the metrological model of the Data acquisition system (DAS). Joachims [129] presented an approach to automatically optimize the retrieval quality of search engines using clickthrough data. Babcock et al. [130] discussed one adaptive strategy for processing bursty streams of data which fluctuate over time without sacrificing the system performance as in many applications, systems are required to produce rapid or realtime query responses. Weekes and Forgel [131] proposed using of evolutionary training of Artificial neural networks (ANNs) to generate predictive models of quantitative structure, based on the data of activity relationships between a set of molecular descriptors and activity. In order to overcome the drawbacks encountered by conventional learning methods, Coelho and Krohling [105] proposed the use of an intelligence methodology called swarm optimization methodology to provide a stochastic global search of B-spline networks for nonlinear system identification. Meekhof and Heckendorn [132] explored the possibility of using string space transformations to reduce the perplexity of the modeling problem and thereby improve model performance of a Markov-based classifier on the problem of classifying English and Spanish character strings, where training set size is arbitrarily limited. Pekar and Stecha [133] proposed a real time system parameter estimation from the set of input-output data by minimization of quadratic norm errors of system equations. Lii et al. [134] proposed translation model training in Statistical machine translation (SMT) using off line data optimization and online model optimization.

2.6 Research gap

In all the data based time series forecasting methods that have been discussed in the previous section, the availability of a sufficient number of data is a must for achieving a good forecasting model. However, in many cases it has been observed that the number of available data is not sufficient for modeling purpose. In such a scenario one alternative solution lies in the generation of more data so that modeling of the system under consideration can be carried out with good accuracy. But generation of more data is not always possible as it may be a very costly affair, particularly in a production environment, like

TH-801_02610204

a thermal power plant. This is because in order to get the training data, the power plant has to be kept running for long period of time. In addition, the thermal power plants have the inherent problem of taking a long time to start before it can be fully loaded to get the requisite data. So the training of the ANFIS has to be carried out with the available limited number of input output data with good modeling accuracy. This necessitates the development of a proper procedure to choose the data critically and optimally for training the neural network. With these objectives in mind it has been proposed to utilize the salient feature of self-adaptability in approximating an arbitrary function with a good accuracy of the neural network and the inherent property of the fuzzy logic to take into account the imprecision and uncertainty of a system which is considered for modeling, with the help of an ANFIS model. A novel method for optimal choice of the available data for training the ANFIS network is proposed in the present work. The major objective of the present work is to formulate an efficient method to choose the available data optimally from the existing data set to build an ANFIS model for the system under consideration and use the model for faithful prediction of future values. The research focuses on modeling of real world systems where input output data required for modeling are either scanty or difficult to generate. The research work then aims to design a controller based on ANFIS so that the controller performs satisfactorily even with plant parameter variations and disturbance.

The methodology followed for achieving the above goal is described in the next chapter.

TH-801_02610204

Chapter 3

Research Objectives and Methodology

In the quest for developing a model for a system based on its available input output data, it has been observed that in the conventional modeling approach the results depend on the mathematical model of the system and its accuracy. In cases where the mathematical model is not available the system analysis becomes very difficult. It is in this context that the soft computing approach can provide a viable alternative for system modeling in cases where the conventional hard computing techniques cannot be used. In the present work it is proposed to use these soft computing techniques for time series modeling of a real world system with the help of its input-output data.

3.1 Objectives of the present work

In data driven modeling techniques like ANFIS, the number of time series data that are available for system modeling has to be in abundance, as this number has a direct bearing on the modeling result. It has been found that more is the number of available input-output data the better will be the modeling result. This is because with more number of training data the neural network of the hybrid ANFIS system will have a better training session which will result in a good modeling validation and prediction. But in most of the real time systems the available data is generally less, so modeling has to be carried out with the available small pool of data only. The primary concern of the thesis is for this type of situations where sufficiently large quantity of input-output data is not available or difficult to generate and the model of the system has to be built for practical applications like forecasting. This research focuses on these real life systems having scanty input-output data and attempts to identify the model using the data selected critically such that these optimally selected data can be used to build the system model faithfully. Hence optimal selection of the available minimal data set for training the

TH-801_02610204

ANFIS model is the key objective of this research. The key objectives of the present research can be highlighted as follows:

- ANFIS based modeling of real world systems
- Building of ANFIS model for real world systems where generation of training data is difficult and time consuming
- Building of ANFIS model for real world system where generation of data is expensive
- Building of ANFIS model for real world system where available input output data are scanty
- Designing of an ANFIS based controller which performs satisfactorily even in the presence of plant parameter variations and disturbances

The building of an ANFIS model faithfully with available scarce input output data is studied in this research work and a novel solution is offered so that this research gap can be bridged.

3.2 Methodology of the present work

ANFIS is the backbone for the present work where a small set of time series data critically selected from a real time system are used for the modeling of the system. The modeling methodology used in the present work is briefly mentioned in the following subsections.

3.2.1 ANFIS based methodology for modeling

The Adaptive network based fuzzy inference system (ANFIS) [68,74,91] is a hybrid system comprising of the neural network and the fuzzy logic. It is a data driven procedure which can be used to provide the solution of function approximation problems in a neural network platform. Here at first a fuzzy inference system comprising of an initial fuzzy model is formed, based on the fuzzy rules extracted from the input output data set. In the next step the neural network is used to fine tune the rules of the initial fuzzy model that was built. Using ANFIS methodology the network is trained. The number of training data used in the ANFIS is drastically reduced by applying an optimal data selection criterion.
Two novel techniques are attempted for selecting the optimal input-output data pairs. One of these methods is known as the Full factorial design (FFD) [135, 136] and is based on statistical design of experiments. The other technique is a cross validation technique known as the V-fold technique

TH-801_02610204

[137–139]. These two techniques are described briefly in the next three subsections. This work is an endeavour to augment the ANFIS as a modeling technique by incorporating the above two novel techniques for accomplishing the training of the network faithfully when the available data is scarce.

3.2.2 Full factorial design based methodology for optimal data selection

From the statistical point of view, in experiments where a large number of tests is involved, the order of selection of the test specimens has to be randomized, so that each specimen has an equal chance of being selected for the test. The training of the neural network can also be replicated to a statistical experiment involving a large number of tests for optimization. In such an experimental setup the identification of the important variables that affects the experimental results forms an important aspect. The experimental variables controlled by the operators on which the system's response depends are called the factors. The effect of the response of one factor may or may not depend on the levels of the other factors. The number of factors and their levels required can be fixed depending upon the complexity of the experiment. The statistical technique based Full factorial design (FFD) [135, 136] methodology identifies the important factors and levels of the experiments conducted to model a real time system. This FFD methodology is used for selecting the critical data set for training the ANFIS model of the real system.

3.2.3 V-fold based methodology for optimal data selection

If a neural network is built using a specific learning data set, it has to be tested with a data set which is independent of the data set that was used to train the network. It has often been found that obtaining an independent test data set is very difficult. The neural network can also be trained by another accurate method called the V-fold technique [137–139]. This method possesses the distinct advantage of not requiring a separate and independent data set for testing the accuracy of the network. In this methodology the available data set is subdivided into some subgroups. All but one of the subgroups is used for training the neural network. Next the remaining subgroup is used for testing. This procedure is continued by testing the network with a new subgroup every time. This approach of the V-fold cross validation technique is used in our research work to select the optimal data set for training the ANFIS network in the model of the real time system considered in our study.

TH-801_02610204

3.2.4 Full factorial design combined with V-fold based methodology for optimal data selection

Both the Full factorial design (FFD) and the V-fold technique are combined to select the critical data set for training the ANFIS model.

3.2.5 Model building with statistical approaches for performance study

The conventional statistical methods of modeling are also studied in the present research and applied to build the system models to study their performance against the ANFIS model. The three statistical models namely Autoregressive with exogenous input (ARX) [140], Autoregressive moving average with exogenous input (ARMAX) [140] and Autoregressive integrated moving average with exogenous input (ARIMAX) [140] are used for modeling the real world system and the results obtained are compared with those obtained by using the proposed models.

3.2.6 Genetic algorithm based fuzzy methodology

A Genetic algorithm(GA) [67,141,142] based fuzzy model is build for studying its performance against the ANFIS model. In this methodology the genetic algorithm is used to update the consequent parameters of the fuzzy model of the system under consideration.

3.2.7 Controller with the FFD-V-fold based methodology

This research also studies the utility of the proposed ANFIS model for designing a controller. A controller based on the proposed FFD-V-fold based ANFIS method is built and its performance is tested against disturbance and plant parameter variation. Out of the different ANFIS based models which are proposed, the model showing the best result is used for the design of a controller.
In the next chapter, the ANFIS based methodology for modeling is discussed in details.

TH-801_02610204

Chapter 4

Adaptive Network based Fuzzy Inference System

The adaptive network based fuzzy inference system (ANFIS) [68, 74, 91] is a data driven procedure representing a neural network approach for the solution of function approximation problems. Data driven procedures for the synthesis of ANFIS networks are typically based on clustering a training set of numerical samples of the unknown function to be approximated. Since introduction, ANFIS networks have been successfully applied to classification tasks, rule-based process control, pattern recognition and similar problems. Here a fuzzy inference system comprises of the fuzzy model [34, 35] proposed by Takagi, Sugeno and Kang to formalize a systematic approach to generate fuzzy rules from an input output data set.

4.1 ANFIS structure

For simplicity, it is assumed that the fuzzy inference system under consideration has two inputs and one output. The rule base contains the fuzzy if-then rules of Takagi and Sugeno's type [143] as follows:

If x is A and y is B then z is f(x,y)

where A and B are the fuzzy sets in the antecedents and $z = f(x, y)$ is a crisp function in the consequent. Usually $f(x, y)$ is a polynomial for the input variables x and y. But it can also be any other function that can approximately describe the output of the system within the fuzzy region as specified by the antecedent. When f(x,y) is a constant, a zero order Sugeno fuzzy model is formed which may be considered to be a special case of Mamdani fuzzy inference system [144] where each rule consequent is specified by a fuzzy singleton. If f(x,y) is taken to be a first order polynomial a first order Sugeno

TH-801_02610204

fuzzy model is formed. For a first order two rule Sugeno fuzzy inference system, the two rules may be stated as:

Rule 1: If x is A_1 and y is B_1 then $f_1 = p_1x + q_1y + r_1$

Rule 2: If x is A_2 and y is B_2 then $f_2 = p_2x + q_2y + r_2$

Here type-3 fuzzy inference system proposed by Takagi and Sugeno [143] is used. In this inference system the output of each rule is a linear combination of the input variables added by a constant term. The final output is the weighted average of each rule's output. The corresponding equivalent ANFIS structure is shown in Fig. 4.1.

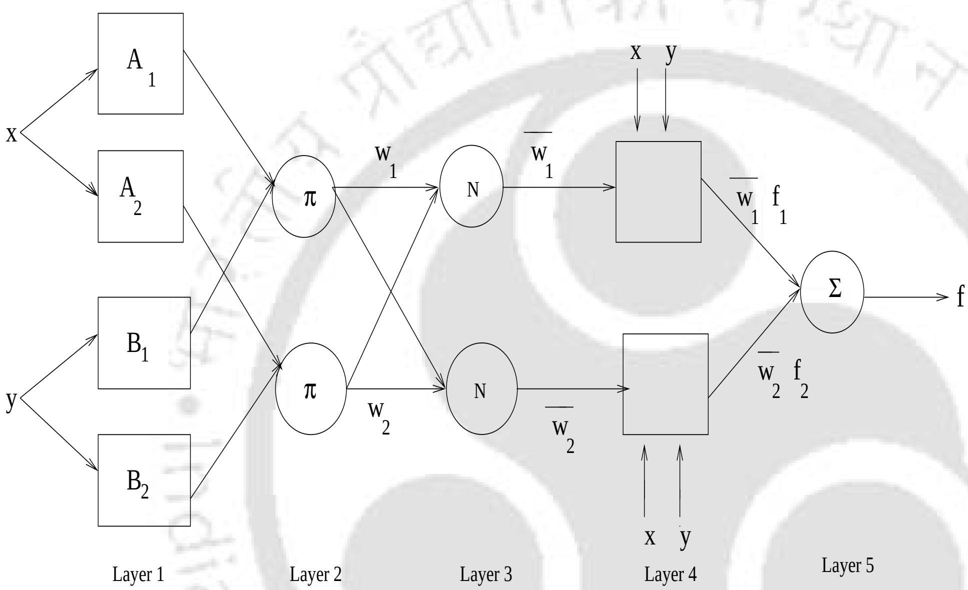

Figure 4.1: Type-3 ANFIS Structure

The individual layers of this ANFIS structure are described below :

Layer 1: Every node i in this layer is adaptive with a node function

$$O_i^1 = \mu_{A_i}(x) \tag{4.1.1}$$

where, x is the input to node i, A_i is the linguistic variable associated with this node function and μ_{A_i} is the membership function of A_i. Usually $\mu_{A_i}(x)$ is chosen as

$$\mu_{A_i}(x) = \frac{1}{1 + [(\frac{x-c_i}{a_i})^2]^{b_i}} \tag{4.1.2}$$

Or

$$\mu_{A_i}(x) = exp\,\{-(\frac{x - c_i}{a_i})^2\} \tag{4.1.3}$$

where x is the input and $\{a_i, b_i, c_i\}$ is the premise parameter set.

Layer 2: Each node in this layer is a fixed node which calculates the firing strength w_i of a rule. The output of each node is the product of all the incoming signals to it and is given by,

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2 \tag{4.1.4}$$

Layer 3: Every node in this layer is a fixed node. Each i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of firing strengths of all the rules. The output from the i^{th} node is the normalized firing strength given by,

$$O_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \tag{4.1.5}$$

Layer 4: Every node in this layer is an adaptive node with a node function given by

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}\,(p_i x + q_i y + r_i), \quad i = 1, 2 \tag{4.1.6}$$

where $\overline{w_i}$ is the output of Layer 3 and $\{p_i, q_i, r_i\}$ is the consequent parameter set.

Layer 5: This layer comprises of only one fixed node that calculates the overall output as the summation of all incoming signals, i.e.

$$O_i^5 = overall\ output = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{4.1.7}$$

4.2 Learning Algorithm

In the ANFIS structure, it is observed that given the values of premise parameters, the final output can be expressed as a linear combination of the consequent parameters. The output f in Fig. 4.1 can be written as

$$\begin{aligned} f &= \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \\ &= \overline{w_1}\, f_1 + \overline{w_2}\, f_2 \\ &= (\overline{w_1}\, x)p_1 + (\overline{w_1}\, y)q_1 + (\overline{w_1})r_1 + (\overline{w_2}\, x)p_2 + (\overline{w_2}\, y)q_2 + (\overline{w_2})r_2. \end{aligned} \tag{4.2.1}$$

where f is linear in the consequent parameters $(p_1, q_1, r_1, p_2, q_2, r_2)$.

In the forward pass of the learning algorithm, consequent parameters are identified by the least squares estimate. In the backward pass, the error signals, which are the derivatives of the squared error with respect to each node output, propagate backward from the output layer to the input layer. In this backward pass, the premise parameters are updated by the gradient descent algorithm [27–29].

4.3 Derivation of the Initial Fuzzy Model

As described earlier, in ANFIS based system modeling for a set of rules with fixed premise parameters, identification of an optimal fuzzy model with respect to the training data reduces to a linear least-squares estimation problem. A fast and robust method for identification of fuzzy models from input-output data was proposed by S.L.Chiu [77]. This method selects the important input variables when building a fuzzy model from data by combining cluster estimation method with a least squares estimation algorithm. The method follows in two steps : i) First step involves extraction of an initial fuzzy model from input output data by using a cluster estimation method incorporating all possible input variables. ii) In the next step the important input variables are identified by testing the significance of each variable in the initial fuzzy model.

4.3.1 Extracting the initial fuzzy model

In order to start the modeling process, an initial fuzzy model has to be derived. This model is required to find the number of inputs, number of linguistic variables and hence the number of rules in the final fuzzy model. The initial model is also required to select the input variables for the final model and also the model selection criteria, before the final optimal model can be derived. This initial fuzzy model can be selected based on the fuzzy rules framed by either using the subtractive clustering technique [145] or the grid partitioning method [74] [68, 91].

TH-801_02610204

Subtractive Clustering Technique

As a first step towards extracting the initial fuzzy model by subtractive clustering, this technique is applied to the input output data pairs, which are obtained from the system which is to be modeled. The cluster estimation technique helps in locating the cluster centers of the input output data pairs. This in turn helps in the determination of the rules which are scattered in input output space, as each cluster center is an indication of the presence of a rule. In addition to this it also helps to determine the values of the premise parameters. This is important because an initial value, which is very close to the final value, will eventually result in the quick convergence of the model towards its final value during the training session with neural network. In this clustering technique the potentials of all the input output data points are calculated as functions of their Euclidian distances from all the other data points. The points having a potential above a certain preset value are considered as cluster centers. After the cluster centers are ascertained the initial fuzzy model can be subsequently extracted as the centers will also give an indication of the number of linguistic variables. The cluster estimation method for determining the number of rules and initial rule parameters [77] is briefly described below.

Let us consider a collection of n data points $\{x_1, x_2, ..., x_n\}$ in an M dimensional space. The data points are assumed to be normalized in each dimension so that they are bounded by a unit hypercube. Each data point is considered to be a potential cluster center. P_i is a measure of the potential of data point x_i to serve as a cluster center and is defined as

$$P_i = \sum_{j=1}^{n} e^{-\alpha \|x_i - x_j\|^2} \tag{4.3.1}$$

where

$$\alpha = \frac{4}{r_a^2}. \tag{4.3.2}$$

$\|.\|$ denotes the Euclidean distance and r_a is a positive constant. Thus measure of the potential for a data point is a function of its distances to all other data points. Here r_a is the radius effectively defining a neighborhood. Data points outside this radius have little influence on the potential. After the potential of every data point has been computed, the data point with the highest potential is selected as the first cluster center. Let x_1^* be the location of the first cluster center and P_1^* be its potential value. Then the potential of each data point x_i is revised by the formula

TH-801_02610204

$$P_i = P_i - P_1^* e^{-\beta \|x_i - x_1^*\|^2} \tag{4.3.3}$$

where

$$\beta = \frac{4}{r_b^2} \tag{4.3.4}$$

and r_b is a positive constant. Thus an amount of potential is subtracted from each data point as a function of its distance from the cluster center. The constant r_b is effectively the radius defining the neighborhood which will have measurable reduction in potential. Typically chosen value of $r_b = 1.25 r_a$ [77].

When the potentials of all data points have been revised according to Eq. 4.3.3, the data point with the highest remaining potential is selected as the second cluster center. The potential of each data point is further reduced according to their distance to the second cluster center. In general, after the k^{th} cluster center has been obtained, the potential of each data point is revised by the formula

$$P_i = P_i - P_k^* e^{-\beta \|x_i - x_k^*\|^2} \tag{4.3.5}$$

where x_k^* is the location of the k^{th} cluster center and P_k^* is the potential value.

The process of acquiring new cluster center and revising potentials repeats until the stopping criterion $P_k^* < 0.15 P_1^*$ [145] is satisfied. Each cluster center as derived above is in essence a data point that describes a characteristic input-output behaviour of the system we wish to model. Hence each cluster center can be used as the basis of a rule that describes the system behaviour.

It is considered that $\{x_1^*, x_2^*, ..., x_c^*\}$ is a set of c cluster centers in an M dimensional space. It is again considered that the first N dimensions correspond to input variables and the last $M - N$ dimensions correspond to output variables. Each vector x_i^* is decomposed into two component vectors y_i^* and z_i^* where y_i^* is the location of the cluster center in input space and z_i^* is the location of the cluster center in output space. Therefore x_i^* may be represented as

$$x_i^* = [y_i^*; z_i^*]$$

Each cluster center x_i^* is considered as a fuzzy rule, "if input is near y_i^* then output is near z_i^*", to describe the system behaviour. Given an input vector y, the degree to which rule i is fulfilled is defined as

TH-801_02610204

$$\mu_i = e^{-\alpha \|y - y_i^*\|^2} \tag{4.3.6}$$

where α is a constant defined by Eq. 4.3.2. Output vector z is computed as

$$z = \frac{\sum_{i=1}^{c} \mu_i z_i^*}{\sum_{i=1}^{c} \mu_i} \tag{4.3.7}$$

This computational scheme can be viewed in terms of an inference system employing fuzzy if-then rules. Each rule has the following form :

IF Y_1 is A_{i1} and Y_2 is A_{i2} and ... THEN Z_1 is B_{i1} and Z_2 is B_{i2}...

where Y_j is the jth input variable and Z_j is the jth output variable; A_{ij} is an exponential membership function in the ith rule with the jth input and B_{ij} is a singleton in the ith rule associated with the jth output. For the ith rule that is represented by cluster center x_i^*, A_{ij} and B_{ij} are given by :

$$A_{ij}(Y_j) = e^{-0.5\left(\frac{Y_j - y_{ij}^*}{\sigma_{ij}}\right)^2} \tag{4.3.8}$$

$$B_{ij} = z_{ij}^* \tag{4.3.9}$$

where y_{ij}^* is the jth element of y_i^* and z_{ij}^* is the jth element of z_i^* and $\sigma_{ij}^2 = \frac{1}{2\alpha}$ [77].

Grid Partitioning Technique

The second method which can be used for framing the rules of the initial fuzzy model is by grid partitioning [74] [68, 91]. This method is used when the number of inputs and their membership functions are less. Here the input space are partitioned into a number of fuzzy regions to form the antecedents of the fuzzy rules. The Grid partitioned fuzzy space for a two input model, with each input having three membership functions each is shown in Fig. 4.2. The two dimensions represent the abscissa and the ordinate of the input space. The rules obtained from either of the two methods are then optimized by using ANFIS methodology developed by Jang [91]. This method involves optimization of the premise membership functions by gradient descent algorithm combined with optimization of the consequent equations by linear least squares estimation.

TH-801_02610204

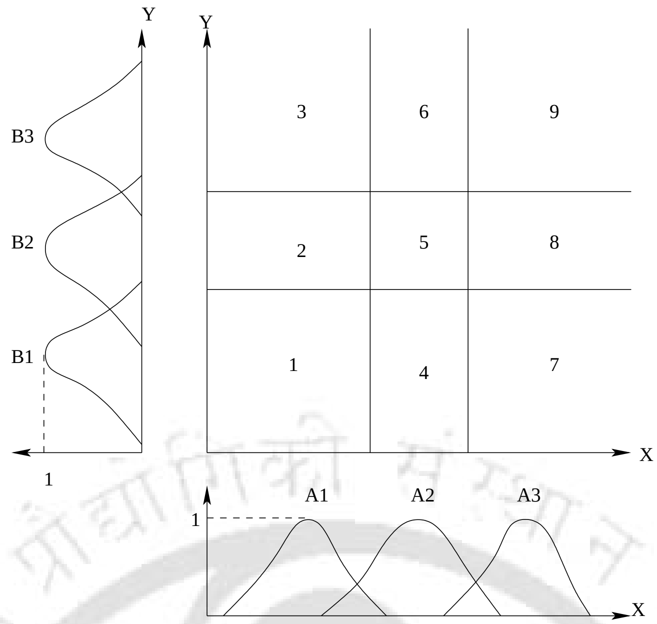

Figure 4.2: Grid partitioned fuzzy subspaces for a type-3, 2-input ANFIS

4.3.2 Selection of input variables and final fuzzy model

Any good criteria which is used for model selection based on some inputs cannot guarantee that the model will be producing the optimal result unless the results from all possible combinations of the input variables in the model are compared. But this is a tedious task as even for a moderate number of input variables N, 2^N possible number of combinations of the variables arises. From the modeling viewpoint, incorporation of only the important variables results in a practical model which is simpler, more reliable and useful for application. This is because now only a fewer variables will have to be measured. For achieving this, the importance of each input variable can be ascertained from the initial fuzzy model. In the proposed ANFIS model this has been accomplished by choosing the model showing the least modeling error from the models obtained using the following two methods:

- First method:-In this method the fuzzy rule framework provides an easy mechanism to test the importance of each input variable without having to generate new models. The basic idea is to remove all antecedent clauses associated with a particular input variable from the rules and then evaluate the performance of the model by applying the checking error criterion [77]. If this decreases the modeling error the process is repeated by eliminating another input variable. If the modeling error increases, the eliminated variable is retained and another variable is eliminated in its place. This process is continued until the modeling error can be decreased no further by eliminating any more extra input variable. The criterion that is used for selection of the final fuzzy

TH-801_02610204

model is the Root mean square error (RMSE). The ANFIS structure of the system which is being modeled is considered as the final model for which the RMSE is the minimum. For example, suppose the initial model has four inputs, with rules of the form :

IF Y_1 is A_{i1} and Y_2 is A_{i2} and Y_3 is A_{i3} and Y_4 is A_{i4} THEN Z_1 is B_{i1}

The importance of the Y_3 variable in the model can be tested by temporarily removing the antecedent clauses that involve Y_3, thus truncating the rules to the form :

IF Y_1 is A_{i1} and Y_2 is A_{i2} and Y_4 is A_{i4} THEN Z_1 is B_{i1}

If the resultant model performance does not degrade with respect to the performance measure which is the RMSE of the output corresponding to an independent set of checking data, then Y_3 can be eliminated from the possible important variables. The variable selection process for a four input initial model is shown in Fig. 4.3.

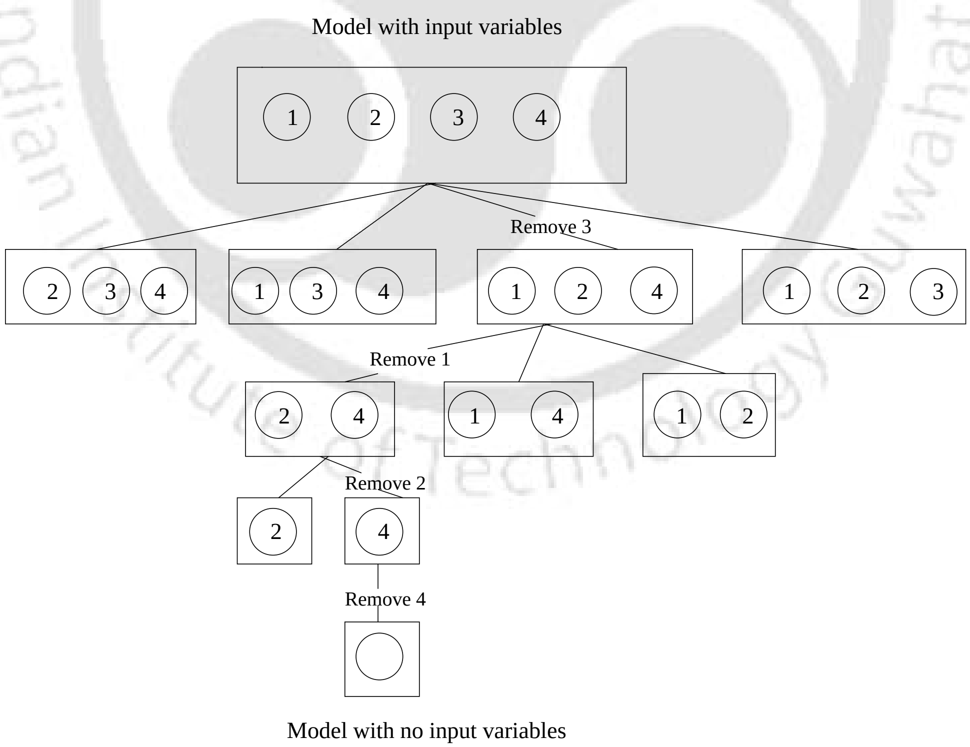

Figure 4.3: Variable Selection Process for a four input initial model

TH-801_02610204

- Second method:- For simplicity, the second model assumes that ANFIS consists of only two inputs. For choosing these inputs a set of 10 dynamic modeling inputs [77] is selected which comprises of 6 from the historical inputs of the system and 4 from the historical outputs of the system to be modeled. Out of these 10 dynamic modeling candidates, for each of the 6 inputs there can be 4 different combinations with the 4 different outputs. So there will be a total of $6 \times 4 = 24$ different combinations of input candidate pairs. Each of these different combinations will lead to the generation of 24 different initial fuzzy models as follows:

$$\{x(t-1), y(t-1)\}, \{x(t-1), y(t-2)\}, \{x(t-1), y(t-3)\}, \{x(t-1), y(t-4)\},$$
$$\{x(t-2), y(t-1)\}, \{x(t-2), y(t-2)\}, \{x(t-2), y(t-3)\}, \{x(t-2), y(t-4)\},$$
$$\{x(t-3), y(t-1)\}, \{x(t-3), y(t-2)\}, \{x(t-3), y(t-3)\}, \{x(t-3), y(t-4)\},$$
$$\{x(t-4), y(t-1)\}, \{x(t-4), y(t-2)\}, \{x(t-4), y(t-3)\}, \{x(t-4), y(t-4)\},$$
$$\{x(t-5), y(t-1)\}, \{x(t-5), y(t-2)\}, \{x(t-5), y(t-3)\}, \{x(t-5), y(t-4)\},$$
$$\{x(t-6), y(t-1)\}, \{x(t-6), y(t-2)\}, \{x(t-6), y(t-3)\}, \{x(t-6), y(t-4)\};$$

  From the above 24 models, the model with the least RMSE is chosen as the initial fuzzy model. In this method the number of membership functions per input can be determined by applying either the subtractive clustering or the grid partition based technique.

4.3.3 Optimization of the initial fuzzy model

The ANFIS structure of the system which is being modeled is considered as the final model for which the RMSE is the minimum. The consequent parameters of the initial fuzzy model are updated by using the Least squares estimation (LSE) algorithm. Similarly, the rules which are obtained from the clustering or the grid partition based method are updated by neural network which uses back propagation learning method with gradient descent algorithm. This updation leads to the optimization of the premise parameters of the fuzzy membership functions to give the final fuzzy model.

4.4 Experimental Data

The data that has been used for the experimental purpose are taken from two different sets.

In the first set the data are taken from the bench mark problem of Box and Jenkins [102]. It comprises of the input-output data that are collected from a gas furnace. In the dataset the input is the gas feed rate to the furnace in cubic feet per minute (ft^3/min)and the output is in the form of percentage of carbon-dioxide (CO_2) concentration in the outlet gas.

The second data set comprises of five different data sets that are obtained by physically visiting a thermal power plant under North East Electric Power Corporation (NEEPCO) Limited situated at Kathalguri, Assam, India. This real time data have been collected from the past records maintained by the power plant authorities in hard form. In the first four data-sets the input is the gas flow rate in cubic kilometer per hour (km^3/hr) and the output is in the form of generated power in Gigawatt (GW). These are the numerical records of the daily data that the power plant authority records after every half hour. These were subdivided into four different sets with the half hourly data covering a few months at different periods of the year.

The power plant authority also keep a record of data on a monthly basis by aggregating the half hourly data over the whole month. The fifth subset of the thermal power plant data has been collected from the records of this monthly data. In this subset the input is the gas feed to the plant in Million metric standard cubic metre (Mmscum)($10^6 m^3$) and the output is in the form of generated energy in Million unit (MU), where 1 unit = 1 kWh (kilo-watt-hour).

These numerical data in the same form as collected from the plant's record are analyzed for determining the different levels and folds for the Full factorial design (FFD) and the V-fold based experiments respectively. The different levels of the Full factorial design experiments are determined by noting the numerical values around the highest, lowest, mid-values and cluster centers of the collected data. Next, these data are used for training and validation of the proposed model. A sample data set that was collected from the thermal power plant is presented in the Appendix A.

4.5 Experimental Results and Discussion

The ANFIS based model identification method is applied to the well known example of Box and Jenkins gas furnace data [102]. Next the modeling of a thermal power plant based on real data is considered. This real time data set has been obtained from a thermal power plant under North East Electric Power Corporation (NEEPCO) Limited situated at Kathalguri, Assam, India. The modeling error is calculated as the difference between the model output and the real system output and either the Mean square error (MSE) or the Root mean square error (RMSE) has been used as the comparative index. The details of these experiments and the results obtained are discussed in the following subsections.

TH-801_02610204

4.5.1 ANFIS Model for Gas Furnace

In this experiment 296 input-output data pairs from Box and Jenkins [102] gas furnace data are considered for modeling with ANFIS. In the dataset the input $x(t)$ is the gas feed rate in cubic feet per minute (ft^3/min)and the output $y(t)$ is in the form of percentage of carbon-dioxide (CO_2) concentration in the outlet gas.

In order to extract the dynamic process model for the prediction of $y(t)$, 10 candidates are considered as input variables following standard method [77]. These 10 input variables are :

$$\{x(t-1), x(t-2), x(t-3), x(t-4), x(t-5), x(t-6), y(t-1), y(t-2), y(t-3), y(t-4)\}$$

After converting the data so that each training data considers

$$\{x(t-1), x(t-2), x(t-3), x(t-4), x(t-5), x(t-6), y(t), y(t-1), y(t-2), y(t-3), y(t-4)\},$$

the number of effective data points reduces to 290. Using a cluster radius of $r_a = 0.5$ [77], the modeling was started with these ten prospective input candidates in the initial fuzzy model. Consequently in the final Takagi-Sugeno type fuzzy model which is derived by using Chiu's [77] input variable selection method, the number of input variables reduced to 2 and the number of rules narrowed down respectively to 3 and 4 for the subtractive clustering and the grid based method. The optimally selected input variables are $x(t-3)$ and $y(t-1)$. The final optimized ANFIS model of the gas furnace process is obtained after the updation of the consequent parameters by LSE algorithm and updation of the premise parameters by the back propagation gradient descent algorithm. In order to provide the same basis for comparison with other published results, the entire dataset of 290 input output pairs is used for training. Fig. 4.4 and Fig. 4.5 show the rules and parameters of the grid partition based and subtractive clustering based ANFIS models for the gas furnace data. In Fig. 4.6 the actual output and the output predicted by the ANFIS model are plotted vs. sample number. In Fig. 4.7 the prediction error of the ANFIS model for the training data is plotted vs. sample number. Fig. 4.8 shows the RMSE of the ANFIS model during training vs epochs. The surface graph corresponding to the output is shown in Fig. 4.9.

TH-801_02610204

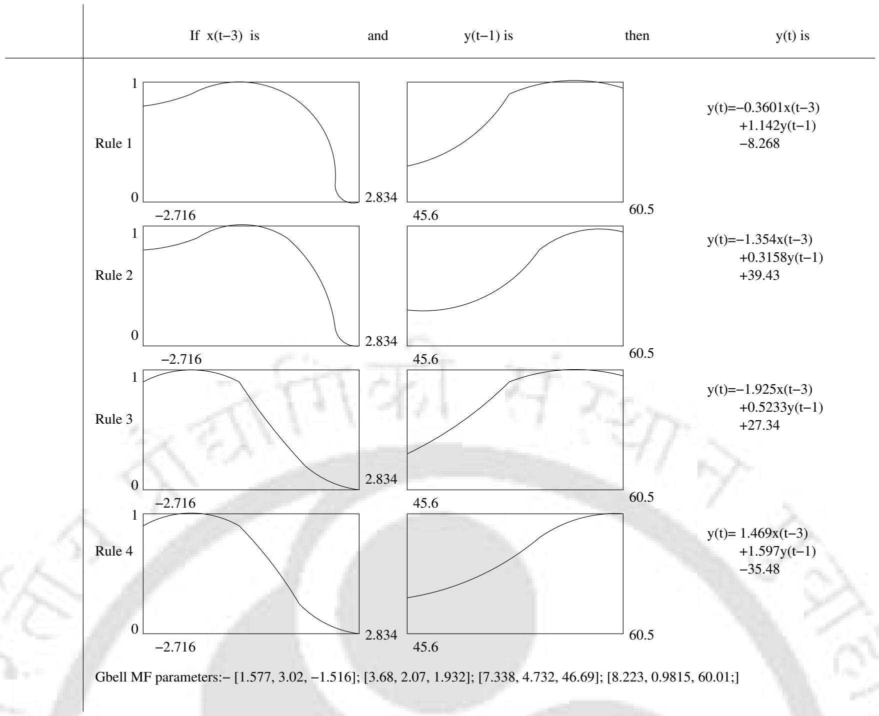

Figure 4.4: Rules for the grid partition based ANFIS model when trained with entire data set for the gas furnace example

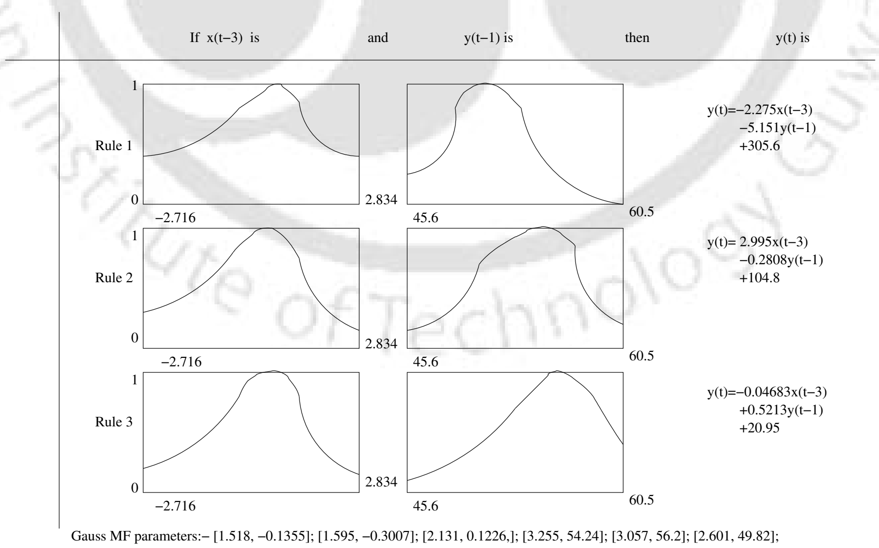

Figure 4.5: Rules for the subtractive clustering based ANFIS model when trained with entire data set for the gas furnace example

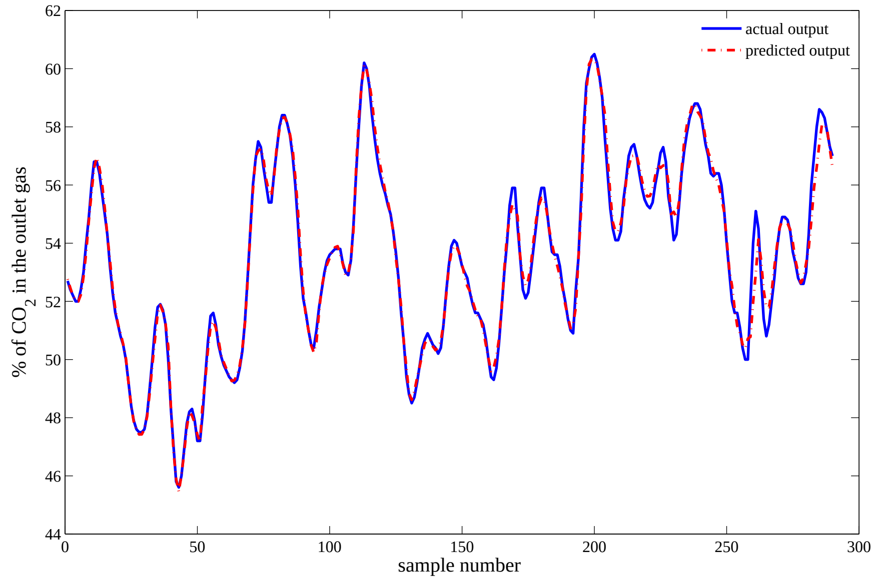

Figure 4.6: Actual and ANFIS model predicted output with Box and Jenkins gas furnace data

In Table 4.1 the 2-input based ANFIS model is compared with other models available in literature. The first 7 rows are excerpted from a table in [77]. The comparison shows that the modeling result of the gas furnace data has improved with the proposed two inputs subtractive clustering and grid based ANFIS models as compared with the other models both in terms of the number of rules required and the RMSE. The proposed grid based model is also showing better performance as compared to Chiu's two input model [77] but with the requirement of one more rule.

Another experiment was performed using the first half of the Box and Jenkins gas furnace data for training the neural network and the second half of the data set for validation of the model. Fig. 4.10 shows the rules and parameters of this model. The modeling result is shown in Fig. 4.11 where the actual output and the output predicted by ANFIS model are plotted vs. sample number.

4.5.2 ANFIS Model for Thermal Power Plant

In this experiment five different data sets, comprising of input-output data pairs collected from the thermal power plant are considered for modeling with ANFIS. In the first four data-sets the input $x(t)$ is the gas flow rate in cubic kilometer per hour $(\mathrm{km}^3/\mathrm{hr})$ and the output $y(t)$ is in the form of generated power in Gigawatt (GW). These data sets are based on the daily data collected from the thermal power plant at an interval of every half hour. These half hourly data have been classified into four different data sets depending on their collection time which is spread over different periods of the year. In the fifth data

Table 4.1: Comparison of various models derived for the Box and Jenkins gas furnace data. The first 7 rows are excerpted from a table in [76]

Sl. No.	Model	Input variables	Number of Rules	Model Error (mean square)
1	Tong's Model [1980]	$x(t-4), y(t-1)$	19	0.4690
2	Pedrycz's Model [1984]	$x(t-4), y(t-1)$	81	0.3200
3	Xu's model [Xu and Yong 1987]	$x(t-4), y(t-1)$	25	0.3280
4	Takagi-Sugeno model [Sugeno and Tanaka 1991]	$x(t-1), x(t-2), x(t-3), y(t-1), y(t-2), y(t-3)$	2	0.0680
5	Sugeno's position-gradient model [Sugeno and Yasukawa 1993]	$x(t-3), x(t-4)$	6	0.1900
6	Takagi-Sugeno model [3-input] [Chiu 1996]	$x(t-3)$ $y(t-1)$ $y(t-3)$	3	0.0720
7	Takagi-Sugeno model [2-input] [Chiu 1996]	$x(t-3), y(t-1)$	3	0.1460
8	Proposed model [2-input]	$x(t-3), y(t-1)$	3	0.1322 (subtractive clustering based)
			4	0.1277 (grid partition based)

TH-801_02610204

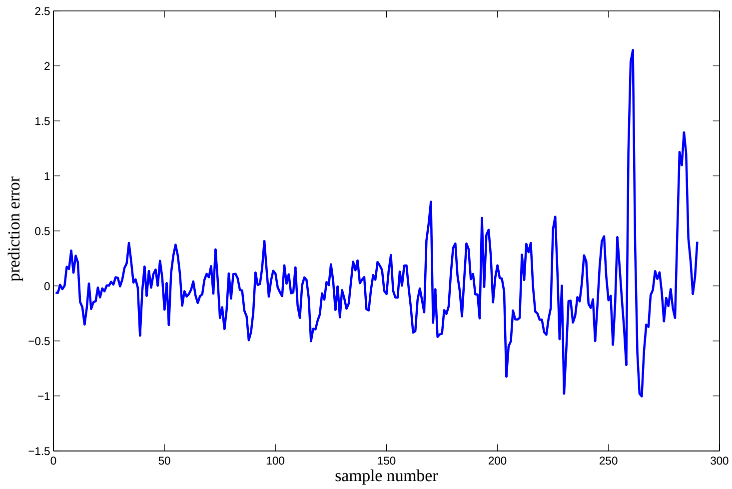

Figure 4.7: ANFIS model prediction error for Box and Jenkins gas furnace data

set the input $x(t)$ is the gas feed to the plant in Million metric standard cubic metre (Mmscum)($10^6 m^3$) and the output $y(t)$ is in the form of generated energy in Million unit (MU), where 1 unit = 1 kWh (kilo-watt-hour). This fifth data set has been collected from the thermal power plant on the basis of the monthly data for a certain period of time.

The dynamic process model for the above thermal power plant is extracted as in the previous case for the gas furnace data. The 10 candidates which are considered for the input variables are the following:

$$\{x(t-1), x(t-2), x(t-3), x(t-4), x(t-5), x(t-6), y(t-1), y(t-2), y(t-3), y(t-4)\}$$

In all the five data sets, modeling of the thermal power plant was considered with these 10 prospective input candidates in the initial fuzzy model. Both the grid partition method and the subtractive clustering method were applied with a cluster radius of $r_a = 0.5$. The number of input variables finally reduced to 2 and the number of rules narrowed down to 4. After the updation of the consequent parameters by LSE algorithm and updation of the premise parameters by the back propagation gradient descent algorithm, the final optimized ANFIS model of the thermal power plant is obtained. The rules of the different models of ANFIS and their experimental results are shown in Figs. 4.12 – 4.21 where the predicted output of the ANFIS models are plotted vs. sample number for the five data sets of the thermal power plant. In addition, the fuzzy rules and the parameters of the models for the thermal power plant data sets are also given.

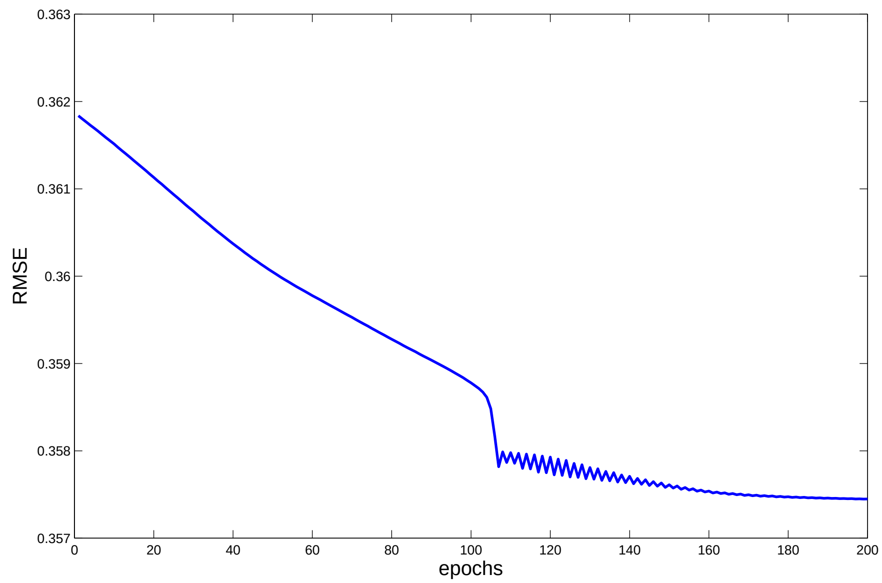

Figure 4.8: RMSE of the ANFIS model during training for Box and Jenkins gas furnace data

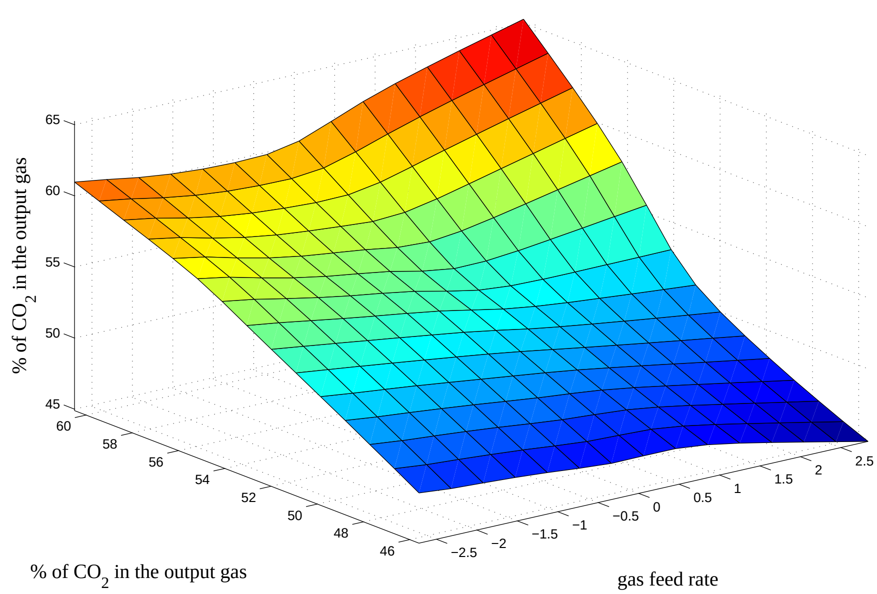

Figure 4.9: Output surface of the data for Box and Jenkins gas furnace example

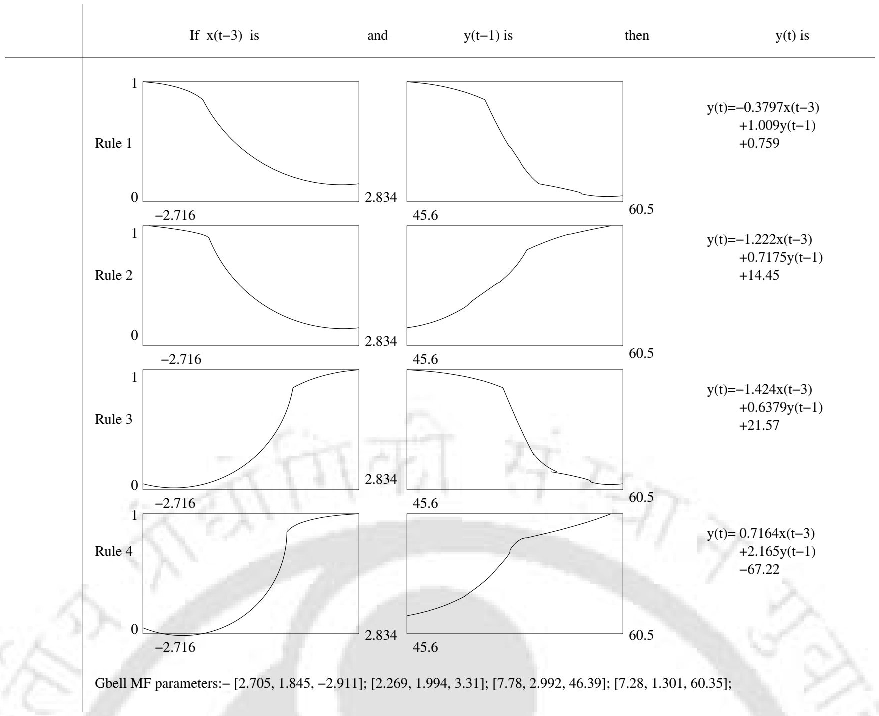

Figure 4.10: Rules for the ANFIS model when trained with half of the data set for the gas furnace example

Case 1: Modeling Results for the first set of the thermal power plant data with ANFIS

In this experiment the data set comprises of 1405 input-output pairs which finally reduced to 1399 effective data pairs as a result of the dynamic process model under consideration [77]. Here the optimally selected input variables are $x(t-6)$ and $y(t-1)$. The number of rules required finally narrowed down to 2 and 4 respectively for the clustering and the grid based method. Fig. 4.12 shows the rules and parameters for this model. In this example the first 699 data pairs are used for training purpose with the remaining half being used for validation. The actual output and the ANFIS model's predicted output for this data set is plotted against sample number as shown in Fig. 4.13.

Case 2: Modeling Results for the second set of the thermal power plant data with ANFIS

In this experiment the data set comprises of 1357 input-output pairs which finally reduced to 1351 effective data pairs [77]. The optimally selected input variables are $x(t-3)$ and $y(t-1)$. The number of rules required finally narrowed down to 2 and 4 respectively for the clustering and the grid partition based method. Fig. 4.14 shows the rules and parameters for this model. In this example the first 675 data pairs are used for training purpose while the other half is used for testing purpose. The modeling result for this data-set is shown in Fig. 4.15.

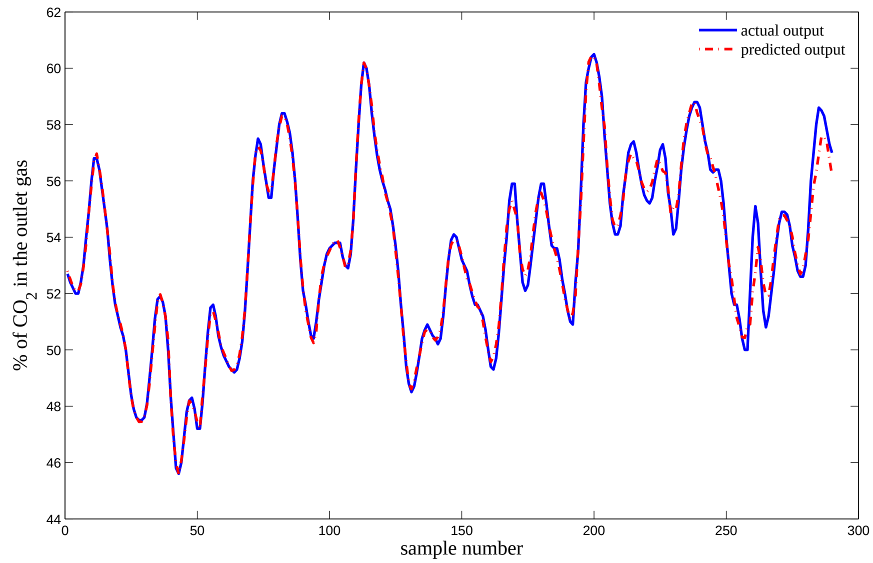

Figure 4.11: Actual and ANFIS model predicted output with Box and Jenkins gas furnace data

Case 3: Modeling Results for the third set of the thermal power plant data with ANFIS

In this experiment the data set comprises of 1432 input-output pairs which finally reduced to 1426 effective data pairs [77]. Here the optimally selected input variables are $x(t-1)$ and $y(t-1)$. The number of rules required finally narrowed down to 3 and 4 respectively for the clustering and the grid based method. Fig. 4.16 shows the rules and parameters for this model. In this example the first half of the 713 data pairs are used for training and the remaining 713 data pairs are used for testing purpose. The modeling result with this data-set is shown in Fig. 4.17.

Case 4: Modeling Results for the fourth set of the thermal power plant data with ANFIS

In this experiment the data set comprises of 994 input-output pairs which finally reduced to 988 effective data pairs as a result of the dynamic process model [77]. Here the optimally selected input variables are $x(t-1)$ and $y(t-1)$. The number of rules required finally narrowed down to 2 and 4 respectively for the clustering and the grid based method. Fig. 4.18 shows the rules and parameters for this model. In this example the first 494 data pairs are used for training the model and the next 494 data pairs are used for testing purpose. The modeling result for this data-set is shown in Fig. 4.19.

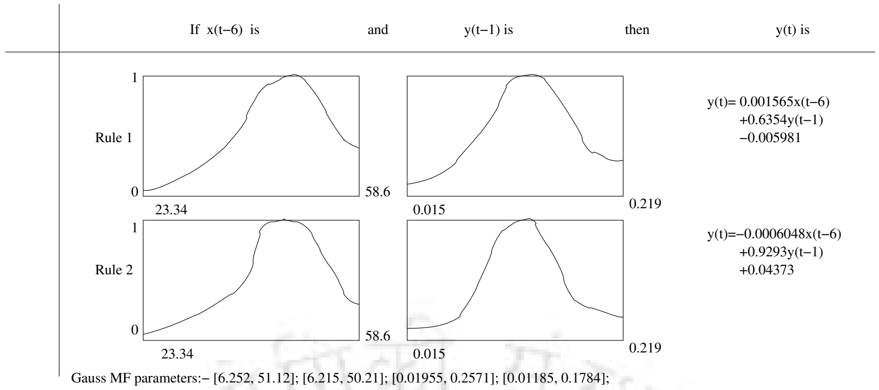

Figure 4.12: Rules for the ANFIS model when trained with half of the data set-I for the thermal power plant example

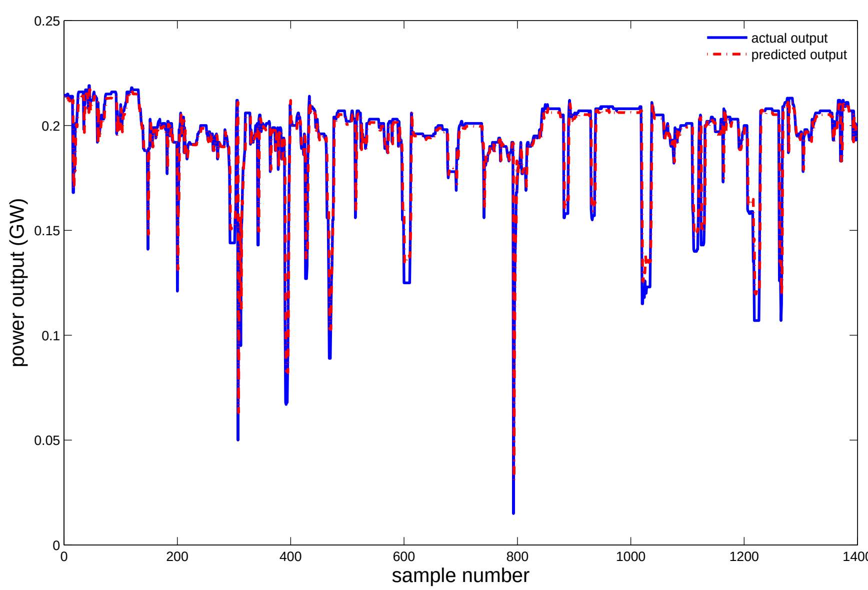

Figure 4.13: Actual and ANFIS model predicted output with the data set-I for thermal power plant

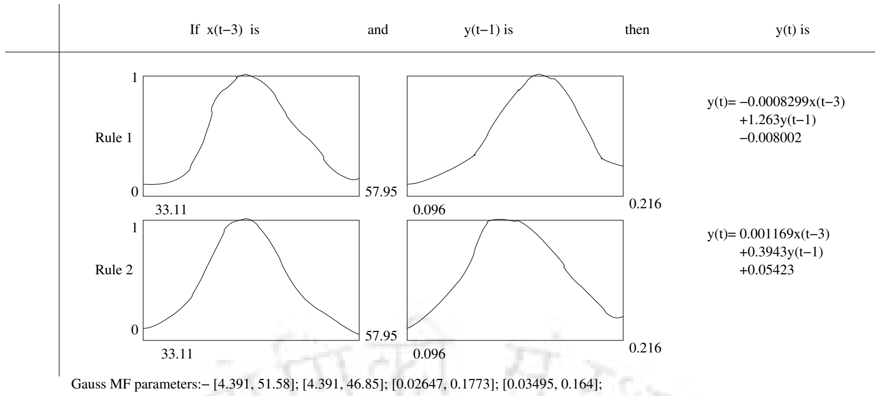

Figure 4.14: Rules for the ANFIS model when trained with half of the data set-II for the thermal power plant example

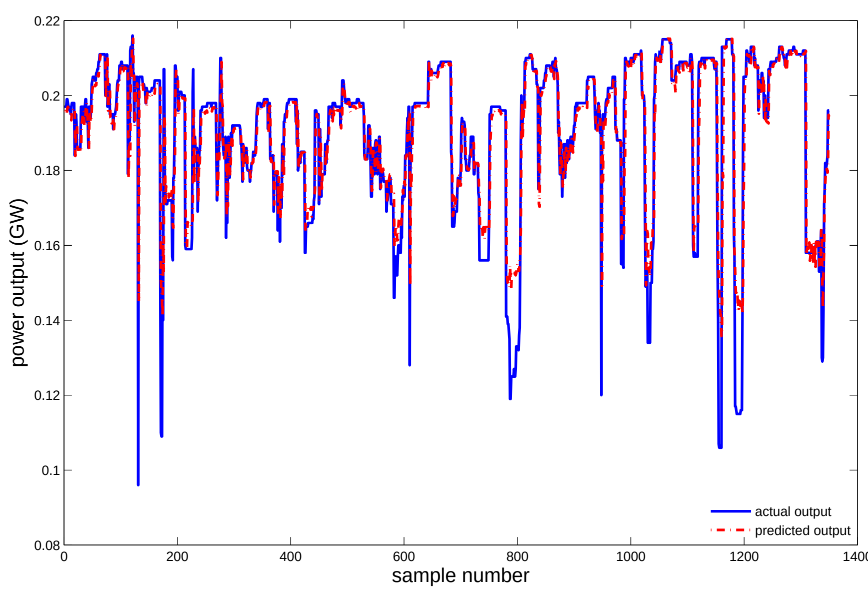

Figure 4.15: Actual and ANFIS model predicted output with the data set-II for thermal power plant

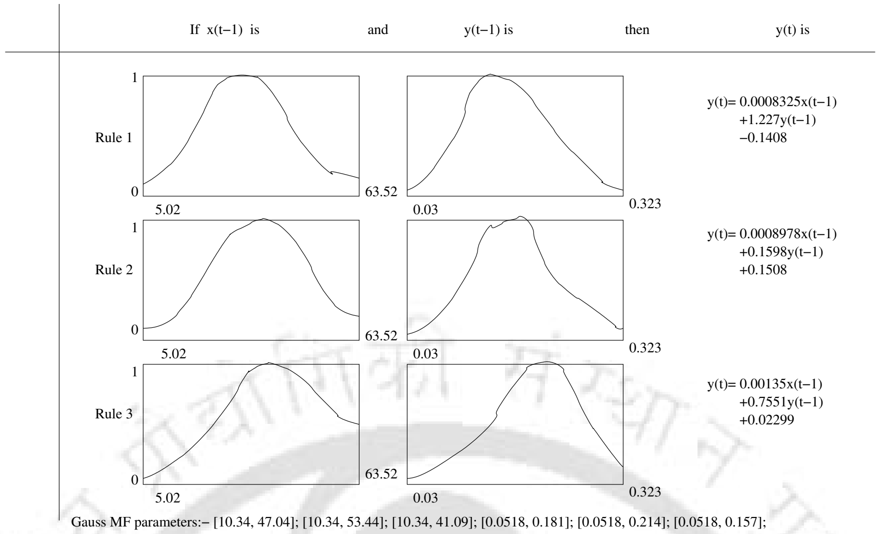

Figure 4.16: Rules for the ANFIS model when trained with half of the data set-III for the thermal power plant example

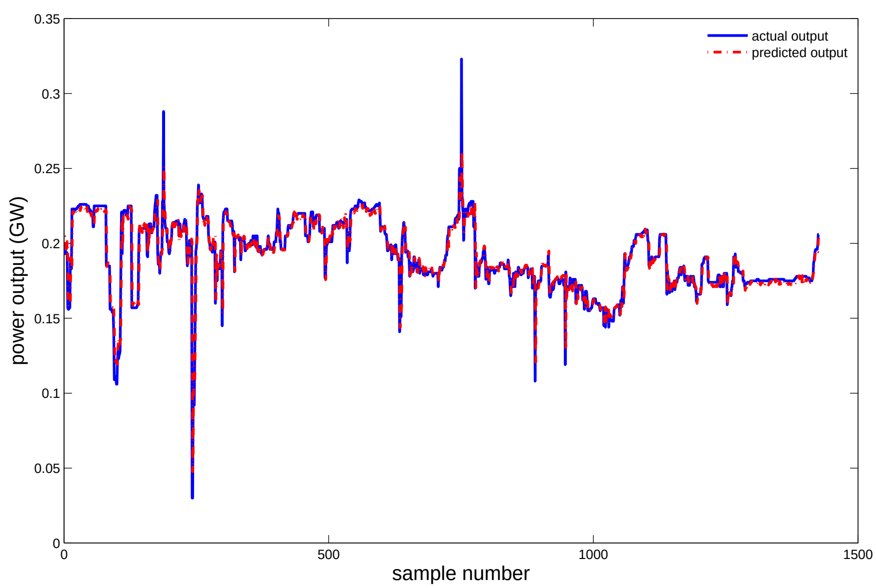

Figure 4.17: Actual and ANFIS model predicted output with the data set-III for thermal power plant

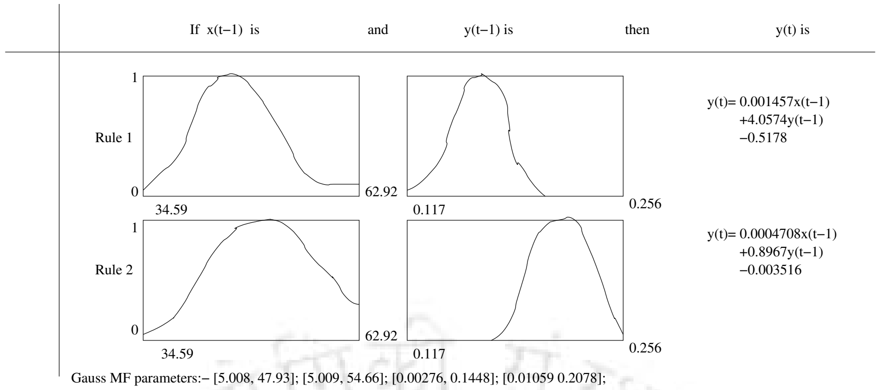

Figure 4.18: Rules for the ANFIS model when trained with half of the data set-IV for the thermal power plant example

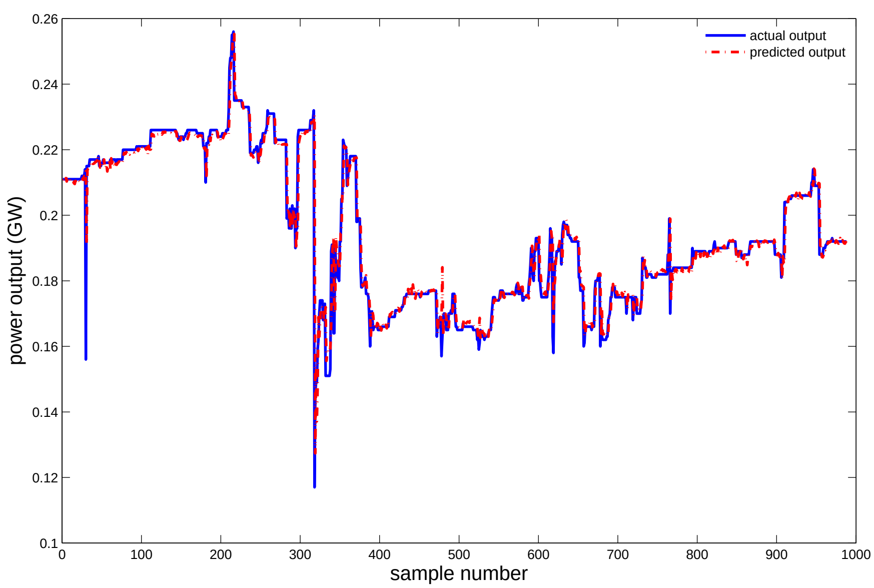

Figure 4.19: Actual and ANFIS model predicted output with the data set-IV for thermal power plant

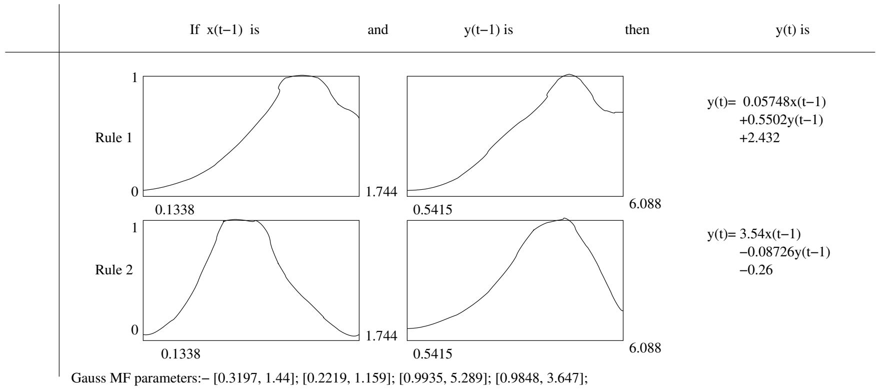

Figure 4.20: Rules for the ANFIS model when trained with half of the data set-V for the thermal power plant example

Case 5: Modeling Results for the fifth set of the thermal power plant data with ANFIS

In this experiment the data set comprises of 645 input-output pairs which finally reduced to 639 effective data pairs as a result of the dynamic process model [77]. Here the optimally selected input variables are $x(t-1)$ and $y(t-1)$. The number of rules required finally narrowed down to 2 and 4 respectively for the clustering and the grid based method. Fig. 4.20 shows the rules and parameters for this model. In this example the first 320 data pairs are used for training the model and the next 319 data pairs are used for validation purpose. The validation result for this data-set are shown in Fig. 4.21.

For the above experiments, initially, the first half of the data set and then the entire data set is used for training the model and the remaining half of the data set is used for validation of the model. The modeling results with our 2-input based ANFIS model using the Box and Jenkins gas furnace and the thermal power plant data are tabulated in Table 4.2. It is observed that when the number of training data is more the RMSE of the test data reduces.

4.6 Conclusions

In this chapter, the neuro-fuzzy modeling of systems using ANFIS has been demonstrated using the input-output data pairs collected from the Box and Jenkins gas furnace example and a thermal power plant. The modeling results obtained by using the gas furnace data are compared with some of the existing results in Table 4.1. In order to keep the same platform for comparing the results with some of

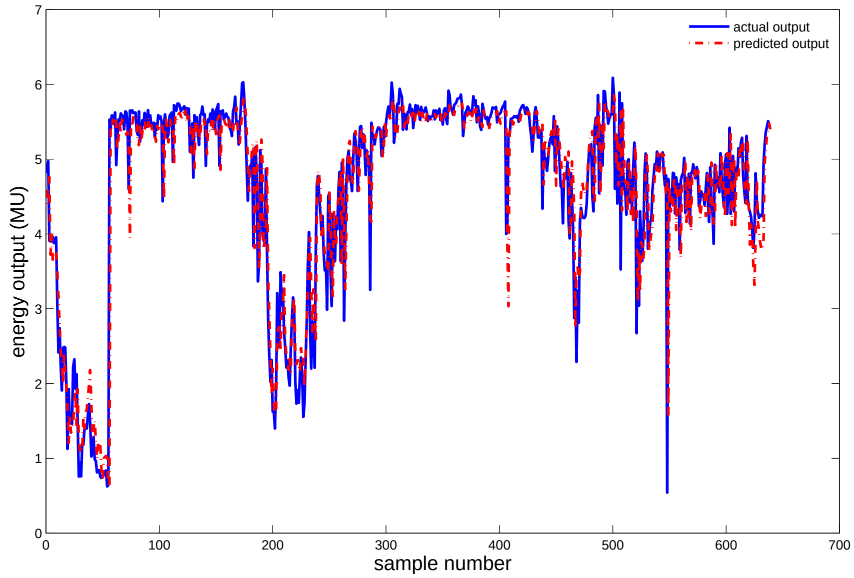

Figure 4.21: Actual and ANFIS model predicted output with the data set-V for thermal power plant

Table 4.2: Modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using ANFIS

Sl. No.	Data	Max No of rules	Input variables	No of training data	RMSE (testing) (Grid partition based)	RMSE (testing) (Subtractive clustering based)
1	Box and Jenkins gas furnace	4	$x(t-3), y(t-1)$	145 290	0.5382 0.3574	0.5724 0.3636
2	Thermal power plant (Data set-I)	4	$x(t-6), y(t-1)$	699 1399	0.0114 0.0108	0.0111 0.0110
3	Thermal power plant (Data set-II)	4	$x(t-3), y(t-1)$	675 1351	0.0118 0.0085	0.0111 0.0089
4	Thermal power plant (Data set-III)	4	$x(t-1), y(t-1)$	713 1426	0.0081 0.0072	0.0076 0.0071
5	Thermal power plant (Data set-IV)	4	$x(t-1), y(t-1)$	494 988	0.0037 0.0033	0.0035 0.0034
6	Thermal power plant (Data set-V)	4	$x(t-1), y(t-1)$	320 639	0.5591 0.4725	0.5361 0.5176

the existing modeling results obtained from the first 7 rows as excerpted from a table in [77], the entire gas furnace data set is used for training the network. From the modeling results shown in Table 4.1, it can be observed that the mean square error (MSE) of the model obtained by using the subtractive clustering based ANFIS is less than the two input Takagi-Sugeno model proposed by Chiu [77]. The MSE for this model obtained by using the grid partitioning based ANFIS model is also less than that obtained by using Chiu's [77] model at the expense of only one more rule. As the addition of one more rule has a very negligible effect on the computational cost, the model may be considered to be comparable to the most accurate model. The modeling results of the Box and Jenkins gas furnace and the thermal power plant are shown in Figs. 4.10 - 4.21. These results show that the 2-input ANFIS model has good prediction capability. Table 4.2 represents the modeling results of the Box and Jenkins gas furnace and the thermal power plant data using at first the initial half and then the entire data set for training. From the results it can be concluded that the RMSE of the ANFIS based model reduces as the number of training data increases. In the ANFIS model, back propagation learning method is used for training the neural network for optimization of the fuzzy rules. A sufficient number of data should be used to guarantee good training. There is still no formula to estimate the number of data required to train a neural network. This number can vary greatly depending on the complexity of the problem and the quality of the data, but many neural networks have been trained successfully with smaller number of data. However, it is not always possible to find the number of training data in abundance. Here the optimal selection of the data set is a major issue which is still a challenge in the field of neural networks. In the present work, an effort is made in that direction keeping in mind the need for the choice of an optimal training data set for the purpose of modeling. In the previous section, the modeling results of the conventional ANFIS model are shown where the training data set comprises of either the entire portion or half of the available data set. The proposed ANFIS model with the optimal choice of the modeling data for training the ANFIS model is the topic of discussion in the next chapter.

TH-801_02610204

Chapter 5

Optimal Data based ANFIS Modeling

In Chapter 4 it has been observed that the performance of the ANFIS based model improves as the number of training data increases. But in many large scale systems the number of available training data is less and the generation of new data is a costly affair. In such a scenario the system has to be modeled with the available limited data only. This chapter proposes three different types of ANFIS based system modeling schemes where the number of data pairs employed for training is minimized by application of the Full factorial design (FFD) technique, the V-fold technique and the combined FFD-V-fold technique in conjunction with ANFIS. Optimal choice of dataset for training is the key step here, subsequent to which the modeling procedure is the the same as that of the ANFIS. These techniques help in selection of the data pairs for training the ANFIS network optimally. The above techniques for optimization of the training data set are described in the subsequent sections.

5.1 Full Factorial Design Based ANFIS Modeling

The statisticians and engineers can make a combined effort to reap the maximum benefit from statistical analysis. For achieving this objective the experiments can be planned in advance to ensure that the proper choice of experimental data can be made in a way that will provide the most unbiased and precise results commensurate with the desired expenditure of time and money. But in many complex systems the number of available training data for modeling is less. In that case, the available data have to be optimally used for training. In this section a system modeling scheme is presented where out of the available data set, a small number of data is critically chosen based on a statistically designed experiment. This statistical design method is called the Full factorial design (FFD) [135, 136]. The full factorial design method is used to select data optimally from the available data set for training the

TH-801_02610204

ANFIS.

There are manifold advantages of statistically designed experiments. Some of the important advantages are as follows:

- As compared to unplanned experimentation more information per experiment can be extracted
- The collection and analysis of information can be done in a more organized manner
- The conclusion from statistically designed experiments is very often evident without extensive statistical analysis
- Credibility is awarded to the conclusions of an experimental analysis when the variability and the sources of experimental error are based on statistical analysis
- These experiments can discover the interaction between experimental variables

For an experimental program involving a large number of tests, the order of selecting the specimens for testing has to be randomized so that each specimen has an equal chance for being selected for testing. The next step is to reduce the large number of possible variables so as to restrict the variables to a few most important ones. The training of the neural network can be visualized as an experiment involving a large number of tests for optimization. An important part of planning an experimental programme is to identify the important variables that affect the response and deciding how to exploit them in the experiment. The experimental variables that are controlled by the investigator are called the factors. The important factors that affect the response have to be identified to use them in the experiment. These factors may be independent i.e the level of one factor may be independent of the levels of the other factors. But the effect on the response of one variable may also depend upon the levels of the other variables due to the interaction of two or more factors.

Factorial designs are experiments in which all levels of each factor in an experiment are combined with all levels of every other factor. In a factorial experiment several factors can be controlled to investigate their effect at each of two or more levels. The experimental design consists of making an observation at each of all possible combinations which can be formed for the different levels of the factors. Every different combination is called a treatment combination.

The simplest and most common type of factorial design is one that uses 2 levels, n factors, i.e. 2^n factorial design. If we consider a 2 level, n factor system, training data have to be selected so that they lie at each of the corners of a n -dimensional space. So in a 2^3 factorial design the training data should

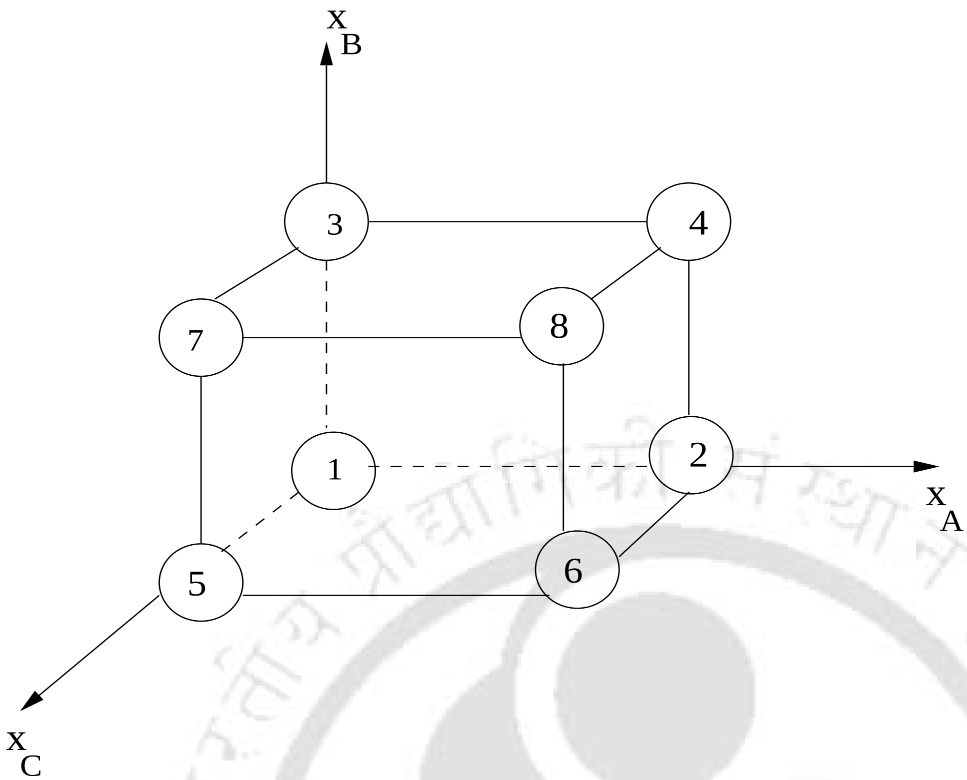

Figure 5.1: Selection of training data in a 2^3 factorial design

be selected such that they lie at each corner of a 3-dimensional hypercube as illustrated in Fig. 5.1. A 2^n factorial design is the simplest type of factorial design as it uses two levels and thereby reduces the number of experimental conditions. But the disadvantage that is associated with a two level factorial design is its inability to distinguish between linear and higher order effects. So the number of levels in a factorial design experiment has to be judiciously chosen.

In addition to the number of levels, the Effect of a factor (EOF) has a significant bearing on the modelling outcome. Effect of a factor may be defined as a change in response produced by a change in the level of a factor and is given by,

$$EOF\ (x_i)\ = \frac{\sum \text{responses at high } x_i - \sum \text{responses at low } x_i}{\text{Half the number of runs in experiment}} \tag{5.1.1}$$

where x_i is a factor. The values of the factors corresponding to the level are called its responses. If m is the number of levels and n is the number of factors, then there will be m^n number of runs in the experiment.

In the present experiments only two factors have been considered where the input and the output are

TH-801_02610204

the two factors. The different levels that have been considered in the experiments are *viz* 2, 3 and 4. When 3 level experiment is considered then there are $3^2 = 9$ number of runs of the experiment. Once the runs for the factors are decided, the corresponding responses are divided into two equal groups in descending order of their values. The differences between the sum of the responses of each individual group determines the numerator of the EOF. This value, when divided by half the number of runs fixes the value of the EOF.

So a parameter having a higher EOF should have a greater representation on the training data set. This is done by increasing the number of data related to that parameter in the training data set. As the modeling of all the ANFIS based models has been carried out with two inputs, the FFD experiments are performed with two factors only with different experimental levels. The FFD based ANFIS model chosen for modeling the complex system is the one with the level having the least RMSE for its modeling result. The different experimental levels that have been used in the modeling schemes for building the test ANFIS network are as follows:

- 2-level full factorial design

  At first only two levels viz the minimum and the maximum of the input-output data pairs are considered. Therefore with full factorial design at first only $2^2 = 4$ data pairs are selected. These 4 data pairs are taken around the minimum and the maximum input-output pairs with equal representation (2 pairs around minimum and 2 pairs around maximum).

- 3-level(1) full factorial design

  Now one more level in the form of the mid-value of the data range is added to the former two levels. So now the 3 levels are the minimum, the maximum and the middle values of the data range. It leads to factorial design of $3^2 = 9$ data pairs for the first run.

- 3-level(2) full factorial design

  For another 3-level combination the cluster center is added instead of the mid-value of the data range in the previous experiment. It again leads to factorial design of $3^2 = 9$ data pairs initially.

- 4-level full factorial design

  Another experiment is performed by taking all the 4 different levels of the dataset viz the maximum and the minimum values, the middle value of the data range and the cluster center. This

TH-801_02610204

leads to factorial design of $4^2 = 16$ data pairs for the first run.

5.2 V-Fold Technique Based ANFIS Modelling

This section presents another ANFIS based complex system modelling method where the number of data pairs employed for training the ANFIS network can be chosen by application of a technique called the V-fold technique [137–139]. This method can be used when the number of available training data are less.

V-fold cross validation technique is a highly accurate method for training a neural network and it has the advantage of not requiring a separate, independent dataset for assessing the accuracy of the neural network. If a network is built using a specific learning dataset, it is necessary to have test data samples independent of the learning dataset that was used to train the network. However, it is difficult or expensive to obtain independent test data frequently and moreover it is undesirable to hold back data from the learning dataset to use for a separate test because that weakens the learning dataset. V-fold cross validation is a technique for performing independent tests without requiring separate test datasets and without reducing the data used to build the network.

The general idea of this method is to divide the overall sample into a number of folds, say V. The same type of analysis is then successively applied to the observations belonging to all the V-1 folds (training samples), and the results of the analysis are applied to the testing sample, which is the Vth sample (the sample or fold that was not used). This is repeated until V random samples are drawn from the data for the analysis. The results for the V replications are aggregated (averaged) to yield a single measure of the stability of the respective model, i.e. the validity of the model for predicting new observations. Thus, this technique allows the analyst to evaluate the overall accuracy of the respective prediction model or method in repeatedly drawn random samples. Here the training and validation process using the V-fold technique for building an ANFIS model process are accomplished as follows :

The learning dataset is partitioned into a number of groups called “folds”. The partitioning is done using stratification methods so that the distribution of categories of the target variable are approximately the same in the partitioned groups. In “V-fold cross classification” V is the number of groups that the dataset is partitioned into. Research has established that 10 partitions are optimum and using more than 10 partitions does not yield any significant improvement in results [137–139]. As such 10 partitions are formed in the experiment. Out of these 10 partitions, 9 are grouped into a new pseudo-learning dataset. An ANFIS model is built using this pseudo-learning dataset. The quality of the network built with this new pseudo-learning data set will in general, be a little inferior to the model obtained by fitting the full learning dataset, because only 90% of the data is used to build it. Since 10% (1 out of 10 partitions) of the data is held back from being used by the network, it can be used as an independent test sample for the ANFIS. The 10% of the data that was held back when the test ANFIS was built is now run through the test network and the classification error for that data is computed.

A different set of 9 partitions is now collected into a new pseudo-learning dataset. The partition that was held back previously is selected this time so that it is different from the partition held back for the first test ANFIS. A second ANFIS is built and its classification error is computed using the data that was held back when it was built. This process is repeated 10 times, building 10 separate networks. In each case, 90% of the data is used to build the network and 10% is held back for independent testing. A different 10% is held back for each test network. The V-fold based ANFIS model showing the least modeling error is considered as the final model for the system.

5.3 FFD-V-fold Based ANFIS Modeling

In this proposed modeling scheme the full factorial design and the V-fold methods have been combined together to generate the optimal data set for training the ANFIS model. This model is basically a V-fold based ANFIS model where the full factorial design method is used to determine the data for the different folds. The use of the full factorial design method also ensures that similar kind of data are selected to the same folds of the V-fold based ANFIS model. This combined method presents a systematic way of choosing and grouping the data for forming the different folds for the V-fold based ANFIS model. The flow chart for this method is drawn in Fig. 5.2.

5.4 Experimental Results and Discussion

In the experiment the same data sets which were used in the previous chapter have been used for modeling. These data are the Box and Jenkins gas furnace data and the NEEPCO thermal power plant data. From these available data sets the data for training the subtractive clustering and the grid partition based ANFIS models are chosen optimally by applying the Full factorial design and the V-fold based techniques. While using the subtractive clustering method a cluster radius of 0.5 was selected to extract the initial fuzzy model. The results and observations have been presented in the subsequent subsections.

TH-801_02610204

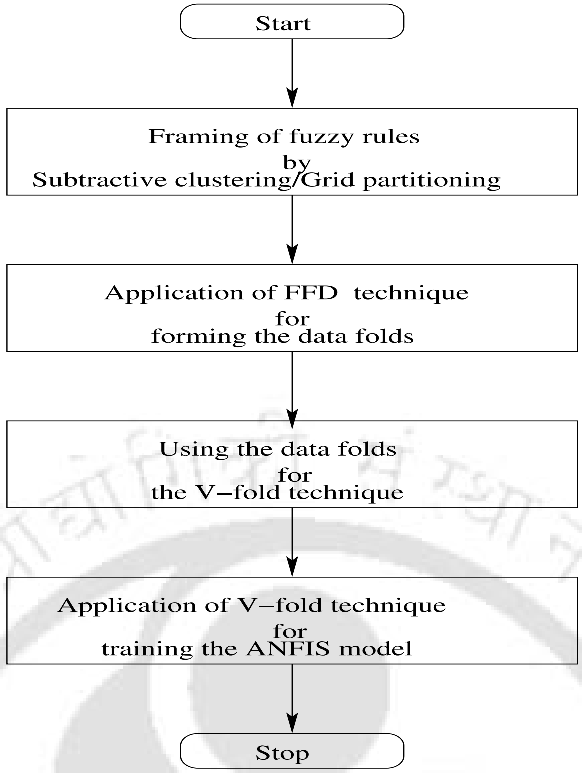

Figure 5.2: Flow Chart for the FFD-V-fold Technique

5.4.1 Modeling results with Full factorial design based ANFIS model

The optimally selected data obtained by applying the full factorial design technique have been used in these experiments to train the ANFIS model. While training the ANFIS model, the effect of selecting an input variable x_i corresponding to different levels in the range of operation of the input variable is investigated. Considering the input-output pairs (x_i, y_i) and 2, 3 and 4 levels of operation, the obtained factorially designed data are2^2, 3^2 and 4^2 respectively for use in training. These experiments are described below:

Case 1:- Modeling with Box and Jenkins gas furnace data

The Box and Jenkins gas furnace data [102] are used in this experiment. Optimal choice of dataset for training is the key step here subsequent to which the modelling procedure is the same as that of ANFIS. In this case, $x(t-3)$ and $y(t-1)$ were found to be the optimal input variables by using the RMSE criterion. So these two inputs are used in the ANFIS model. The method for choosing the number of modeling data for the different levels of the FFD based models is explained below.

- 2-level full factorial design

  Considering only the minimum and the maximum values of the input-output data pairs, at first only $2^2 = 4$ data pairs are selected. These 4 data pairs are taken around the minimum and the maximum valued input-output pairs with equal representation (2 pairs around the minimum and 2 pairs around the maximum). Now EOF for both x_i and y_i (from Eq. 5.1.1) are calculated and found to be 1 for both the cases. So 4 (1 each around the minimum and the maximum values of both x_i and y_i) additional data pairs are now needed and these are taken around the minimum and the maximum valued input-output data pairs. Again EOF is calculated and is found to be 1. So 4 additional data pairs are now selected around the minimum and the maximum input-output pairs with equal representation (2 pairs around the minimum and 2 pairs around the maximum) for making a balanced representation of data around the minimum and the maximum values. EOF is again calculated and is found to be 1. Therefore 4 more input-output data pairs are taken around the minimum and the maximum of the input-output pairs. Finally $4+4+4+4 = 16$ data pairs are selected for training.

- 3(1)-level full factorial design

  Now one more level in the form of the mid-value of the data range is added to the former two extreme levels. So now the 3 levels are the minimum, the maximum and the middle value of the data range. It leads to factorial design of $3^2 = 9$ data pairs. Proceeding in a similar manner as in the 2-level case, EOF values of 2, 1 and 1 are obtained for y_i, x_i and both of x_i and y_i respectively, in 3 successive evaluations. So 6 (2 each around y_i of the minimum, the maximum and the centroid), 3 (1 each around x_i of the minimum, the maximum and the centroid) and 6 (1 each around x_i and y_i of the minimum, the maximum, and the centroid) additional data pairs are required. Therefore for this experiment $9+6+3+6 = 24$ data pairs are used for training.

- 3(2)-level full factorial design

  For another 3-level combination the cluster centers are added instead of the mid-value of the data range which was the case in the previous experiment. After three successive evaluations of the $3^2 = 9$ data pairs, EOF values of 1, 1 and 1 are obtained for x_i. Therefore in this factorially designed experiment additional data pairs of 4, 4 and 4 have to be considered, thereby requiring a total of $9+4+4+4 = 21$ data pairs for training.

TH-801_02610204

- 4-level full factorial design

  Another experiment is performed by taking all the 4 different levels of the data set viz the minimum, the maximum, the middle value of the data range the cluster center. After three successive evaluations on the $4^2 = 16$ data pairs, EOF values of 1, 1 and 1 are obtained for y_i, x_i and (x_i, y_i) of the different levels. Therefore with 1 data pair each around the minimum, the maximum, the centroid and each of the two cluster centers for y_i, x_i and (x_i, y_i), a value of 5, 5 and 10 additional data pairs are taken making a total number of $16 + 5 + 5 + 10 = 36$ training data pairs.

Case 2:- Modeling with thermal power plant data

(i) Data set-I

The first set of the NEEPCO's thermal power plant data is used in this experiment. In this case, $x(t-6)$ and $y(t-1)$ are used in the ANFIS model as they were found to be the optimal input variables by using the RMSE criterion. The procedure for choosing the optimal number of training data is given below.

- 2-level full factorial design

  Considering only the minimum and the maximum valued input-output data pairs, only $2^2 = 4$ data pairs are selected initially. After three successive evaluations, EOF values of 1, 1 and 1 were obtained for the two successive $(x_i\ , y_i)$ values and lastly for y_i. Therefore with 4, 4 and 2 additional data pairs, a total of $4 + 4 + 4 + 2 = 14$ input-output pairs are used for training.

- 3(1)-level full factorial design

  Now with the addition of one more level in the form of the mid-value of the data range to the former two levels, a factorial design of $3^2 = 9$ data pairs, comprising of the minimum, the maximum and the middle value of the data range is obtained. Again after two successive evaluations for y_i, the values of EOF obtained are 2 and 3 respectively. So with the inclusion of 6 and 9 additional data pairs, a total of $9 + 6 + 9 = 24$ input-output pairs are used in this experiment for training.

- 3(2)-level full factorial design

  For another 3-level combination the cluster center is added instead of the mid-value of the data range. After three successive evaluations of the $3^2 = 9$ data pairs, EOF values of 1,1 and 2 are obtained for y_i. Therefore in this factorially designed experiment, $9+4+4+8=25$ input-output data pairs are used for training.

- 4-level full factorial design

  Another experiment is performed by taking all the 4 different levels of the dataset, viz the maximum and the minimum values, the middle value of the data range and the cluster center. After two successive evaluations on the $4^2=16$ data pairs, EOF values of 1 and 2 were obtained for (x_i, y_i). So with 10 and 20 additional data pairs, $16+10+20=46$ input-output data pairs are used for training.

(ii) Data set-II

The second set of the thermal power plant data was used in this experiment. In this case, $x(t-3)$ and $y(t-1)$ were found to be the optimal input variables. So these two inputs are used in the ANFIS model. The number of data pairs required for the different levels of the experiment are explained below.

- 2-level full factorial design

  Here at first $2^2=4$ data pairs are selected considering the minimum and the maximum values of the input-output data pairs. With three successive evaluations of EOF, the values obtained are 1, 1 and 1 respectively for (x_i, y_i). So the total number of data used for training are $4+4+4+4=16$.

- 3(1)-level full factorial design

  Starting with $3^2=9$ data pairs initially for the three levels of the maximum, the minimum and the mid-values of the data pairs, the values of EOF obtained are 1, 1 and 1 after three successive evaluations for (x_i, y_i). So a total of $9+6+6+6=27$ data pairs are used for training in the ANFIS model.

- 3(2)-level full factorial design

  With $3^2=9$ data pairs used initially for the minimum, the maximum, and the cluster center levels, after three successive evaluations the values of EOF found are 1, 1 and 1 for x_i, (x_i, y_i) and y_i. This results in a total number of $9+4+8+4=25$ training data pairs.

- 4-level full factorial design

TH-801_02610204

Starting initially with $4^2 = 16$ data pairs and evaluating the EOF after two successive runs, the values of 2 and 1 are obtained for y_i and (x_i, y_i) respectively. Therefore the total number of data pairs used for training is $16 + 10 + 10 = 36$.

(iii) Data set-III

In the third set of the thermal power plant data, $x(t-1)$ and $y(t-1)$ were found to be the optimal inputs. The number of training data chosen for the different models are shown below.

- 2-level full factorial design

  With $2^2 = 4$ data pairs chosen initially and EOF values of 1, 1 and 1 for (x_i, y_i) respectively, the training for this model was carried out with $4 + 4 + 4 + 4 = 16$ data pairs.

- 3(1)-level full factorial design

  Starting with $3^2 = 9$ data pairs and EOF values of 1, 1 and 1 for (x_i, y_i), a total of $9+6+6+6 = 27$ training data are used for training the ANFIS model.

- 3(2)-level full factorial design

  In this experiment, with $3^2 = 9$ data taken initially, the values of EOF are obtained as 1, 1 and 1 respectively for (x_i, y_i). So the total number of data pairs used is $9 + 8 + 8 + 8 = 33$.

- 4-level full factorial design

  With $4^2 = 16$ data pairs selected initially, the evaluated values of EOF are 1, 1 and 1 for input-output pair (x_i, y_i). So the total number of data pairs used in training is $16 + 10 + 10 + 10 = 46$.

(iv) Data Set-IV

In the fourth set of the thermal power plant data, $x(t-1)$ and $y(t-1)$ were found to be optimal inputs. The number of modeling data chosen for the different models are as follows.

- 2-level full factorial design

  With $2^2 = 4$ data pairs selected initially and computed EOF values of 1, 1 and 2 for two successive evaluations of (x_i, y_i) and y_i, the total number of data pairs used in training the ANFIS model is $4 + 4 + 4 + 4 = 16$.

- 3(1)-level full factorial design

  Starting with $3^2 = 9$ pairs of initial data and corresponding EOF values of 1, 1 and 1 for (x_i, y_i), the total number of data pairs needed for training is $9 + 6 + 6 + 6 = 27$.

- 3(2)-level full factorial design

  For this experiment, EOF values of 1, 1 and 1 were obtained for $3^2 = 9$ initial data for y_i and then for two successive input-output pairs (x_i, y_i). Hence the number of data pairs required for training is $9 + 4 + 8 + 8 = 29$.

- 4-level full factorial design

  For this case, the values of EOF for x_i, (x_i, y_i) and y_i are 1, 1 and 1 for $4^2 = 16$ input-output data pairs. Therefore the ANFIS model requires $16 + 5 + 10 + 5 = 36$ data pairs.

(V) Data set-V

In the fifth set of the thermal power plant data, $x(t-1)$ and $y(t-1)$ were found to be optimal and hence are used in our model.

- 2-level full factorial design

  Using $2^2 = 4$ number of data initially, the EOF values obtained are 1, 1 and 1 for (x_i, y_i). This resulted in the requirement of $4 + 4 + 4 + 4 = 16$ data pairs for training.

- 3(1)-level full factorial design

  The calculated values of EOF obtained for $3^2 = 9$ data pairs for two successive y_i and (x_i, y_i) are 2, 2 and 1 respectively . So $9 + 6 + 6 + 6 = 27$ data pairs are needed for training.

- 3(2)-level full factorial design

  With $3^2 = 9$ initial data pairs, the values of EOF obtained are 1, 1 and 1 for (x_i, y_i). For these values of EOF, the total number of data pairs required for training is $9 + 8 + 8 + 8 = 33$.

- 4-level full factorial design

  Here $4^2 = 16$ input-output data pairs are chosen initially. Corresponding values of EOF are 1, 1 and 1 for (x_i, y_i). Therefore the number of data pairs needed for training is $16+10+10+10 = 46$.

The modeling results so obtained with the FFD based ANFIS model for the Box and Jenkins gas furnace and the thermal power plant data are tabulated in Table 5.1. Here the ANFIS model used is of 2-input 4-rule type with the FFD based optimally selected data for training. The second half of the data set is used for validation of the model.

TH-801_02610204

Table 5.1: Modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using FFD based ANFIS

Sl. No.	Data	Input variables	No of rules	Model FFD	No of training data	RMSE (testing) (Grid partition based)	RMSE (testing) (Subtractive clustering based)
1	Box and Jenkins gas furnace	$x(t-3), y(t-1)$	4	2-level	16	0.6473	0.6437
				3(1)-level	24	0.7254	0.6875
				3(2)-level	21	0.5607	0.6229
				4-level	36	0.5589	0.5972
2	Thermal power plant (Data set-I)	$x(t-6), y(t-1)$	4	2-level	14	0.0264	0.0275
				3(1)-level	24	0.0152	0.0205
				3(2)-level	25	0.0216	0.0117
				4-level	46	0.0151	0.0125
3	Thermal power plant (Data set-II)	$x(t-3), y(t-1)$	4	2-level	16	0.0175	0.0200
				3(1)-level	27	0.0125	0.0095
				3(2)-level	25	0.0140	0.0119
				4-level	36	0.0097	0.0090
4	Thermal power plant (Data set-III)	$x(t-1), y(t-1)$	4	2-level	16	0.0107	0.0175
				3(1)-level	27	0.0109	0.0106
				3(2)-level	33	0.0090	0.0083
				4-level	46	0.0099	0.0095
5	Thermal power plant (Data set-IV)	$x(t-1), y(t-1)$	4	2-level	16	0.0038	0.0036
				3(1)-level	27	0.0041	0.0036
				3(2)-level	29	0.0036	0.0035
				4-level	36	0.0061	0.0053
6	Thermal power plant (Data set-V)	$x(t-1), y(t-1)$	4	2-level	16	0.5890	0.9613
				3(1)-level	27	0.6956	0.5714
				3(2)-level	33	1.0259	0.5347
				4-level	46	0.8392	0.9149

TH-801_02610204

5.4.2 V-fold technique based ANFIS model

In the following experiment the V-fold method is used to choose the data set for training the ANFIS model. After the selection of the data for the different folds, which is around one-eighth of the training data used in the conventional model, the results so obtained are presented below.

Case 1:- Modeling with Box and Jenkins gas furnace data

The data set for the Box and Jenkins gas furnace example is again used for modeling the V-fold based ANFIS. The optimal inputs so chosen are $x(t-3)$ and $y(t-1)$ along with a cluster center of 0.5. The different folds for applying the V-fold technique were constituted with 18 data pairs which is around one-eighth of the 145 input-output pairs used for training the conventional ANFIS model.

Case 2:- Modeling with thermal power plant data

(i) Data set-I

With $x(t-6)$ and $y(t-1)$ as the optimal inputs and a cluster center of 0.5 for the subtractive clustering method, the modeling is carried out with 81 data pairs, which is around one-ninth of the 699 data pairs required in the conventional ANFIS model.

(ii) Data set-II

Here modeling is carried out with 72 data pairs which is around one-ninth of the 675 input-output data pairs . The optimal inputs are $x(t-3)$ and $y(t-1)$.

(iii) Data set-III

The optimal inputs for this model are $x(t-1)$ and $y(t-1)$. For this data set the V-fold based ANFIS model is built with about one-tenth of the 713 input-output data pairs i.e 72 data pairs.

(iv) Data set-IV

For this data set, $x(t-1)$ and $y(t-1)$ are found to be the optimal inputs. The V-fold based ANFIS model is built with 72 data pairs which is around one-seventh of 494 data pairs.

(v) Data set-V

Here the optimal inputs are $x(t-1)$ and $y(t-1)$. For the purpose of modeling, around one-ninth of 320 i.e 36 input-output data pairs are chosen.

The modeling results obtained by using the V-fold technique based ANFIS model are tabulated in Table 5.2.

TH-801_02610204

Table 5.2: Modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using V-fold based ANFIS model

Sl. No.	Data	No of training data	Input variables	Number of rules	RMSE (testing)	
					(Grid partition based)	(Subtractive clustering based)
1	Box and Jenkins gas furnace	18	$x(t-3), y(t-1)$	4	0.5943	0.5556
2	Thermal power plant (Data set-I)	81	$x(t-6), y(t-1)$	4	0.0110	0.0115
3	Thermal power plant (Data set-II)	72	$x(t-3), y(t-1)$	4	0.0092	0.0104
4	Thermal power plant (Data set-III)	72	$x(t-1), y(t-1)$	4	0.0078	0.0082
5	Thermal power plant (Data set-IV)	72	$x(t-1), y(t-1)$	4	0.0034	0.0036
6	Thermal power plant (Data set-V)	36	$x(t-1), y(t-1)$	4	0.6508	0.6097

5.4.3 Modeling results with FFD-V-fold technique based ANFIS model

The V-fold technique is now combined with the FFD based ANFIS. The following experiments are carried out with this combined model.

Case 1:- Modeling results for Box and Jenkins gas furnace data

The optimally selected data set obtained from the Box and Jenkins gas furnace data set is used for modeling at various levels like 2-level, 3(1)-level, 3(2)-level and 4-level. The modeling is carried out by using both the grid partition based method and subtractive clustering technique by choosing a cluster radius of 0.5. The best modeling result is observed in the case of the 2-level FFD-V-fold based ANFIS model both for the grid and the subtractive clustering based models by using $x(t-3)$ and $y(t-1)$ as the two optimal inputs.

Case 2:- Modeling results for the thermal power plant data

(i) Data set-I

Here the modeling is carried out by using $x(t-6)$ and $y(t-1)$ as the two optimal inputs. For this data set the best result was obtained for the 2-level FFD-V-fold based ANFIS model.

(ii) Data set-II

The two optimal inputs used here are $x(t-3)$ and $y(t-1)$. The 3(1)-level FFD-V-fold based ANFIS model produced the best result.

(iii) Data set-III

The optimal inputs used here for modeling are $x(t-1)$ and $y(t-1)$. The FFD-V-fold based ANFIS model with the 3(2) level showed the least MSE.

(iv) Data set-IV

Here $x(t-1)$ and $y(t-1)$ are the optimal inputs used for modeling and the least MSE was obtained for the 2-level FFD-V-fold based ANFIS model.

(vi) Data set-V

Here the modeling is carried out by using the optimal inputs $x(t-1)$ and $y(t-1)$. The 3(1)-level FFD-V-fold based ANFIS model shows the least MSE.

The results obtained by using the combined FFD-V-fold technique based ANFIS model are tabulated in Table 5.3. In Table 5.4, the results obtained by using the different models viz conventional ANFIS, FFD based ANFIS, V-fold based ANFIS and FFD-V-fold based ANFIS are compared with respect to their RMSE. From Table 5.4, it is observed that the best modeling results are obtained with the FFD-V-fold based ANFIS model as the RMSE (testing) is the least in this case for all the different types of data. The modeling results for the different data sets as obtained by using different modeling methodologies are shown in Figs. 5.3 - 5.8. In Fig. 5.3 the actual output is plotted vs the predicted output for the FFD-V-fold based ANFIS model using the Box and Jenkins gas furnace data. Fig. 5.4 shows the actual and the predicted output for the FFD-V-fold based ANFIS model using the data set-I for the thermal power plant. Fig. 5.5 shows the actual and the predicted output for the FFD-V-fold based ANFIS model using the data set-II for the thermal power plant. Fig. 5.6 shows the actual and the predicted output for the FFD-V-fold based ANFIS model using the data set-III for the thermal power plant. Fig. 5.7 shows the actual and the predicted output for the FFD-V-fold based ANFIS model using the data set-IV for the thermal power plant. Fig. 5.8 shows the actual and the predicted output for the FFD-V-fold based ANFIS model using the data set-V of the thermal power plant.

TH-801_02610204

Table 5.3: Modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using FFD-V-fold based ANFIS

Sl. No.	Data	Input variables	No of rules	Model FFD-V-fold	No of training data	RMSE (testing) (Grid partition based)	RMSE (testing) (Subtractive clustering based)
1	Box and Jenkins gas furnace	$x(t-3), y(t-1)$	4	2-level	18	0.5378	0.5332
				3(1)-level	18	0.5705	0.6017
				3(2)-level	18	0.6153	0.5373
				4-level	18	0.5627	0.5684
2	Thermal power plant (Data set-I)	$x(t-6), y(t-1)$	4	2-level	81	0.0110	0.0108
				3(1)-level	81	0.0116	0.0109
				3(2)-level	81	0.0109	0.0112
				4-level	81	0.0112	0.0114
3	Thermal power plant (Data set-II)	$x(t-3), y(t-1)$	4	2-level	72	0.0092	0.0092
				3(1)-level	72	0.0086	0.0089
				3(2)-level	72	0.0091	0.0093
				4-level	72	0.0088	0.0091
4	Thermal power plant (Data set-III)	$x(t-1), y(t-1)$	4	2-level	72	0.0084	0.0079
				3(1)-level	72	0.0082	0.0080
				3(2)-level	72	0.0076	0.0075
				4-level	72	0.0083	0.0082
5	Thermal power plant (Data set-IV)	$x(t-1), y(t-1)$	4	2-level	72	0.0033	0.0034
				3(1)-level	72	0.0036	0.0035
				3(2)-level	72	0.0035	0.0035
				4-level	72	0.0035	0.0036
6	Thermal power plant (Data set-V)	$x(t-1), y(t-1)$	4	2-level	36	0.5740	0.5293
				3(1)-level	36	0.5687	0.5118
				3(2)-level	36	0.6651	0.5900
				4-level	36	0.5174	0.5312

TH-801_02610204

Table 5.4: Comparison of modeling results for the Box and Jenkins gas furnace data and the thermal power plant data using various ANFIS models

Sl. No.	Data	Input variables	No of rules	Model	No of training data	RMSE (testing)	
						(Grid partition based)	(Subtractive clustering based)
1	Box and Jenkins gas furnace	$x(t-3), y(t-1)$	4	ANFIS FFD V-Fold FFD-V-fold	145 18 18 18	0.5382 0.5589 0.5943 0.5378	0.5724 0.5972 0.5556 0.5332
2	Thermal power plant (Data set-I)	$x(t-6), y(t-1)$	4	ANFIS FFD V-Fold FFD-V-fold	699 $46, 25$ 81 81	0.0114 0.0151 0.0110 0.0109	0.0111 0.0117 0.0115 0.0108
3	Thermal power plant (Data set-II)	$x(t-3), y(t-1)$	4	ANFIS FFD V-Fold FFD-V-fold	675 36 72 72	0.0118 0.0097 0.0092 0.0086	0.0111 0.0090 0.0104 0.0089
4	Thermal power plant (Data set-III)	$x(t-1), y(t-1)$	4	ANFIS FFD V-Fold FFD-V-fold	713 27 72 72	0.0081 0.0090 0.0078 0.0076	0.0076 0.0083 0.0082 0.0075
5	Thermal power plant (Data set-IV)	$x(t-1), y(t-1)$	4	ANFIS FFD V-Fold FFD-V-fold	494 29 72 72	0.0037 0.0036 0.0034 0.0033	0.0035 0.0035 0.0036 0.0034
6	Thermal power plant (Data set-V)	$x(t-1), y(t-1)$	4	ANFIS FFD V-Fold FFD-V-fold	319 $16, 27$ 36 36	0.5591 0.5890 0.6508 0.5174	0.5361 0.5347 0.6097 0.5118

TH-801_02610204

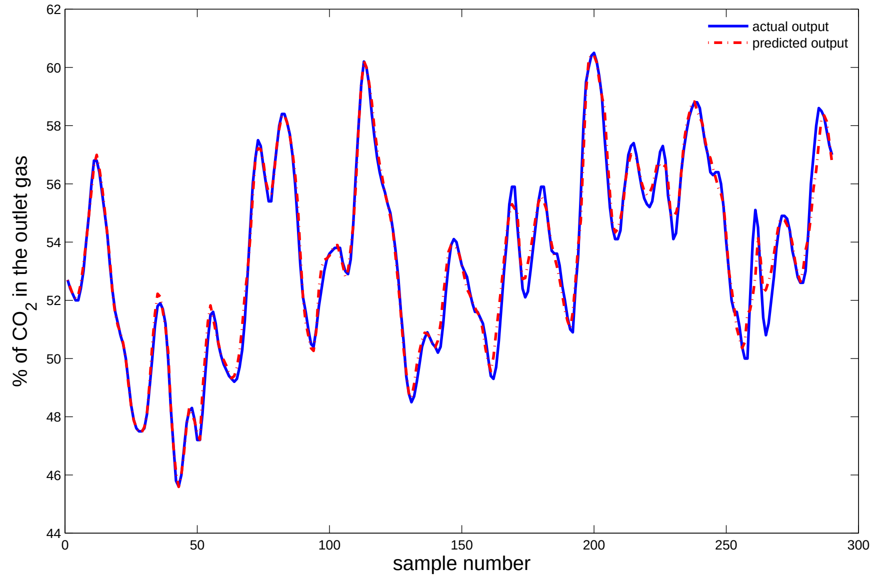

Figure 5.3: Actual and FFD-V-fold based ANFIS model predicted output with the Box and Jenkins gas furnace data

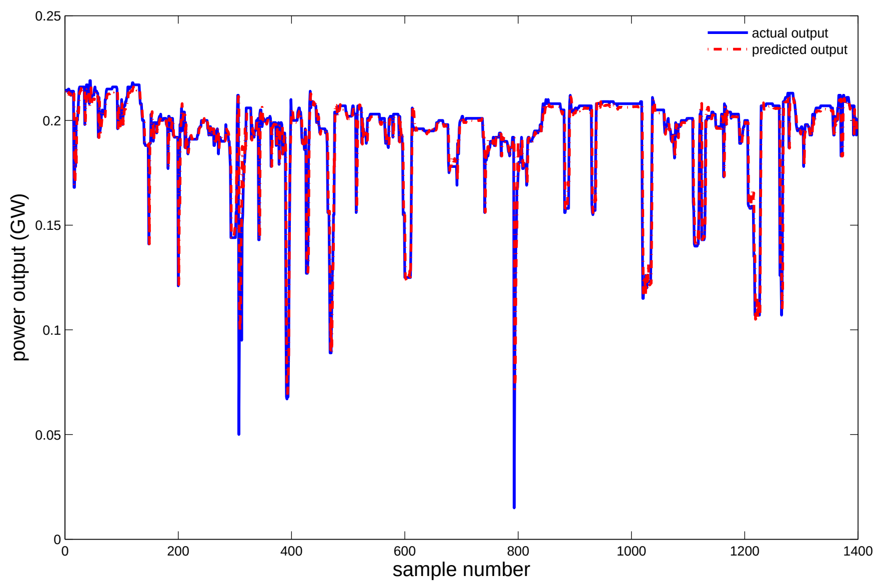

Figure 5.4: Actual and FFD-V-fold based ANFIS model predicted output with data set-I of thermal power plant

TH-801_02610204

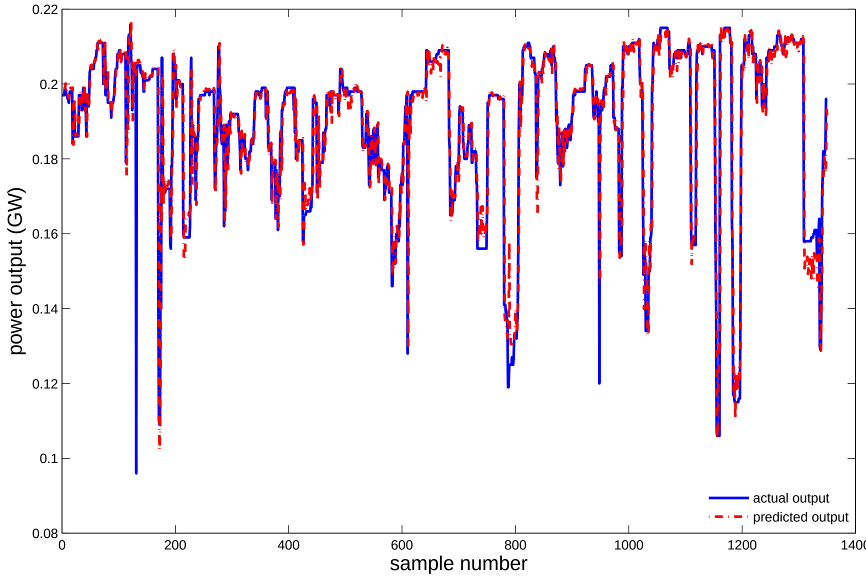

Figure 5.5: Actual and FFD-V-fold based ANFIS model predicted output with data set-II of thermal power plant

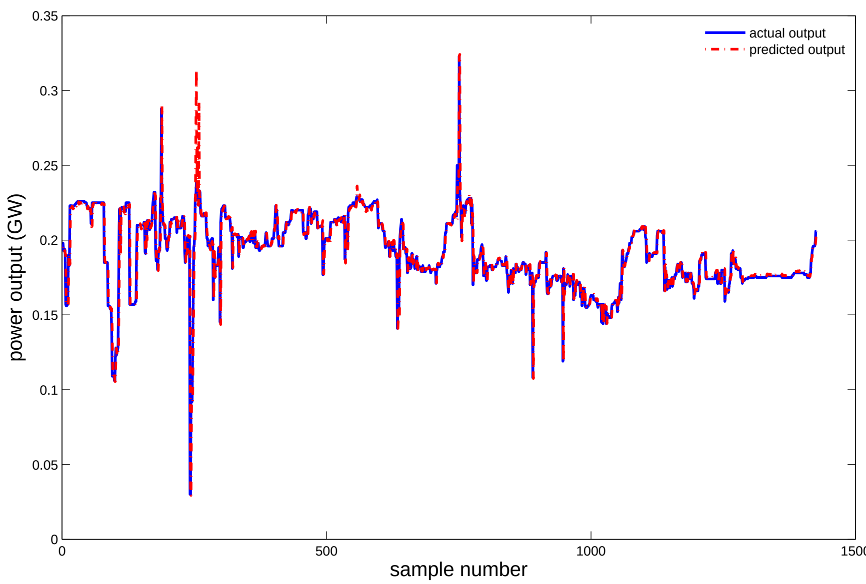

Figure 5.6: Actual and FFD-V-fold based ANFIS model predicted output with data set-III of thermal power plant

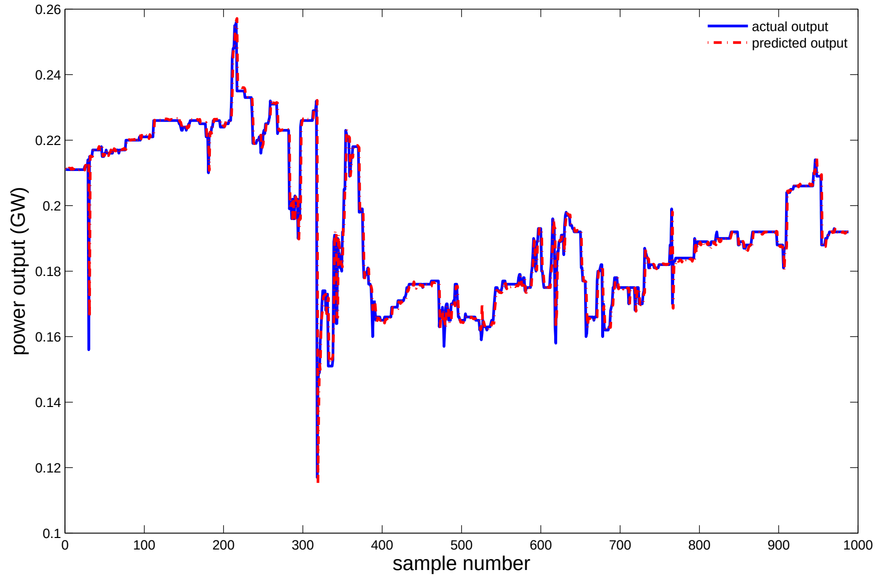

Figure 5.7: Actual and FFD-V-fold based ANFIS model predicted output with data set-IV of thermal power plant

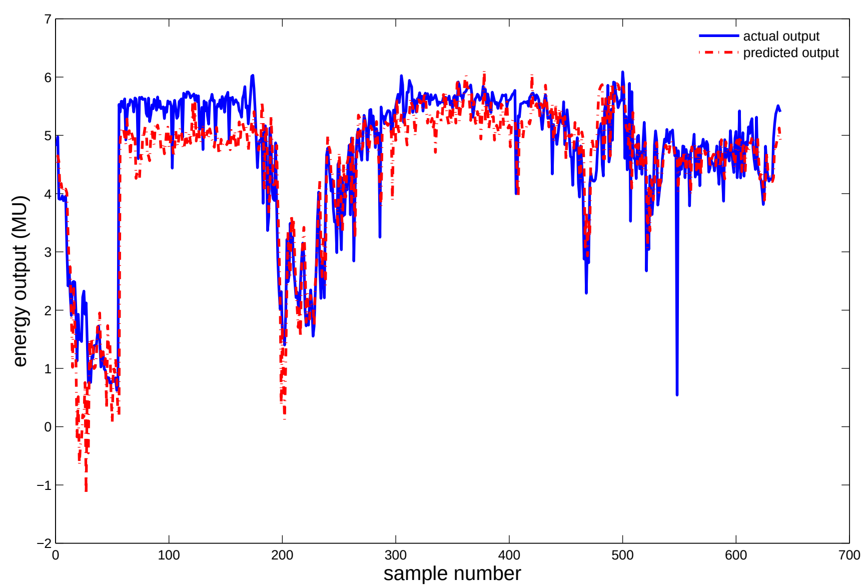

Figure 5.8: Actual and FFD-V-fold based ANFIS model predicted output with data set-V of thermal power plant

5.5 Conclusion

From Table 5.4 it is observed that the FFD-V-fold based ANFIS model shows the best performance even though only around one-eighth of the dataset used in the conventional ANFIS model has been selected for training. The FFD-V-fold based ANFIS model was built on the basis of optimally chosen data for modeling. This shows that the ANFIS model based on the training data set selected by using the FFD-V-fold technique shows the best prediction capability. To further test the performance of the proposed FFD-V-fold based ANFIS model, the modeling results of this chapter are compared with standard statistical models as well as another soft computing based model in the next chapter. The two types of models that are chosen for the comparative analysis are the statistical model and the GA based fuzzy model.

TH-801_02610204

Chapter 6

Statistical Models and Genetic Algorithm Based Fuzzy Model

From the preceding chapters it is observed that by using an optimally selected training data set, the performance of the conventional ANFIS model can be enhanced. The proposed FFD-V-fold technique augments the conventional ANFIS model and it even outperforms the conventional ANFIS model with lesser RMSE but utilizing far fewer training data. In order to determine the efficacy of this proposed FFD-V-fold based ANFIS model, it is compared with conventional statistical models like ARX, ARMAX and ARIMAX in this chapter. The proposed model is also compared with the genetic algorithm (GA) based fuzzy model which is a new technique in the emerging area of artificial intelligence. Lastly a controller is designed using the proposed FFD-V-fold based ANFIS model and its performance is studied.

6.1 Statistical Models

Engineers and scientists have been using probability and statistics as a working tool in many areas of engineering practices. Most often, during engineering design, many problems crop up due to poorly defined situation or having to use data with low precision. This problem can be solved by using statistical models [140, 146] by proper application of statistical analysis. The statistical models help in making important decisions in engineering. Here the observed samples are used to estimate the statistical population whose properties provide the basis for decision making. The results from the analysis of experimental data can be unequivocally described by appropriate statistical parameters. Many of the techniques used for data analysis are based on univariate and multivariate statistics.

TH-801_02610204

In this chapter some of the existing statistical models are fitted into the systems under study and they are compared with the models proposed in the previous chapters. Some of the statistical models which are used for comparison with the soft computing based models proposed in the previous chapter are described in the successive subsections. The statistical models that have been used for modeling are the Auto regressive with exogenous input (ARX) model, Auto regressive moving average with exogenous input (ARMAX) model and Auto regressive integrated moving average with exogenous input (ARIMAX) model.

6.1.1 Auto Regressive with Exogenous Input (ARX) Model

The Auto regressive (AR) [140] is a very common model with the help of which one can shape the frequency characteristics of the model with a few linear parameters. AR is a very powerful tool for analysis of weakly damped oscillatory systems which may be hidden under a high noise level. The Auto regressive with exogenous input (ARX) [140] model is the most widely applied linear dynamic model because of its easily computable parameters. The model uses linear least square technique as the prediction error is linear in the parameters.

6.1.2 Auto Regressive Moving Average with Exogenous Input (ARMAX) Model

The Moving average (MA) [140] model has lesser significance in practical engineering applications than the AR model because it cannot be used to model oscillations with a few parameters. In addition to this, the MA model is nonlinear in its parameters when the prediction error approach is considered. In the Auto regressive moving average (ARMA) [140] model, the MA and the AR models are combined together so that the flexibility of the AR model can be enhanced. After the ARX model, the Auto regressive moving average with exogenous input (ARMAX) [140] model is the next most popular model as it possesses an extended noise model and thereby becomes more flexible.

6.1.3 Auto Regressive Integrated Moving Average with Exogenous Input (ARIMAX) Model

In many time series data set, the homogeneity property is reflected even though the series behaves as though they have no fixed mean. Apart from local level or a combination of local level and trend, one part of such series is found to be like any other part. Such homogeneous non stationary behavior of the data can be described by a model with the assumption that some suitable difference (say d^{th}) of the

TH-801_02610204

process is stationary. The autoregressive moving average with exogenous input model for which the d^{th} difference is stationary, is called the Autoregressive integrated moving average with exogenous input (ARIMAX) [102] model.

6.2 Genetic Algorithm Based Fuzzy Model

Applications requiring the optimization of a multi-dimensional function can be successfully solved by using evolutionary algorithms. These algorithms are based on the evolution of a population towards the solution of the application problem. The population of possible solutions which evolves in one generation is progressively taken to the next successive generations until it converges at a satisfactory solution of the problem. The genetic algorithm (GA) [141, 142] is one such evolutionary algorithm. This algorithm was envisaged by Holland [67] in the 1970s. It is based on the Darwinian theory of survival of the fittest, which states that the fitter and stronger individuals in a population have a higher chance of creating offsprings for the next generation. It can be implemented as an optimization search procedure which uses the principles of genetics and natural selection by modeling possible solutions to a search problem as strings of zeroes and ones. This algorithm comprises of basic genetic operators. They are selection, crossover and mutation. The flow chart for this method is presented in Fig. 6.1.

A genetic algorithm based fuzzy model is studied in this chapter. Here the genetic algorithm is fused with fuzzy logic to model a system. In this GA based model, the process of deriving the initial fuzzy model is the same as that of the ANFIS model. After the initial fuzzy model is formed, the genetic algorithm is used to update the consequent parameters of the generated fuzzy rules, so as to produce a final GA based fuzzy model of the system.

6.3 Experimental Results and Discussion

The statistical and the GA based fuzzy model for the Box and Jenkins gas furnace and the thermal power plant data are presented in the following two subsections. In each of these cases, the first half of the data set is used for training and the next half is used for validation purpose. For the Box and Jenkins gas furnace data, the input x(t) is the gas feed rate in cubic feet per minute (ft^3/min) and the output y(t) is in the form of percentage of carbon-dioxide (CO_2) concentration in the outlet gas. Similarly for the first four sets of thermal power plant data, the input $x(t)$ is the gas flow rate in cubic kilometer per hour ($\mathrm{km}^3/\mathrm{hr}$) and the output $y(t)$ is in the form of generated power in Gigawatt (GW). In the fifth set

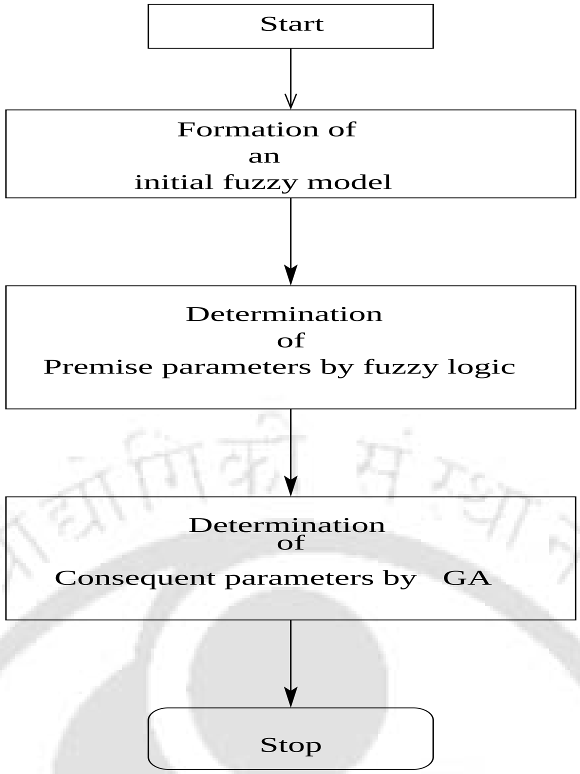

Figure 6.1: Flow Chart for the GA based Fuzzy model

of the thermal power plant data, the input x(t) is the gas feed in Million metric standard cubic metre (Mmscum) and the output y(t) is in the form of generated energy in Million unit (MU), where 1 unit = 1kWh (kilo-watt-hour).

6.3.1 Statistical models

For the model identification purpose, three different statistical models have been used. These are the ARX, ARMAX and the ARIMAX model. The modeling results obtained from these three statistical models are arranged in a tabular form in Table 6.1.

From Table 6.1 it is observed that the performance of the ARIMAX model is the best having the least RMSE as compared to the ARX and the ARMAX model. Hence, the ARIMAX model is selected to compare with the proposed model. The ARIMAX model is built for both the Box and Jenkins gas furnace and the thermal power plant data using the optimally selected inputs.

TH-801_02610204

Table 6.1: Results obtained for Statistical models using the Box and Jenkins gas furnace and the thermal power plant data

Sl. No.	Data	No of training data	Model	RMSE with testing data
1	Box and Jenkins gas furnace	145	ARX ARMAX ARIMAX	1.5796 1.5353 1.2305
2	Thermal power plant (Data set-I)	699	ARX ARMAX ARIMAX	0.0676 0.0533 0.0110
3	Thermal power plant (Data set-II)	675	ARX ARMAX ARIMAX	0.0980 0.0219 0.0163
4	Thermal power plant (Data set-III)	713	ARX ARMAX ARIMAX	0.1302 0.0839 0.0113
5	Thermal power plant (Data set-IV)	494	ARX ARMAX ARIMAX	0.0947 0.0106 0.0096
6	Thermal power plant (Data set-V)	320	ARX ARMAX ARIMAX	0.5871 0.5834 0.5451

Case 1:- Box and Jenkins gas furnace data

Here, the 145 Box and Jenkins gas furnace data pairs are used for building the ARIMAX model with $x(t-3)$ and $y(t-1)$ as the two optimal inputs. Fig. 6.2 shows the actual output and the ARIMAX model predicted output which are plotted versus sample number.

Case 2:- Thermal power plant data

(i) Data set-I

The first set of the thermal power plant data comprises of 699 data pairs. The optimal inputs used are $x(t-6)$ and $y(t-1)$. Fig. 6.3 shows the ARIMAX model's predicted output along with actual output versus sample number.

(ii) Data set-II

For this data set the ARIMAX model is built with $x(t-3)$ and $y(t-1)$ as the optimal inputs and by using 675 input output data pairs. The actual and the ARIMAX model predicted output for this data set are shown in Fig. 6.4.

(iii) Data set-III

TH-801_02610204

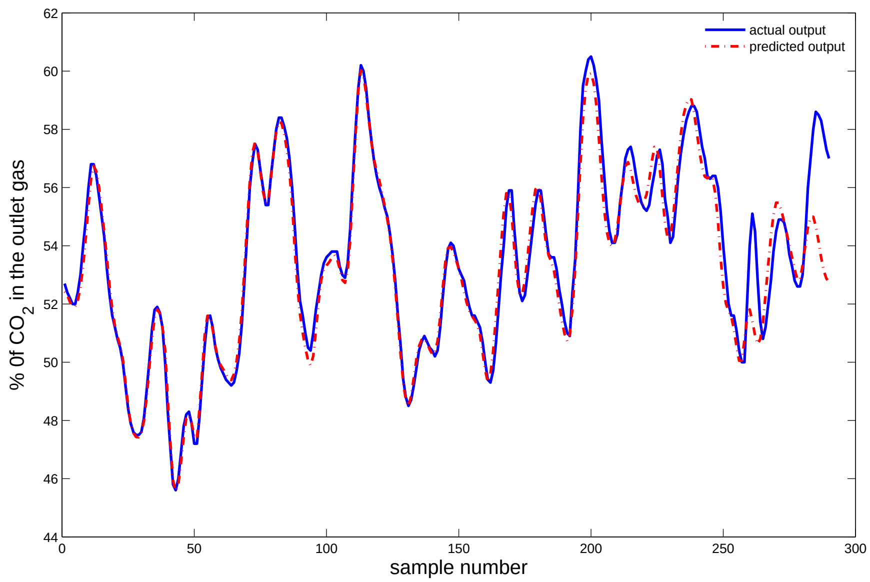

Figure 6.2: Actual and ARIMAX model predicted output with Box and Jenkins gas furnace data

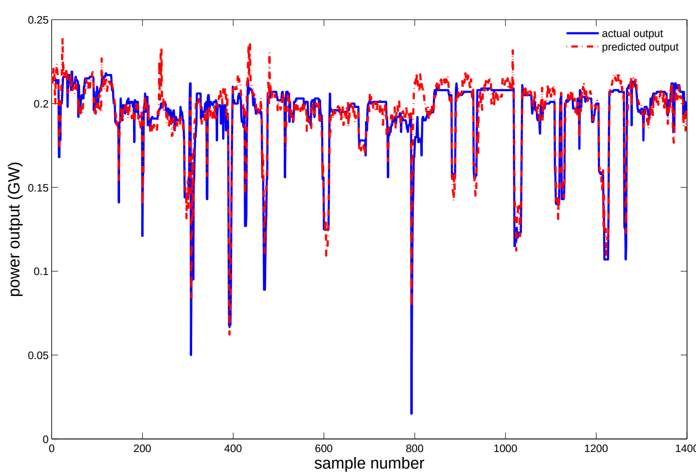

Figure 6.3: Actual and ARIMAX model predicted output with data set-I of thermal power plant

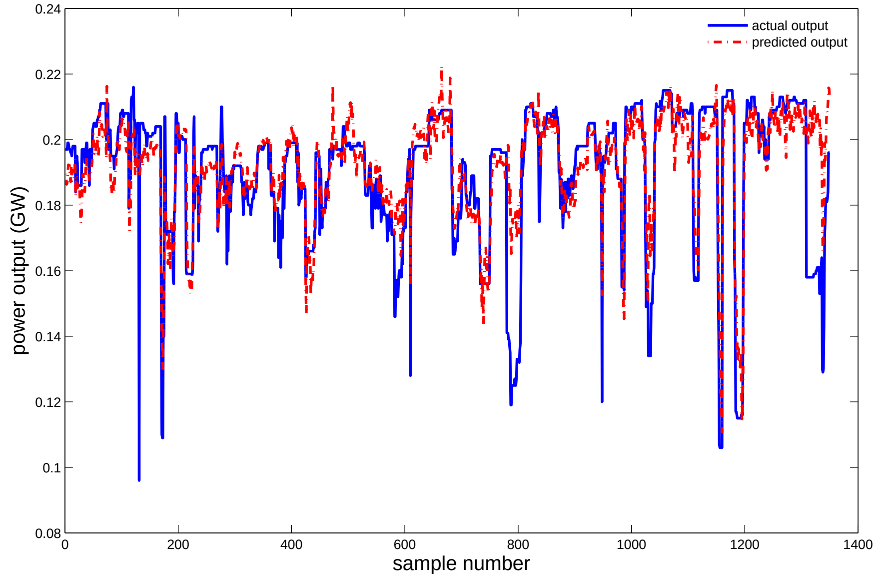

Figure 6.4: Actual and ARIMAX model predicted output with data set-II of thermal power plant

In this data set $x(t-1)$ and $y(t-1)$ are used as the optimal inputs and the ARIMAX model is built with 713 input-output data pairs. The actual and the ARIMAX model's predicted output plotted vs sample number are shown in Fig. 6.5.

(iv) Data set-IV

With $x(t-1)$ and $y(t-1)$ as the optimal inputs, the ARIMAX model is built with 494 input-output data pairs. Fig. 6.6 shows the actual and the ARIMAX model predicted output plotted vs sample number.

(v) Data set-V

The ARIMAX model for this data set is built with $x(t-1)$ and $y(t-1)$ as the optimal inputs and using 320 data pairs. The output predicted by the ARIMAX model along with the actual output plotted vs sample number is shown in Fig. 6.7.

6.3.2 GA based fuzzy model

In GA based fuzzy model, the initial fuzzy model is built in the same line as that of the ANFIS model. After the initial fuzzy model is developed based on both the subtractive clustering and the grid based partitioning, the updation of the consequent parameters to fine tune the fuzzy model is carried out by using genetic algorithm. The genetic algorithm searches the entire solution space for the best fit of the consequent parameters for the fuzzy model. The GA based fuzzy model is built for both the Box and Jenkins gas furnace and the thermal power plant data.

TH-801_02610204

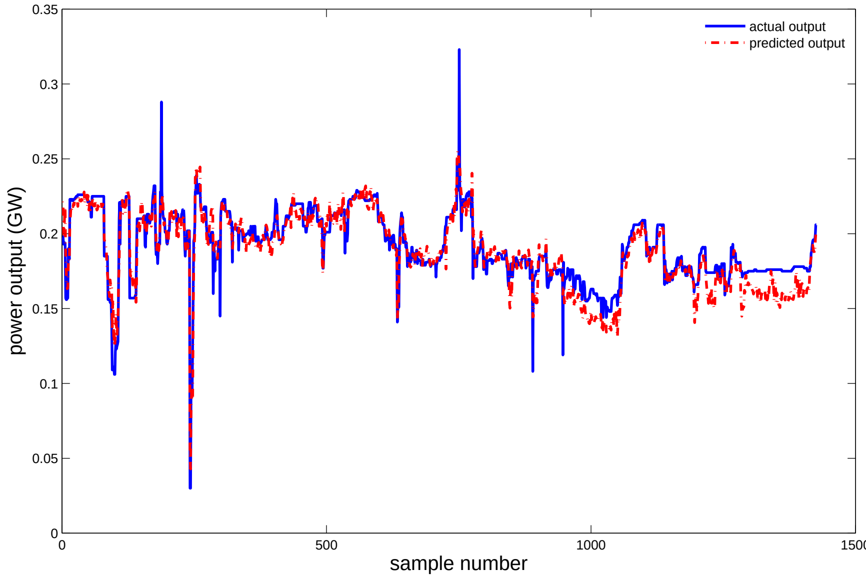

Figure 6.5: Actual and ARIMAX model predicted output with data set-III of thermal power plant

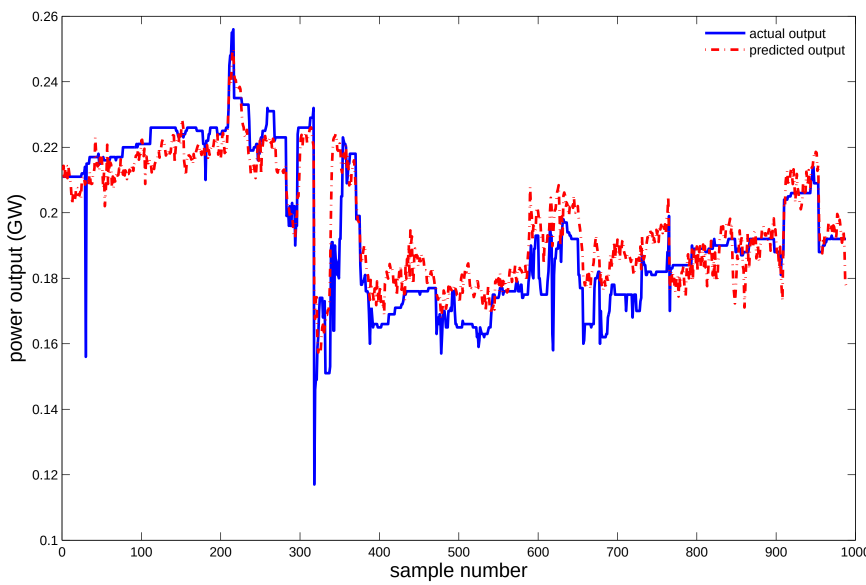

Figure 6.6: Actual and ARIMAX model predicted output with data set-IV of thermal power plant

TH-801_02610204

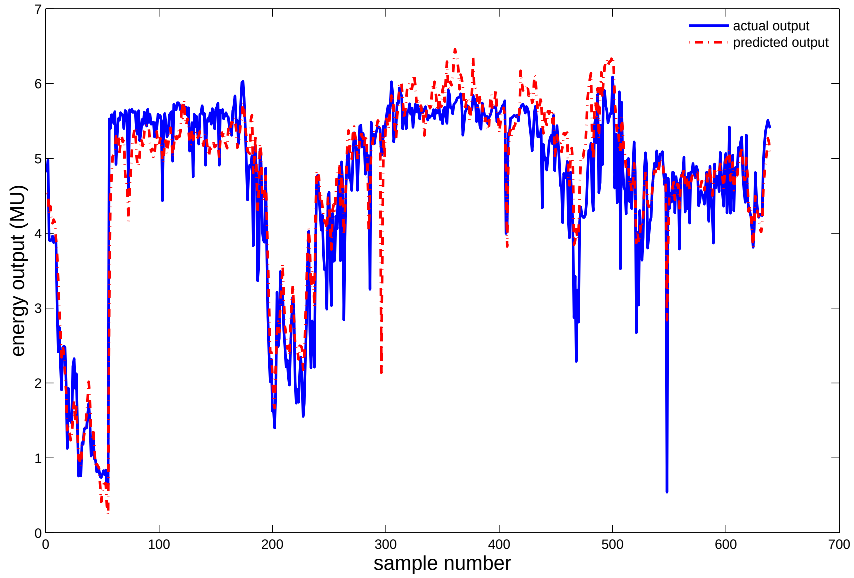

Figure 6.7: Actual and ARIMAX model predicted output with data set-V of thermal power plant

The different parameters that have been used in all GA based fuzzy models are as follows:-
The number of generations are restricted to a maximum of 1000, with a crossover fraction of 0.8, a migration fraction of 0.2, migration interval of 20, stall generation limit of 100, a stall time limit of 50 seconds, migration interval of 20 and a penalty factor of 100. The fitness function in this methodology scales the raw scores based on the rank of each individual. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is assigned a value of 1, the next most fittest is assigned a value of 2, and so on. Rank fitness scaling removes the effect of the spread of the raw scores.

Case 1:- Box and Jenkins gas furnace data

This GA based fuzzy model is built using the 145 gas furnace data pairs, where $x(t-3)$ and $y(t-1)$ are the two optimal inputs. The actual output and the output predicted by the GA based fuzzy model are shown in Fig. 6.8.

Case 2:- Thermal power plant data

(i) Data set-I

The first data set of the thermal power plant comprising of 699 data pairs is used here. The two optimal inputs are $x(t-6)$ and $y(t-1)$. The actual and the GA based fuzzy model's predicted output plotted vs sample number are shown in Fig. 6.9.

TH-801_02610204

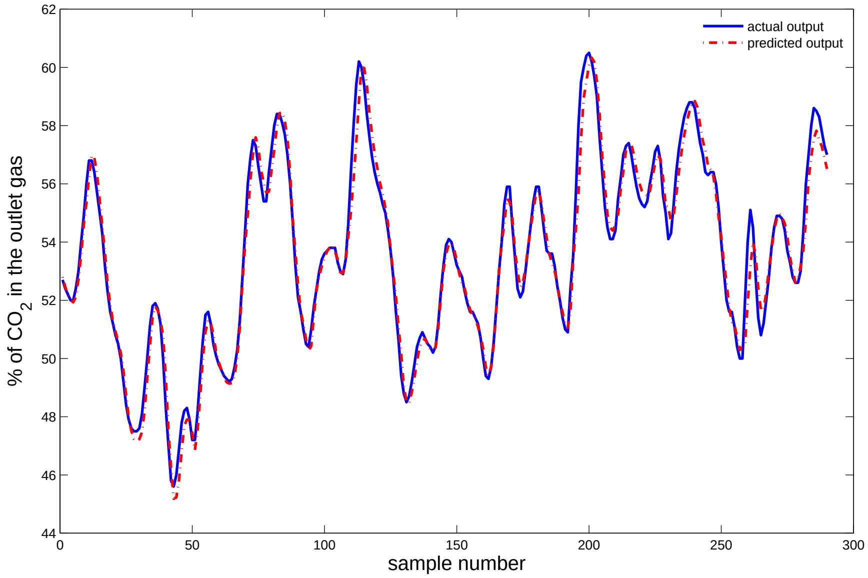

Figure 6.8: Actual and GA based fuzzy model predicted output with Box and Jenkins gas furnace data

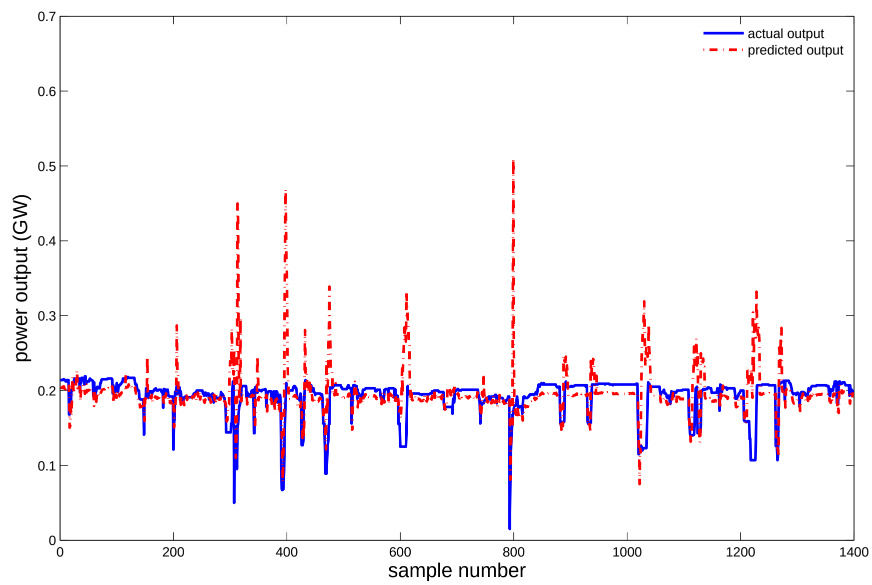

Figure 6.9: Actual and GA based fuzzy model predicted output with data set-I of thermal power plant

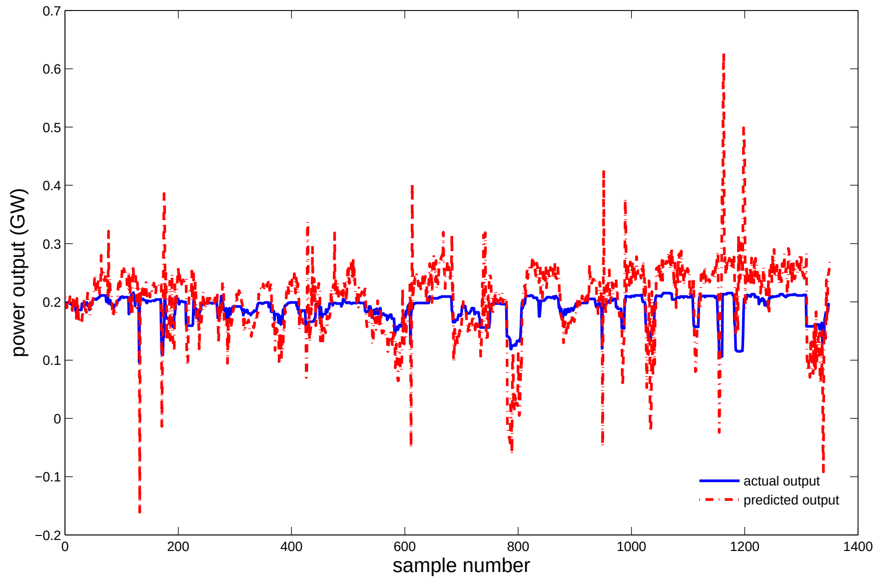

Figure 6.10: Actual and GA based fuzzy model predicted output with data set-II of thermal power plant

(ii) Data set-II

The second data set comprises of 675 input output data pairs. Here $x(t-3)$ and $y(t-1)$ are the optimal inputs. Fig. 6.10 shows the actual output and the GA based fuzzy model's predicted output plotted vs sample number.

(iii) Data set-III

In this experiment 713 input-output data pairs are used. The optimal inputs are $x(t-1)$ and $y(t-1)$. The output predicted by the GA based fuzzy model and the actual output are plotted vs sample number in Fig. 6.11.

(iv) Data set-IV

Here 494 input-output data pairs are used with $x(t-1)$ and $y(t-1)$ as the optimal inputs. The actual and the GA based fuzzy model predicted output are shown in Fig. 6.12.

(v) Data set-V

In this experiment 320 data pairs are used with $x(t-1)$ and $y(t-1)$ as the optimal inputs.The actual and the GA based fuzzy model's predicted output vs sample number is shown in Fig. 6.13.

The results obtained with the GA based fuzzy model for the Box and Jenkins gas furnace and the thermal power plant data are tabulated in Table 6.2. The table shows that for the Box and Jenkins gas furnace data set, the grid partition based method produces slightly better result whereas for the thermal

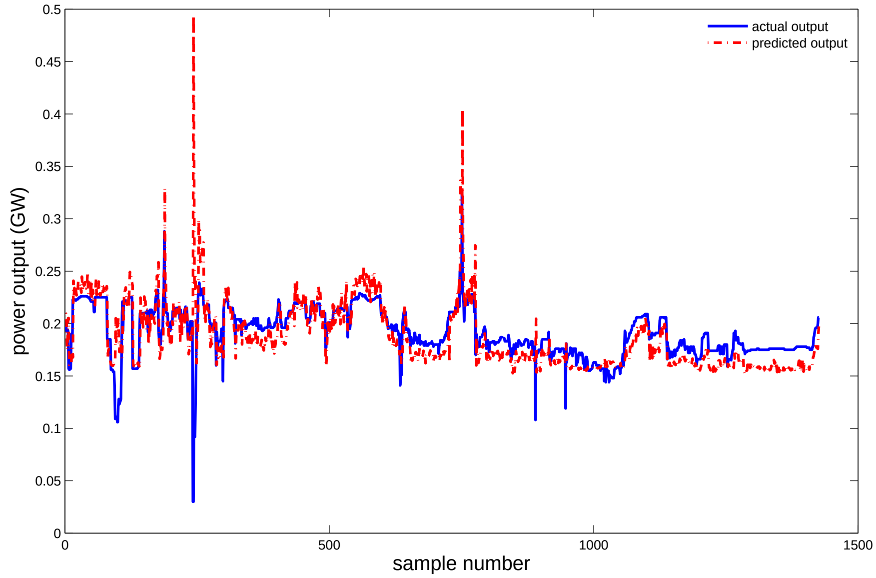

Figure 6.11: Actual and GA based fuzzy model predicted output with data set-III of thermal power plant

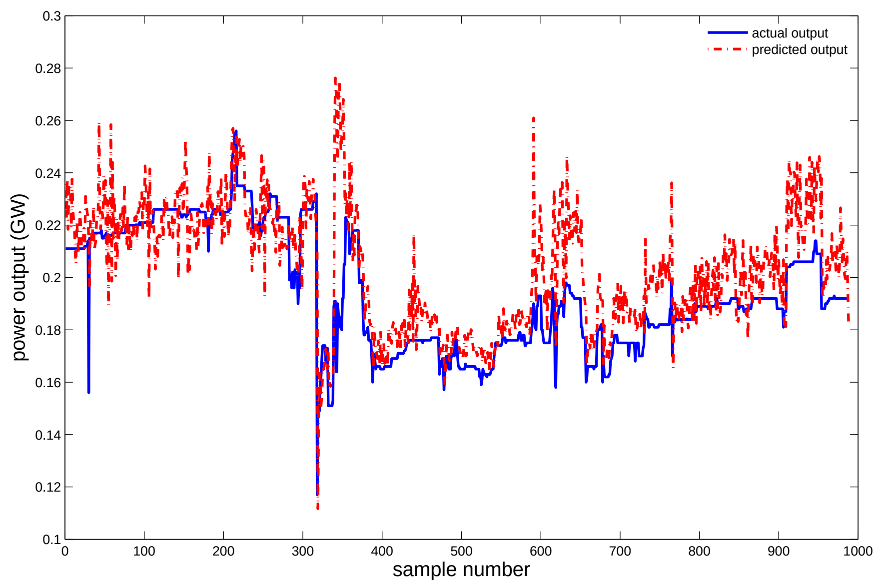

Figure 6.12: Actual and GA based fuzzy model predicted output with data set-IV of thermal power plant

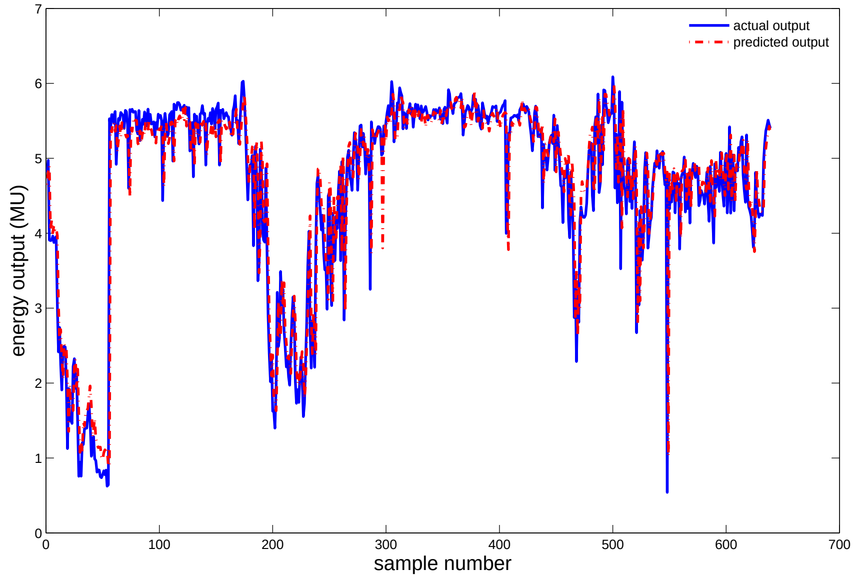

Figure 6.13: Actual and GA based fuzzy model predicted output with data set-V of thermal power plant

power plant data the subtractive clustering based model shows better performance.

The proposed FFD-V-fold based ANFIS model is now compared with the GA based fuzzy model as well as the statistical model. The performance of the proposed model is studied along with the GA based fuzzy model and the ARIMAX model. The ARIMAX model is chosen among the three different statistical models as it turns out to be the best among the three. The results obtained for these three models are now arranged in a tabular form in Table 6.3. From the table it can be observed that the proposed FFD-V-fold based ANFIS model outperforms the GA based fuzzy model as well as the ARIMAX model having the least RMSE even though the training data used by this proposed model is only about one-eighth of that used by the other two models.

6.4 Controller Using FFD-V-fold based ANFIS Model

In this section a controller based on the proposed FFD-V-fold based ANFIS model is designed. Here the ANFIS based inverse model controller is used with an open loop strategy in which the controller is the inverse of the ANFIS based plant [79]. The inverse model of the plant [79] is obtained from the input-output data that has been used to obtain the ANFIS based model for the system. This controller uses the inverse learning or general learning in two different stages.

Table 6.2: Results obtained for the GA based fuzzy model using the Box and Jenkins gas furnace and the thermal power plant data

Sl. No.	Data	No of training data	RMSE (testing) (Grid partition based)	RMSE (testing) (Subtractive clustering based)
1	Box and Jenkins gas furnace	145	0.5830	0.5998
2	Thermal power plant (Data set-I)	699	0.0430	0.0360
3	Thermal power plant (Data set-II)	675	0.0964	0.0644
4	Thermal power plant (Data set-III)	713	0.1509	0.0173
5	Thermal power plant (Data set-IV)	494	0.3879	0.0174
6	Thermal power plant (Data set-V)	320	0.5562	0.5399

Table 6.3: Comparison of the FFD-V-fold based ANFIS, GA based fuzzy and the statistical ARIMAX model obtained for the Box and Jenkins gas furnace and the thermal power plant data

Sl. No.	Data	Model	No of training data	RMSE (testing) (Grid partition based)	RMSE (testing) (Subtractive clustering based)
1	Box and Jenkins gas furnace	FFD-V-fold ARIMAX GA based fuzzy	18 145 145	0.5378 1.2305 0.5830	0.5332 1.2305 0.5998
2	Thermal power plant (Data set-I)	FFD-V-fold ARIMAX GA based fuzzy	81 699 699	0.0109 0.0110 0.0430	0.0108 0.0110 0.0360
3	Thermal power plant (Data set-II)	FFD-V-fold ARIMA GA based fuzzy	72 675 675	0.0086 0.0163 0.0964	0.0089 0.0163 0.0644
4	Thermal power plant (Data set-III)	FFD-V-fold ARIMAX GA based fuzzy	72 713 713	0.0076 0.0113 0.1509	0.0075 0.0113 0.0173
5	Thermal power plant (Data set-IV)	FFD-V-fold ARIMAX GA based fuzzy	72 494 494	0.0033 0.0096 0.3879	0.0034 0.0096 0.0174
6	Thermal power plant (Data set-V)	FFD-V-fold ARIMAX GA based fuzzy	36 320 320	0.5174 0.5451 0.5562	0.5118 0.5451 0.5399

TH-801_02610204

- In the learning phase the inverse model of the ANFIS based plant is obtained with the available input-output data. This is called the training phase of the inverse controller. Fig. 6.14 illustrates this training phase.

- In the next phase the inverse model of the ANFIS based plant is used to initiate the control action. This is called the application phase of the ANFIS based controller and it is illustrated in Fig. 6.15.

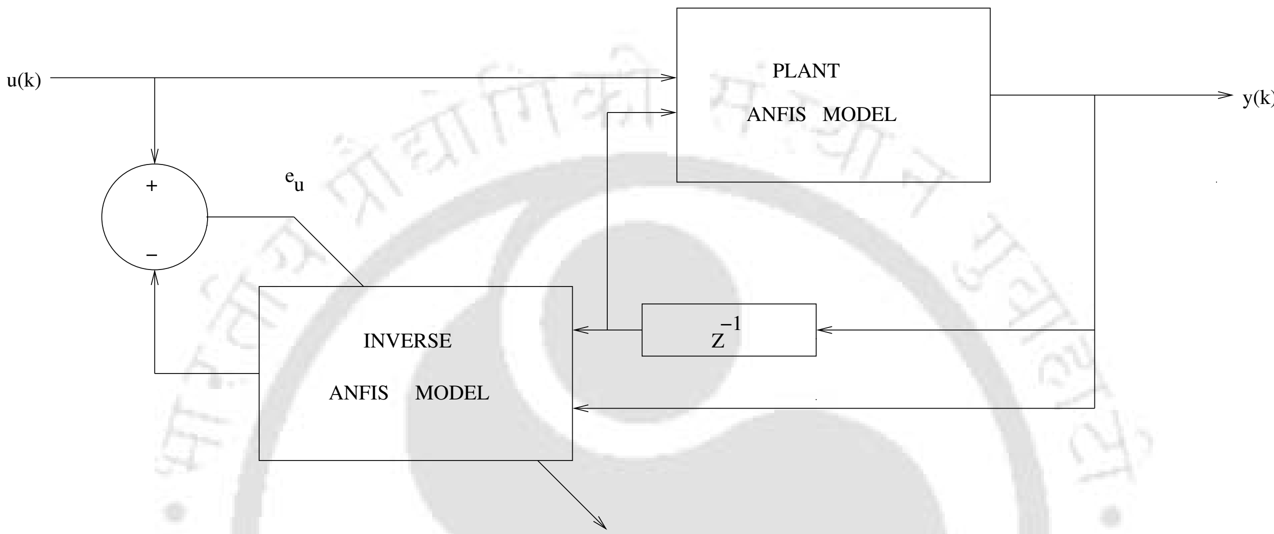

Figure 6.14: Training phase of the inverse controller

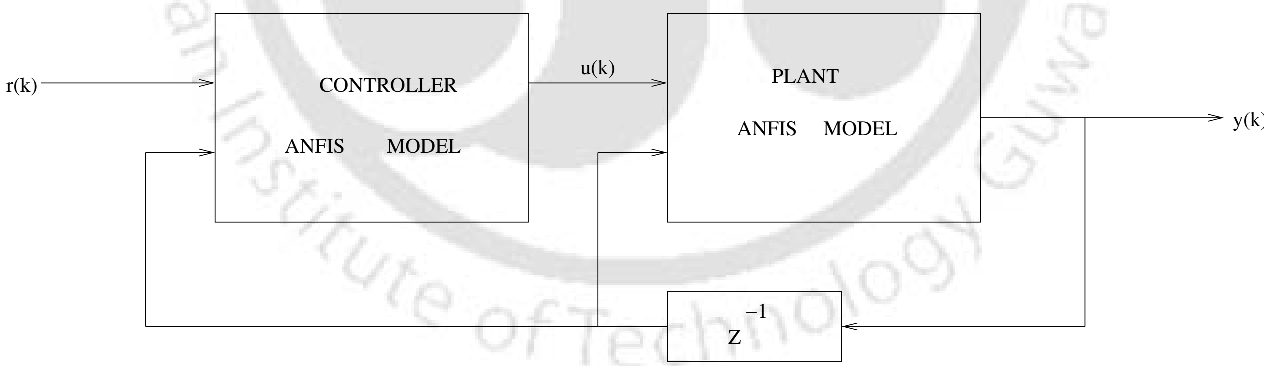

Figure 6.15: Application phase of the inverse controller

6.4.1 Controller Design

The controller designed using the general or the off-line training method does not take into consideration the aspect of minimizing the output error to ensure reference tracking. This results in a large

TH-801_02610204

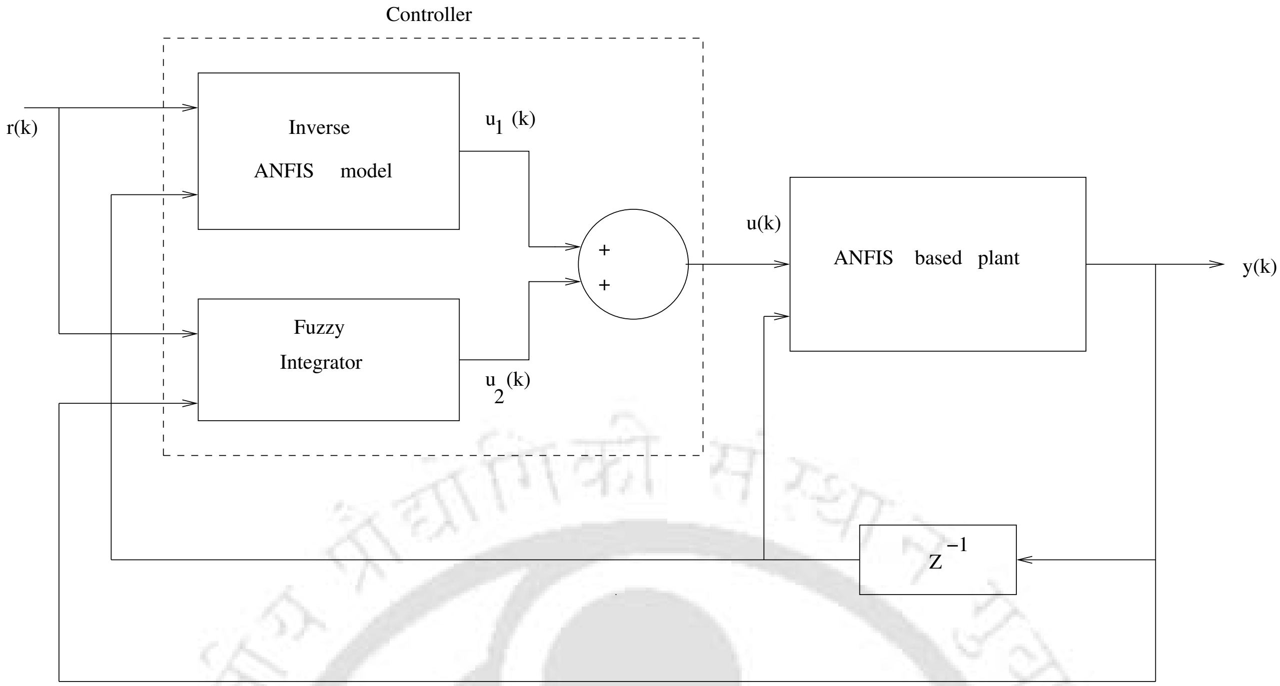

Figure 6.16: Inverse controller with fuzzy integrator block

steady state error. To reduce the sustained steady state error of the plant a fuzzy integrator block is incorporated along with the ANFIS based controller to form the total controller block. The integral action increases the control signal if there is a small positive error. Similarly for a negative error the integral action decreases. The prediction error is used as an input to the fuzzy integrator which is illustrated in Fig. 6.16. This is done to generate the integrating constant K_i so that the final steady state error can be brought nearly to zero. The fuzzy rules for the integrator block are framed heuristically to tune the value of K_i, as there is no systematic method for framing the fuzzy rules for building the controller. In the present work these rules have have been framed based on the results of the simulations that were conducted with different values of the controller inputs.

The rules for the integrator block are of the following form:

Control rule b:

IF $e_1(k)$ is M_1^b and $e_2(k)$ is M_2^b and $\ldots e_N(k)$ is M_N^b THEN

TH-801_02610204

$u_2(k) = K_i \sum_{j=1}^{k} e_j T_s$,

$b = 1, 2, ...q$ and $n = 1, 2, ..., N$

where

M_n^b is the fuzzy set $(n = 1, 2, ...N)$, T_s is the sampling time and error $e_j = r(k) - y(k)$, $r(k)$ being the reference input and $y(k)$ is the system's output.

For the above rule, the fuzzy integral controller system may be inferred as follows:

Given a combination of (K_i, e_j, T_s), the final output of fuzzy integrator system may given as follows:

$$u_2(k) = \sum_{b=1}^{q} \mu_b(e(k)) K_i \sum_{j=1}^{k} e_j T_s, \tag{6.4.1}$$

$$b = 1, 2, ..., q \tag{6.4.2}$$

where

$$\mu_b(e(k)) = w_b(e(k)) / \sum_{b=1}^{q} w_b(e(k))$$

and

$$w_b(e(k)) = \prod_{n=1}^{N} M_n^b(e(k)).$$

$M_n^b(e(k))$ is the degree of the membership of $e(k)$ in M_n^b. Here $w_b(k) \geq 0$, for $b = 1, 2, ..., q$ and $\sum_{b=1}^{q} w_b(k) > 0$ for all k. Therefore, $\mu_b(e(k)) \geq 0$ for $b = 1, 2, ..., q$ and $\sum_{b=1}^{q} \mu_b(e(k)) = 1$. The output values of the available input-output data set range from $0.0150 - 0.219$. So the fuzzy rules for the ANFIS based controller is framed to suit this range. The reference input which is a step signal is also chosen to suit the available data set.

6.4.2 Experimental Results and Discussion

The performance of the controller is illustrated in Figs. 6.17 - 6.24. In Fig. 6.17 the system output is shown against the reference step input and the overshoot here is found to be 1.6%. Fig. 6.18 shows the prediction error of the controller. This error is the difference between the desired output and the actual output of the plant. In Fig. 6.19 the control signal i.e the output of the controller is shown. In

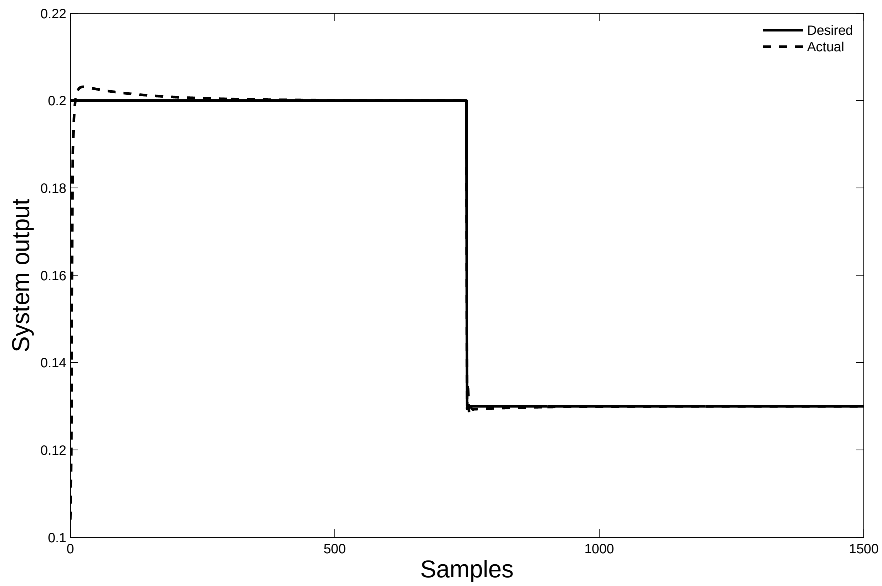

Figure 6.17: Desired output and the actual output of the plant with the controller

order to check the performance of the controller in presence of disturbance, two separate noises (step inputs) were injected at around the 400^{th} and 1000^{th} sample. The result in Fig. 6.20 demonstrates the fast recovery of the controlled system to its final steady state after being disturbed with noise.

In order to check the performance of the controller under variable plant dynamics, a random variation of the plant parameters by 10% is allowed. The results under the changing plant dynamics are illustrated in Figs. 6.21 - 6.24. The controller's performance under variable plant dynamics along with disturbance is plotted in Fig. 6.24.

6.5 Conclusion

It is observed that the FFD-Vfold based ANFIS controller can track the reference step input quite faithfully even under plant parameter variation alongwith disturbance.

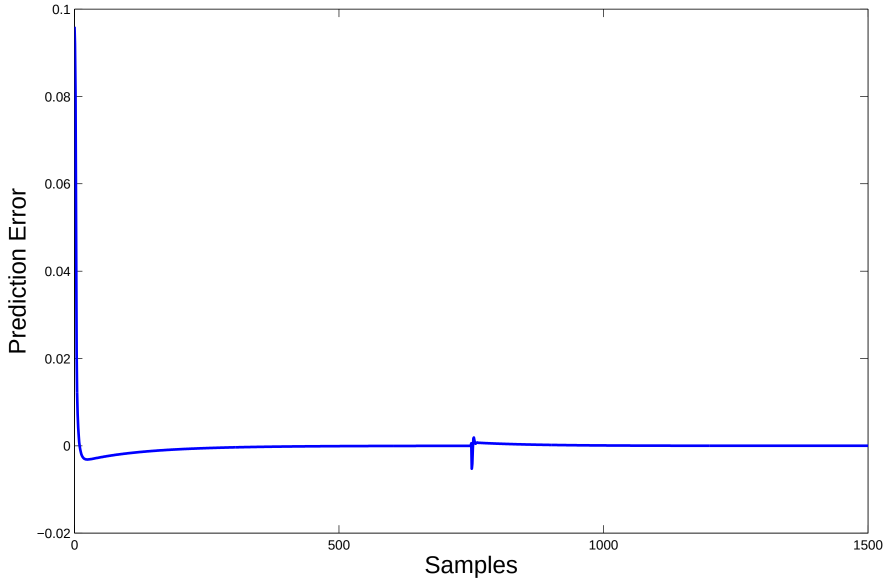

Figure 6.18: Prediction error of the controller

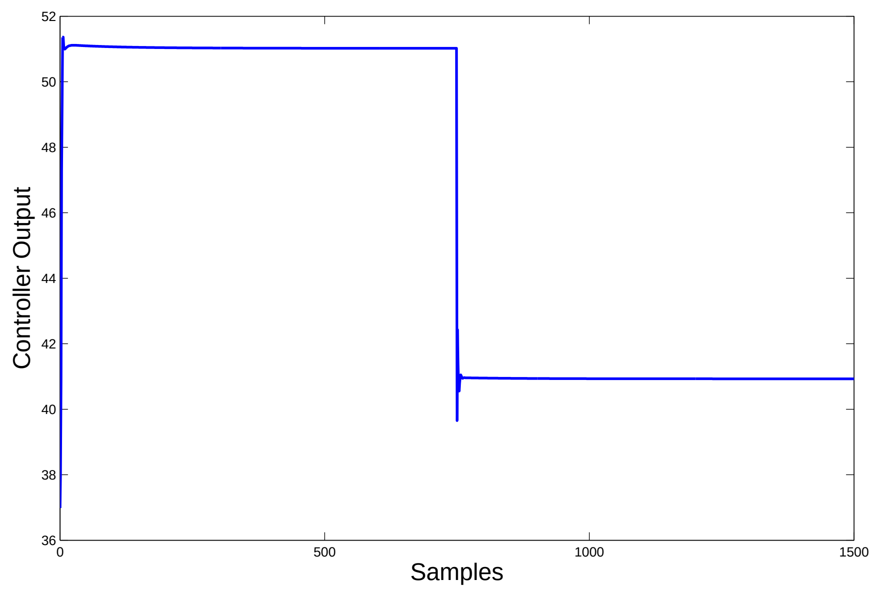

Figure 6.19: Output of the controller

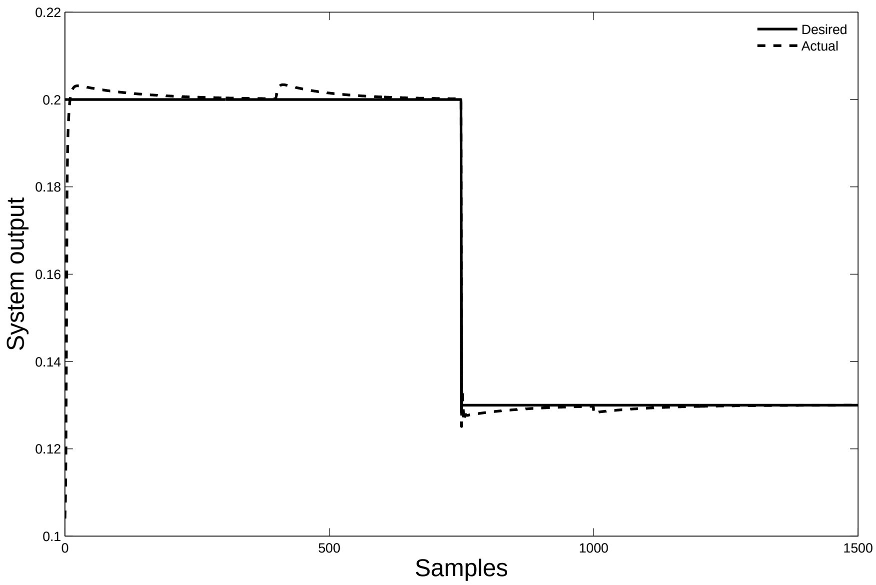

Figure 6.20: Desired and actual output of the plant with controller in presence of disturbance

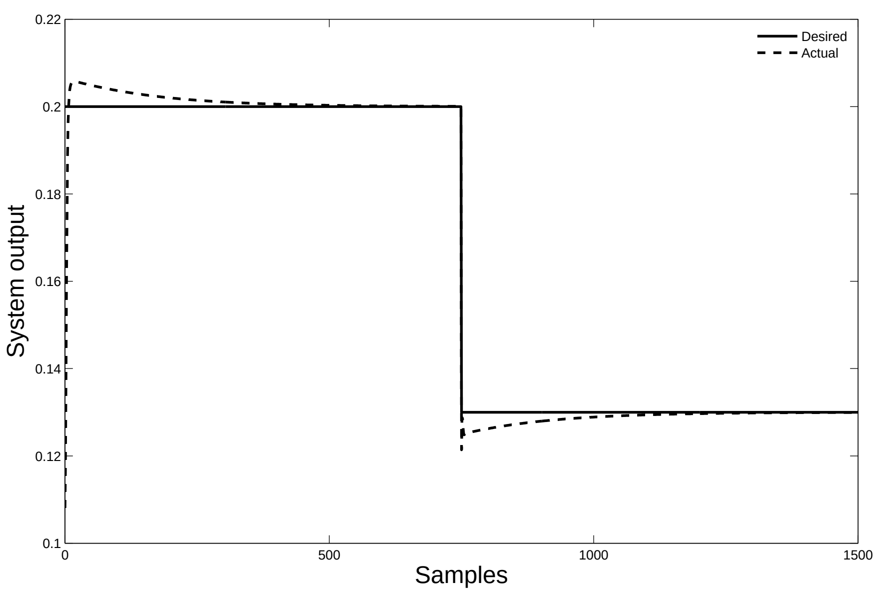

Figure 6.21: Desired and actual output of the plant with controller having plant parameter variations

TH-801_02610204

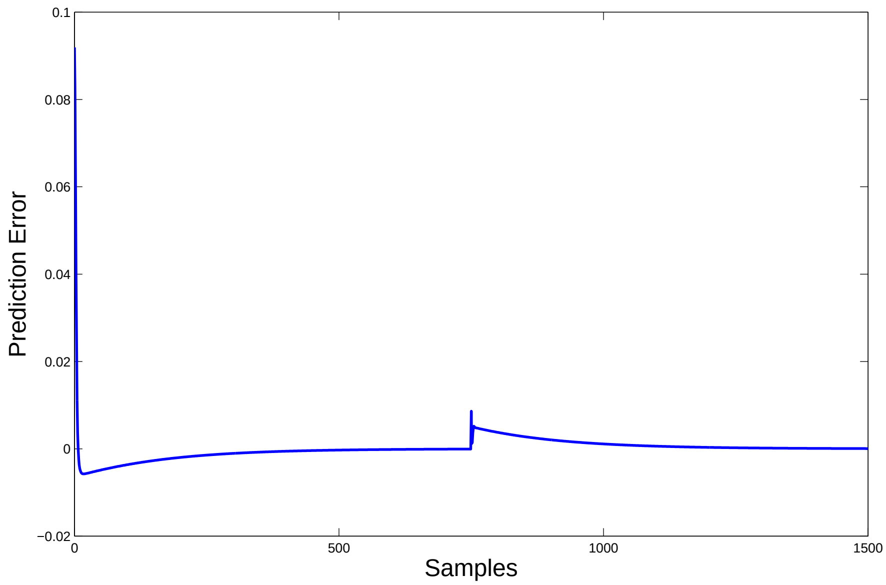

Figure 6.22: Prediction error of the controller with plant parameter variations

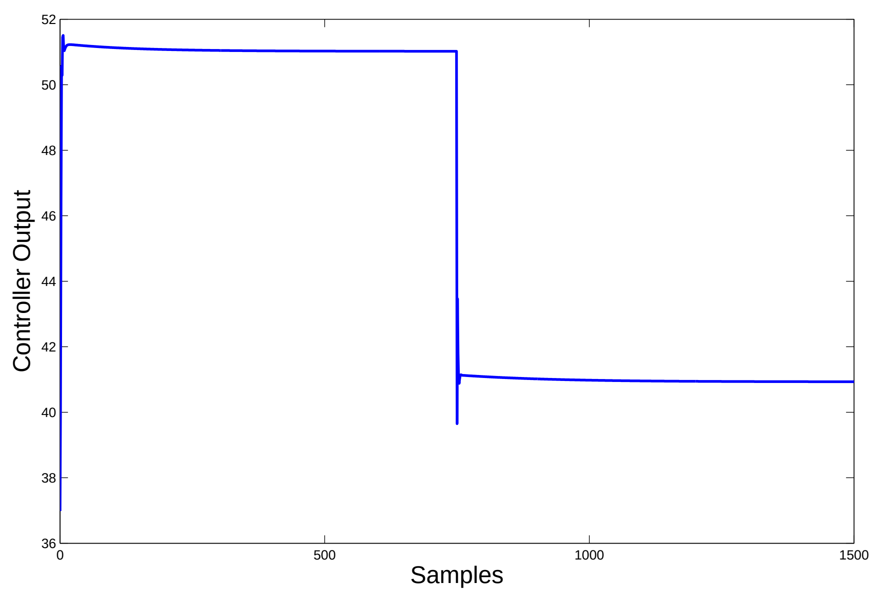

Figure 6.23: Output of the controller with plant parameter variations

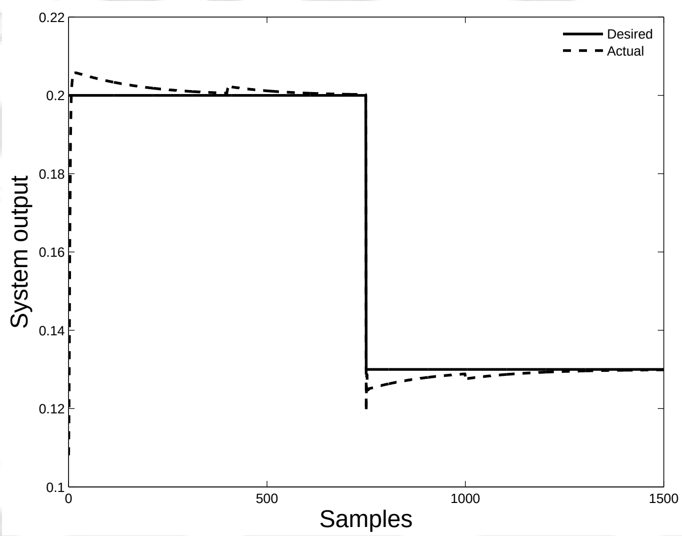

Figure 6.24: Desired and actual output of the controller based plant with plant parameter variations along with disturbance

Chapter 7

Conclusions and Future Work

In this concluding chapter the salient contributions of the thesis are summarized. Also a few aspects which may be explored for further research are outlined here.

7.1 Summary of the thesis

The major contribution in this thesis is towards newer development in the ANFIS methodology. The main emphasis is to build an ANFIS model based on an optimally selected data set. The advantage of the proposed method has been demonstrated in terms of improved modeling results as compared to the conventional ANFIS model as well as conventional statistical models. The proposed method is used to design a controller and study its performance.

In Chapter 1 the aim of the present research work is discussed. Chapter 2 presents a brief overview of different system identification techniques based on soft computing method. Here some important forecasting techniques are also discussed briefly. Chapter 3 outlines the research objectives and methodology followed in the present work. In Chapter 4 the ANFIS model which forms the backbone of the proposed method for system identification is explained in details. Chapter 5 describes the two different techniques namely the Full factorial design and the V-fold method which are used to augment the ANFIS model for optimal selection of the training data. In Chapter 6 the results obtained by using the proposed model are compared with the GA based fuzzy model and statistical models. Chapter 7 concludes the thesis with a discussion on the important findings of the research work.

The major contributions of the thesis are summarized below:

- The statistical Full factorial design technique is used to choose an optimal data set for training the neural network in the conventional ANFIS model. The available data is selected optimally

TH-801_02610204

for training the subtractive clustering and grid partition technique based ANFIS models.

- The V-fold technique which is used for cross validation purpose is applied for selecting the optimal data for training. By using this technique, it has been possible to build an ANFIS model using a training data set which is around one-eighth of the size of the data set used in the conventional ANFIS model.

- The full factorial design technique and the V-fold method are combined for optimal data selection in training the ANFIS model. The obtained results show that performance of the combined FFD-V-fold based ANFIS model is better than the individual FFD based and V-fold based ANFIS models.

- A controller is designed using the proposed FFD-V-fold based ANFIS model which performs faithfully even with plant parameter variations and disturbance.

7.2 Discussion

The prime advantage of the soft computing tools which have been used in the present work is that the requirement of a mathematical model is not a prerequisite. The model of the system under consideration can be built around the available input-output data of the system. The inspiring features of the neural network in its ability to learn and adapt and the capability of a fuzzy system to take into account the imprecision and prevailing uncertainty are vouched to be the prime tools in the present research work. To exploit the advantages that are associated with the respective methods, the Adaptive network based fuzzy inference system (ANFIS) model comprising of both the neural network and the fuzzy logic has been used in the present work. ANFIS model requires a sufficient pool of input-output data for training. As the absence of sufficient number of input-output data in many a real world systems poses a major hurdle in modeling, an endeavor is made to use the available data, to its best possible ability such that a small critical data set can be selected optimally and used for training the ANFIS model. For this purpose, two novel techniques namely Full factorial design and V-fold cross validation technique are proposed for optimal data selection to train the ANFIS.

The experimental results obtained by applying the proposed method are encouraging enough. The proposed model shows reliable prediction performance so far as the actual values of the available data are concerned. The performance of the proposed model was also found to be at par with conventional statistical models used in forecasting. The comparable results were achieved even though a very small

TH-801_02610204

critically selected training data set was used in training the ANFIS. The strength of the input output data used is only around one-eighth of the total number of the available data set.
The results obtained from the proposed model can be very encouraging to industries associated with the real time systems, like a thermal power plant that has been considered in the present work. These results are expected to give a boost to the economic front as precious fossil fuel whose reserves are depleting day by day can be saved as the power plant need not be run for long hours to generate the modeling data for future load prediction. Besides this the time required for starting the thermal power plant before it can be fully loaded can be saved as it takes a pretty long time to start real world systems like the thermal power plant. Therefore in terms of human resources management also the results can prove to be a boon as a great many number of man hours can be saved, as the plant does not have to be run for long periods of time just to generate the modeling data.

So far as the implementation of the present findings are concerned, it is not expected to be too difficult a proposition, as the soft computing model can be connected in parallel to the hard computing and manual control system that is already there in place. Even in the remote possible case of the proposed model failing, the hard computing based tools or the manual controller will always be there to act as a backup protection system.

The results with the proposed model are, however bounded with certain limitations. This is because the modeling results are fairly accurate with respect to the choice of the critical training data set. In the FFD based ANFIS model, the levels selected in the available training data set are crucial. The proper selection of the FFD levels are important as the ANFIS model is decided based upon the optimal choice of these levels. Rigorous experiments need to be performed to determine the best FFD levels to fit into the optimal ANFIS model. Hence it requires time and the model cannot be predetermined. Another limitation of the model is the possible presence of outliers and noise in the data set that is used for the system modeling. The cross checking for a possible model which is a hybrid between an evolutionary methodology and the proposed model also could have produced a more better result.

7.3 Conclusion

System identification is the primary step in modeling of a system as its model should correctly represent the system for further analysis and control. A practical important utility of modeling is prediction

TH-801_02610204

of future values. This aspect of forecasting is even more significant in real world systems like thermal power plants. The research work carried out in this thesis studied modeling of a real world system like gas furnace and thermal power plant based on actual input output data for which a soft computing methodology ANFIS is used. This methodology has been considered to make the best use of the inherent salient features of the neural network of self adaptability and that of the fuzzy logic in its ability to take into account the imprecision and uncertainty of systems which are considered for modeling. It has been a traditionally accepted fact that for successful training of ANFIS, the number of data used should be sufficiently large. However, it may be difficult to get a large number of data in those systems like thermal power plant where generation of input output data is quite expensive as precious fuel is involved. Hence, this research work attempted to build a model for such a system using the existing small pool of input output data. For this purpose, the emphasis was on optimal selection and use of the available resource of data. It has been observed from experimental results that it is also possible to build a successful model for those types of systems where the available data set is scanty. This thesis proposed two novel techniques for optimal selection of data and applied the same in the modeling methodology. Performance of the model built using these two proposed techniques was tested for prediction and the results were compared with traditional statistical models commonly used in literature. The results obtained are promising and encouraging enough deserving further investigation by the concerned industry. The experimental results obtained verify the competence of the proposed modeling technique. Hence conclusions can be drawn that the proposed ANFIS based system identification technique augmented by optimal data selection strategy for training is comparable with other existing conventional techniques of modeling.

There is a scope for further exploring the possibility of bettering the present modeling results by taking into account the possible presence of outliers in the model's training data. The scope of strengthening the proposed ANFIS model by creating its hybrid with another evolutionary technique is another direction which might be worth looking into.

7.4 Future Work

Following are the potential directions which can be explored for future research:

- Developing a proper methodology for framing and determining the number of fuzzy rules for the proposed ANFIS model

TH-801_02610204

- Preprocessing the data before modelling is carried out so as to remove the outliers
- Exploring the possibility of developing the ANFIS model strengthened by other evolutionary algorithms
- Developing a formal methodology to decide an optimal data size for the V-folds so that modeling can be carried out with the least possible minimum number of data

TH-801_02610204

Appendix A

Sample Power Plant Data

One sample data set that was collected from the thermal power plant under North East Electric Power Corporation (NEEPCO) Limited, located in Kathalguri, Assam, India has been presented here. This set of the input output data consists of an input to the thermal power plant in the form of the gas feed to the plant in Million metric standard cubic metre (Mmscum)($10^6 m^3$). Similarly the output from the thermal power plant is in the form of generated energy in Million unit (MU), where 1 unit = 1 kWh (kilo-watt-hour). The thermal power plant data set comprising of the inputs and the outputs are collected from the hard copy of the records maintained by the power plant authorities. This data set has been shown in tabular form in Table A.1 which spreads over the next six successive pages.

TH-801_02610204

Table A.1: Thermal power plant data

Sample No	Input (Mmscum)	Output (MU)	Sample No	Input (Mmscum)	Output (MU)	Sample No	Input (Mmscum)	Output (MU)
1	1.324	4.9452	41	0.574	1.3966	81	1.32	5.6243
2	1.322	4.6551	42	0.618	1.3997	82	1.4534	5.6538
3	1.313	4.8245	43	0.673	1.5822	83	1.4387	5.6143
4	1.191	4.5939	44	0.73	1.7254	84	1.4749	5.6229
5	1.188	4.2019	45	0.598	1.4773	85	1.4641	5.6235
6	1.241	4.5838	46	0.455	1.023	86	1.3119	5.3734
7	1.28	4.8133	47	0.497	1.1173	87	1.3136	5.325
8	1.25	4.973	48	0.529	1.2861	88	1.319	5.6387
9	1.153	3.9096	49	0.447	0.986	89	1.3993	5.2355
10	1.232	3.9021	50	0.374	0.9672	90	1.4414	5.3554
11	1.129	3.9072	51	0.437	0.8112	91	1.4045	5.5192
12	1.159	3.9627	52	0.412	0.8423	92	1.3778	5.4554
13	1.169	3.8762	53	0.421	0.838	93	1.4796	5.6182
14	1.157	3.9246	54	0.325	0.7497	94	1.4598	5.5868
15	1.083	3.9547	55	0.283	0.7372	95	1.4679	5.4311
16	1.025	3.0296	56	0.372	0.7779	96	1.4582	5.6565
17	0.922	2.4153	57	0.379	0.8284	97	1.4408	5.4803
18	0.819	2.7417	58	0.382	0.7776	98	1.3682	5.5866
19	0.77	2.2473	59	0.378	0.8381	99	1.3574	5.4574
20	0.678	1.9082	60	0.257	0.6241	100	1.3831	5.4638
21	0.804	2.4771	61	0.262	0.6413	101	1.3372	5.4873
22	0.724	2.4917	62	1.402	5.5288	102	1.3671	5.2863
23	0.674	2.4805	63	1.4491	5.3996	103	1.4481	5.4758
24	0.598	2.0008	64	1.4677	5.5825	104	1.4554	5.3959
25	0.439	1.1264	65	1.4577	5.582	105	1.3766	5.5891
26	0.556	1.9277	66	1.4161	5.468	106	1.3897	5.5449
27	0.485	1.4958	67	1.3967	5.6255	107	1.433	5.5731
28	0.493	1.4993	68	1.4723	4.9201	108	1.3962	5.5036
29	0.515	1.461	69	1.4039	5.2882	109	1.2972	4.436
30	0.627	2.2222	70	1.4499	5.5317	110	1.3581	5.466
31	0.66	2.3257	71	1.3176	5.5103	111	1.4581	5.6001
32	0.521	1.9292	72	1.4471	5.6107	112	1.3911	5.4636
33	0.663	2.1237	73	1.4267	5.591	113	1.3568	5.2911
34	0.461	1.4513	74	1.4003	5.5735	114	1.4164	5.3258
35	0.401	0.7558	75	1.4198	5.3997	115	1.4649	5.5799
36	0.422	0.9432	76	1.2299	5.6397	116	1.4609	5.6018
37	0.404	0.7583	77	1.2666	5.6239	117	1.391	5.6248
38	0.54	1.2023	78	1.2331	5.4488	118	1.3226	4.9621
39	0.494	1.181	79	1.0844	4.5987	119	1.4091	5.7229
40	0.559	1.3888	80	1.3462	5.65	120	1.4565	5.6937

TH-801_02610204

Sample No	Input Mmscum	Output MU	Sample No	Input Mmscum	Output MU	Sample No	Input Mmscum	Output MU
121	1.4046	5.6481	161	1.4514	5.6148	201	1.0818	3.7636
122	1.4227	5.7434	162	1.4216	5.6523	202	0.9669	2.885
123	1.4581	5.7434	163	1.4487	5.5285	203	0.794	2.3875
124	1.4078	5.7161	164	1.3801	5.5787	204	0.638	2.0173
125	1.4615	5.643	165	1.3978	5.6097	205	0.7455	2.3181
126	1.5922	5.6752	166	1.371	5.5655	206	0.6064	1.624
127	1.4414	5.6831	167	1.4916	5.5533	207	0.5539	1.6297
128	1.5922	5.7081	168	1.4093	5.4241	208	0.5714	1.3992
129	1.4079	5.5961	169	1.4812	5.3844	209	0.8361	1.9668
130	1.4641	5.6839	170	1.4714	5.3092	210	0.9027	3.2116
131	1.4744	5.5765	171	1.4681	5.6352	211	0.9078	2.4865
132	1.4162	5.6774	172	1.4598	5.6986	212	0.8212	2.5966
133	1.3626	5.0974	173	1.491	5.835	213	0.9907	3.4888
134	1.3944	5.1586	174	1.4121	5.47	214	1.0112	3.0417
135	1.4749	5.4782	175	1.3523	5.5049	215	1.1095	3.2544
136	1.3827	4.7539	176	1.4041	5.1959	216	0.8682	2.7095
137	1.4534	5.5895	177	1.4264	5.4507	217	0.801	2.4917
138	1.4126	5.4194	178	1.5057	5.7048	218	0.787	2.2152
139	1.4518	5.5629	179	1.4833	6.0147	219	0.7407	2.3094
140	1.4009	5.3345	180	1.576	6.0291	220	0.7879	2.0827
141	1.4354	5.1913	181	1.4829	5.8037	221	0.7294	1.9713
142	1.4416	5.5908	182	1.4271	5.4411	222	0.9181	2.4715
143	1.4146	5.6031	183	1.3938	4.941	223	0.9833	2.9014
144	1.4078	5.7035	184	1.3375	4.4442	224	1.027	3.15
145	1.3782	5.5866	185	1.3653	4.8132	225	0.8044	2.922
146	1.4371	5.5992	186	1.3583	4.9125	226	0.7579	2.0179
147	1.4322	4.9094	187	1.5534	4.8933	227	0.7683	1.7303
148	1.4076	5.4534	188	1.4916	5.0371	228	0.7597	1.9127
149	1.3406	5.5334	189	1.1866	3.8338	229	0.7235	1.7416
150	1.3542	5.4973	190	1.3944	5.0443	230	0.8203	2.1081
151	1.4172	5.6085	191	1.4521	5.1634	231	0.7821	2.3469
152	1.3743	5.4574	192	1.2189	4.7144	232	0.7907	2.103
153	1.4078	5.6656	193	1.1288	3.3662	233	0.6632	1.553
154	1.4045	5.603	194	1.2875	3.6034	234	0.7855	1.7264
155	1.3844	5.3959	195	1.4923	5.1222	235	0.925	2.2885
156	1.3961	5.6544	196	1.3559	4.9382	236	1.0508	2.808
157	1.4146	5.685	197	1.1572	4.0445	237	1.2034	3.5663
158	1.4075	5.7383	198	1.2138	3.9089	238	1.2067	4.0265
159	1.3173	4.9094	199	1.2549	3.8814	239	1.0577	2.6875
160	1.4598	5.641	200	1.3406	4.8714	240	0.8854	2.2007

Sample No	Input (Mmscum)	Output (MU)	Sample No	Input (Mmscum)	Output (MU)	Sample No	Input (Mmscum)	Output (MU)
241	1.0559	2.8302	281	1.3064	4.5745	321	1.5001	5.3927
242	0.9588	2.6181	282	1.3609	5.0043	322	1.4315	5.6101
243	0.8492	2.2116	283	1.4632	5.4324	323	1.5087	5.5803
244	1.4039	3.7935	284	1.4497	5.4443	324	1.5447	5.3973
245	1.3961	4.664	285	1.4179	5.1507	325	1.6115	5.6272
246	1.335	4.7778	286	1.4229	5.3323	326	1.5841	5.6337
247	1.3264	4.5189	287	1.3609	5.1421	327	1.5978	5.7309
248	1.1766	4.2765	288	1.4607	4.7898	328	1.5532	5.59
249	1.2671	4.0429	289	1.3852	5.1154	329	1.5687	5.6709
250	1.1706	4.2399	290	1.2498	4.9163	330	1.5978	5.4148
251	1.1662	3.7326	291	0.9913	5.0368	331	1.6269	5.5838
252	1.1438	3.5132	292	1.324	3.2511	332	1.5532	5.7224
253	1.1524	3.5352	293	1.4281	5.4863	333	1.492	5.625
254	1.1126	2.985	294	1.4347	5.2931	334	1.5346	5.7142
255	1.3166	4.4993	295	1.4495	5.3268	335	1.5322	5.6245
256	1.2018	4.547	296	1.4675	5.1976	336	1.4889	5.5269
257	1.1092	3.513	297	1.4658	5.3189	337	1.519	5.6984
258	1.0252	3.0348	298	1.4624	5.3774	338	1.4577	5.609
259	1.2858	4.2664	299	1.4864	5.4225	339	1.5104	5.6215
260	1.2138	3.6258	300	1.4607	5.4206	340	1.3603	5.6538
261	1.2084	3.6389	301	1.4281	5.3925	341	1.4884	5.6918
262	1.2926	4.229	302	0.1338	5.0432	342	1.5427	5.6148
263	1.371	4.0629	303	1.4401	5.0623	343	1.5233	5.512
264	1.3475	4.6036	304	1.5134	5.433	344	1.5389	5.5724
265	1.3458	4.8307	305	1.3869	5.3003	345	1.4951	5.5488
266	1.13	3.6346	306	1.5961	5.4927	346	1.5261	5.5808
267	1.3928	4.9537	307	1.4665	5.6163	347	1.6153	5.5724
268	1.1761	4.6311	308	1.4534	5.7034	348	1.5516	5.6513
269	1.0525	2.8417	309	1.4845	5.5608	349	1.6195	5.5318
270	1.2419	3.9535	310	1.5369	5.7098	350	1.5313	5.6522
271	1.3408	4.6652	311	1.5789	6.0245	351	1.5009	5.606
272	1.4916	5.0152	312	1.563	5.8668	352	1.476	5.6187
273	1.4833	5.1074	313	1.46	5.2192	353	1.5786	5.6733
274	1.381	4.7367	314	1.4142	5.3613	354	1.4831	5.5214
275	1.3207	4.7058	315	1.3733	5.4151	355	1.5199	5.5845
276	1.3425	4.5717	316	1.5344	5.739	356	1.5922	5.4624
277	1.49	4.9783	317	1.5447	5.7368	357	1.5887	5.6533
278	1.4665	5.3421	318	1.6269	5.9413	358	1.4968	5.6832
279	1.4146	5.2614	319	1.5978	5.8773	359	1.6589	5.5447
280	1.4028	5.1238	320	1.5838	5.8336	360	1.6763	5.4693

TH-801_02610204

Sample No	Input Mmscum	Output MU	Sample No	Input Mmscum	Output MU	Sample No	Input Mmscum	Output MU
361	1.5254	5.9156	401	1.5102	5.6807	441	1.5214	5.2991
362	1.5685	5.8613	402	1.5461	5.6802	442	1.516	5.2847
363	1.593	5.6776	403	1.5919	5.6983	443	1.561	5.3937
364	1.57	5.6831	404	1.4594	5.4908	444	1.4002	4.3372
365	1.6136	5.5962	405	1.4795	5.5897	445	1.4143	5.1852
366	1.6965	5.7247	406	1.381	5.6056	446	1.5418	4.9834
367	1.7126	5.7391	407	1.4862	5.6184	447	1.4326	5.0081
368	1.6633	5.761	408	1.509	5.6667	448	1.4615	4.8435
369	1.6589	5.8093	409	1.5133	5.7054	449	1.5218	5.0014
370	1.5945	5.8111	410	1.5168	5.7218	450	1.4862	5.2092
371	1.5198	5.8198	411	1.4689	5.7706	451	1.5232	5.2754
372	1.6374	5.8631	412	1.1808	3.9978	452	1.5003	5.3092
373	1.5735	5.7124	413	0.916	4.0574	453	1.437	5.198
374	1.5655	5.31	414	1.4765	5.3341	454	1.5179	5.5755
375	1.4845	5.4675	415	1.4656	5.3112	455	1.2531	4.8364
376	1.4816	5.6235	416	1.4932	5.5365	456	1.418	5.2137
377	1.5458	5.6525	417	1.4135	5.5452	457	1.426	5.4378
378	1.5226	5.7043	418	1.4613	5.5509	458	1.4	5.2222
379	1.5473	5.6844	419	1.46	5.5727	459	1.3889	4.9787
380	1.4173	5.6001	420	1.4659	5.5711	460	1.4028	4.6367
381	1.5267	5.6298	421	1.4925	5.5565	461	1.3546	4.6151
382	1.5938	5.6656	422	1.4943	5.5805	462	1.4103	4.1426
383	1.7443	5.8414	423	1.4781	5.4996	463	1.4563	4.7997
384	1.5018	5.7291	424	1.513	5.6461	464	1.4252	4.737
385	1.5968	5.7892	425	1.7171	5.6773	465	1.4413	4.7756
386	1.5377	5.4932	426	1.5543	5.7158	466	1.4349	4.9599
387	1.4734	5.7086	427	1.5696	5.7249	467	1.1952	4.1722
388	1.5027	5.6201	428	1.5681	5.6764	468	1.2605	3.94
389	1.5688	5.4349	429	1.5774	5.6057	469	1.4092	4.8165
390	1.5029	5.3902	430	1.5659	5.6933	470	1.4017	4.4719
391	1.5298	5.7404	431	1.5867	5.6658	471	1.3182	4.0426
392	1.4183	5.668	432	1.5178	5.612	472	0.9063	2.8749
393	1.4655	5.5867	433	1.565	5.4375	473	1.1171	3.4271
394	1.4751	5.5357	434	1.5505	5.3128	474	1.1321	2.2888
395	1.4581	5.5635	435	1.5014	5.1009	475	1.0871	3.2424
396	1.4222	5.4617	436	1.542	5.3041	476	1.1246	2.8123
397	1.4619	5.5473	437	1.659	5.6067	477	1.3792	4.1522
398	1.5727	5.5732	438	1.5997	5.6964	478	1.4178	4.3501
399	1.5699	5.619	439	1.5175	5.5347	479	1.3942	4.246
400	1.4833	5.6923	440	1.4492	5.2899	480	1.3717	4.2139

Sample No	Input Mmscum	Output MU	Sample No	Input Mmscum	Output MU	Sample No	Input Mmscum	Output MU
481	1.4081	4.2123	521	1.2348	4.424	561	1.2947	4.8076
482	1.3921	4.245	522	1.284	4.7181	562	1.1984	4.2725
483	1.5194	4.4006	523	1.2145	4.3307	563	1.2808	4.7571
484	1.5964	4.7967	524	1.3225	4.899	564	1.2851	4.621
485	1.6157	5.1056	525	1.3717	5.2216	565	1.2829	4.6642
486	1.6264	5.0966	526	1.1717	4.8175	566	1.1513	3.7889
487	1.636	5.2014	527	1.0005	2.6721	567	1.3332	4.718
488	1.2776	4.5022	528	1.1503	4.0102	568	1.314	4.7808
489	1.3075	4.3023	529	1.0465	3.0405	569	1.3215	4.8193
490	1.5868	5.4289	530	1.1428	4.3035	570	1.3471	5.0174
491	1.5376	5.5534	531	1.1866	3.9606	571	1.2305	4.2833
492	1.4691	5.2084	532	1.1588	3.6403	572	1.3225	4.7482
493	1.636	5.8577	533	1.1995	3.9429	573	1.2348	4.3296
494	1.2904	4.5422	534	1.3364	4.8332	574	1.3215	4.8044
495	1.6382	4.6743	535	1.3621	5.0755	575	1.2177	4.1353
496	1.6478	5.4469	536	1.284	4.9141	576	1.3075	4.8193
497	1.498	4.9146	537	1.1781	3.8019	577	1.3086	4.8132
498	1.6489	5.9131	538	1.2016	3.9578	578	1.3182	4.9313
499	1.6521	5.9033	539	1.208	4.1534	579	1.2872	4.7966
500	1.6724	5.7017	540	1.2284	4.2113	580	1.3097	4.7426
501	1.6628	5.6355	541	1.284	4.5927	581	1.3182	4.8939
502	1.6296	5.4745	542	1.3215	4.7669	582	1.3161	4.8427
503	1.6553	5.54	543	1.3429	4.9341	583	1.2284	4.368
504	1.6488	5.6014	544	1.3204	4.8847	584	1.2626	4.4404
505	1.6628	5.5977	545	1.3536	5.1006	585	1.2808	4.5692
506	1.6703	6.0881	546	1.3022	4.921	586	1.3033	4.5241
507	1.6125	5.8677	547	1.3493	4.868	587	1.3022	4.5065
508	1.4413	4.6036	548	1.3696	5.0596	588	1.2337	4.3886
509	1.4413	5.3241	549	1.3482	5.0562	589	1.3033	4.6095
510	1.376	4.4069	550	1.3493	5.0899	590	1.3193	4.8156
511	1.452	4.6676	551	1.2915	4.8116	591	1.2551	4.4135
512	1.3354	5.8898	552	1.2647	4.6472	592	1.2819	4.1596
513	1.3664	3.526	553	1.2808	4.7844	593	1.3707	4.8925
514	1.5494	5.7493	554	0.7008	3.4926	594	1.2391	4.4775
515	1.4028	4.8649	555	0.5339	0.5415	595	1.2273	4.2588
516	1.3418	4.608	556	1.2861	4.7351	596	1.2883	3.8693
517	1.3546	4.5833	557	1.2669	4.3244	597	1.36	4.9252
518	1.2787	4.257	558	1.2883	4.534	598	1.3311	4.608
519	1.3503	4.9746	559	1.2487	4.417	599	1.314	4.7617
520	1.3193	4.9995	560	1.3065	4.8166	600	1.3161	4.7271

TH-801_02610204

Sample No	Input (Mmscum)	Output (MU)	Sample No	Input (Mmscum)	Output (MU)	Sample No	Input (Mmscum)	Output (MU)
601	1.3097	4.5577	617	1.3022	4.8304	633	1.2979	4.8111
602	1.3621	5.0799	618	1.3643	5.1449	634	1.1909	4.4569
603	1.3108	4.7921	619	1.3568	5.0312	635	1.1952	4.291
604	1.2851	4.5264	620	1.3974	5.2211	636	1.1609	4.2125
605	1.2337	4.3612	621	1.4049	5.2859	637	1.1214	4.2567
606	1.3536	4.9453	622	1.3075	4.777	638	1.1224	4.2445
607	1.3728	5.0658	623	1.2594	4.5762	639	1.1695	4.2372
608	1.3652	4.2713	624	1.2829	4.413	640	1.3461	4.9516
609	1.3803	4.602	625	1.4017	5.3109	641	1.3461	5.1956
610	1.3964	5.4194	626	1.2637	4.4816	642	1.391	5.3655
611	1.2198	4.2685	627	1.1224	4.3469	643	1.4049	5.4376
612	1.33	4.697	628	1.2155	4.2982	644	1.4606	5.5094
613	1.3525	5.0446	629	1.193	4.2245	645	1.4017	5.4505
614	1.2005	4.2995	630	1.0903	4.0368	646	1.3386	5.4017
615	1.3311	4.8712	631	1.0411	3.8126			
616	1.3225	4.8863	632	1.1984	4.0609			

Bibliography

[1] M. Jamshidi, *Large-Scale Systems : Modelling, Control and Fuzzy Logic*. Prentice Hall PTR, 1997.

[2] A. Kamiya, S. J. Ovaska, and R. Roy, “Fusion of soft computing and hard computing for large-scale plants : a general model,” *J. Applied Soft Computing*, vol. 5, pp. 265–279, 2005.

[3] V. Cherkassky, *Fuzzy Inference Systems : A Critical Review, Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications*, O. Kayak and L. A. Zadeh, Eds. Springer, Berlin, 1998.

[4] Y. Dote and S. J. Ovaska, “Industrial applications of soft computing : a review,” in *Proc. of the IEEE*, 2001, pp. 1243–1265.

[5] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas imminent in nervous activity,” *Bull. of Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[6] D. O. Hebb, *The Organization of Behaviours*. John Wiley & Sons, New York, 1949.

[7] F. Rosenblatt, *Principles of Neuro-Dynamics: Perceptrons and the theory of Brain Mechanisms*. New York: Spartan, 1962.

[8] B. Widrow and M. E. Hoff, “Adaptive switching circuits,” *IRE WESTCON Convention Record*, vol. 4, pp. 96–104, 1960.

[9] M. L. Minsky and S. Papert, *Perceptrons*. Cambridge, MA:M.I.T. Press, 1969.

[10] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” in *Proc. Nat. Acad. Sci., USA*, vol. 79, pp. 2554–2558, 1982.

[11] D. B. Parker, “Learning logic,” Massachusetts Institute of Technology, Centre for Computational Research in Economics and Management Sci., Cambridge, MA., Tech. Rep. TR-47, 1985.

TH-801_02610204

[12] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*. Foundations MIT Press, Cambridge MA, 1986, vol. 1.

[13] G. A. Carpenter and S. Grossberg, "Art2 self organisation of stable category recognition codes for analog input patterns," *Applied Optics*, vol. 26, pp. 4919–4930, 1987.

[14] T. Kohonen, *Self Organisation and Associative Memory*, 3rd ed. Springer Verlag, Heidelberg, 1989.

[15] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic system using neural networks," *IEEE Trans. on Neural Netw.*, vol. 1, no. 1, pp. 4–27, March 1990.

[16] M. M. Polycarpou and P. A. Ioannou, "Identification and control of nonlinear systems using neural network models: Design and stability analysis," University of Southern Cal. Los Angeles, Tech. Rep. 91-09-01, September 1991.

[17] R. M. Sanner and J. J. E. Slotine, "Gaussian networks for direct adaptive control," in *Proc. American Control Conf., ACC91*, 1991, pp. 2153–2159.

[18] D. T. Pham and X. Liu, "Identification of linear and nonlinear dynamic systems using recurrent neural networks," *Artificial Intell. in Engineering*, vol. 8, pp. 67–75, 1993.

[19] E. B. Kosmatopoulos, P. Ioannou, and M. A. Christodoulou, "Identification of nonlinear systems using dynammic neural network structures," in *Proc. 31st. Conf. on Decision and Control, Tucson, Arizona*, December 1992, pp. 20–25.

[20] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two state neurons," in *Proc. Nat. Acad. Sci.*, vol. 81, 1984, pp. 3088–3092.

[21] M. A. Cohen and S. Grosberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC 13, pp. 815–826, 1983.

[22] B. Hong, X. Yun, and C. Xinkuo, "Generalized fuzzy RBF networks and nonlinear system identifications," in *Proc. of the 4th World Congress on Intelligent Control and Automation, Shanghai, China*, June 10–14 2002, pp. 2508–2512.

TH-801_02610204

[23] S. M. Ahmad, M. H. Shaheed, A. J. Chipperfield, and M. O. Tokhi, "Nonlinear modelling of a twin rotor mimo system using radial basis function netw." in *Proc. of the IEEE Nat. Aerospace and Electronics Conf. (NAECON)*, 2000, pp. 313–320.

[24] R. R. Selmic and F. L. Lewis, "Multi model neural networks identification and failure detection of nonlinear systems," in *Proc. of the 40th IEEE Conf. on Decision and Control, Orlando, Florida USA*, Dec 2001, pp. 3128–3133.

[25] F. Azam and H. F. Vanlandingham, "An alternate radial basis function neural network model," in *Proc. of the IEEE Int. Conf. on Syst., Man and Cybern.*, 2000, pp. 2679–2684.

[26] R. Grino, G. Cembrano, and C. Torras, "Nonlinear system identification using additive dynamic neural networks- two on-line approaches," *IEEE Trans. Circuits and Syst.-Part 1: Fundamental Theory and Appl.*, vol. 47, no. 2, pp. 150–165, February 2000.

[27] S. Haykin, *Neural Networks - A Comprehensive Foundation*, 4th ed. Pearson Education (Singapore) Pvt. Ltd., Indian Branch, 2003.

[28] J. M. Zurada, *Introduction to Artificial Neural Systems*. Jaico Publishing House 121, Mumbai, 1999.

[29] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, 2nd ed. PWS Publishing, Boston, MA, USA, 1996.

[30] Q. Song, L. Yin, and Y. C. Soh, "Robust adaptive identification of nonlinear system using neural network," in *Proc. IEEE Signal Processing Society Workshop*, 2000, pp. 95–104.

[31] J. C. Patra, A. C. Kot, and Y. Q. Chen, "Chebyschev functional link artificial neural networks for nonlinear dynamic system identification," in *Proc. IEEE Conf. on Syst., Man and Cybern.*, Oct 2000, pp. 2655–2660.

[32] J. C. Patra and A. C. Kot, "Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks," *IEEE Trans. Syst., Man and Cybern.*, vol. 32, no. 4, pp. 505–511, Aug. 2002.

[33] X. M. Ren, A. B. Rad, P. T. Chan, and W. L. Lo, "Identification and control of continuous-time nonlinear systems via dynamic neural networks," *IEEE Trans. Industrial Electron.*, vol. 50, no. 3, pp. 478–486, June 2003.

TH-801_02610204

[34] M. Sugeno and G. T. Kang, “Structure identification of fuzzy model,” *Fuzzy Sets and Syst.*, vol. 28, pp. 15–33, 1988.

[35] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Trans. Syst., Man and Cybern.*, vol. 15, pp. 116–132, 1985.

[36] L. A. Zadeh, “Fuzzy sets,” *Inform. and Control*, vol. 8, pp. 338–353, 1965.

[37] ——, “Outline of a new approach to the analysis of complex systems and decision process,” *IEEE Trans. Syst., Man and Cybern.*, vol. 3, pp. 28–44, 1973.

[38] R. M. Tong, “Analysis and control of fuzzy systems using finite discrete relations,” *Int. J. Control*, vol. 27, pp. 431–440, 1978.

[39] E. Czogala and W. Pedrycz, “On identification in fuzzy systems and its applications in control problems,” *Fuzzy Sets Syst.*, vol. 6, pp. 73–83, 1981.

[40] W. Pedrycz, “An identification algorithm in fuzzy relational systems,” *Fuzzy Sets and Syst.*, vol. 13, pp. 153–167, 1984.

[41] E. H. Mamdani, “Applications of fuzzy algorithms for control of simple dynamic plants,” *Proc. IEE.*, vol. 121, pp. 1585–1588, 1974.

[42] T. T. Lie, G. B. Shrestha, and A. Ghosh, “Design and application of a fuzzy logic control scheme for transient stability enhancement in power systems,” *Electric Power Syst. Research*, vol. 33, pp. 17–23, 1995.

[43] P. S. Babu, A. Ghosh, and Sachchidanand, “An optimal fuzzy PI controller for a permanent magnet dc motor,” *Proc. Int. Conf. Automation IC-AUTO’95*, vol. 121, pp. 713–716, 1995.

[44] T. J. Procyk, “A linguistic self organizing process controller,” *Automatica*, vol. 15, pp. 15–30, 1979.

[45] Y. M. Park, U. C. Moon, and K. Y. Lee, “A self organising fuzzy logic controller for dynamic systems using a fuzzy auto regressive moving average model,” *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 75–82, 1995.

[46] B. P. Graham and R. P. Newell, “Fuzzy identification and control of liquid level rig,” *Fuzzy Sets and Syst.*, vol. 26, pp. 253–273, 1983.

TH-801_02610204

[47] E. Sanchez, "Resolution of composite relation equations," *Inf. and Control*, vol. 30, pp. 38–48, 1976.

[48] W. Pedrycz, "Numerical and applicational aspects of fuzzy relational equations," *Fuzzy Sets and Syst.*, vol. 11, pp. 1–18, 1983.

[49] R. M. Tong, "Synthesis of fuzzy models for industrial processes," *Int. J. Gen. Syst.*, vol. 4, pp. 143–162, 1978.

[50] B. S. Li and Z. J. Liu, "Identification of fuzzy models using fuzzy theory," *Inf. and Control*, vol. 9, no. 3, 1980.

[51] P. S. Babu, A. Ghosh, and Sachchidanand, "Fuzzy identification and control of a class of nonlinear systems," in *Proc. European Control Conf. (ECC)*, no. 8217, 1997.

[52] Y. C. Lee, C. Hwang, and Y. P. Shih, "A combined approach to fuzzy model identification," *J. Fuzzy Sets and Syst.*, vol. 24, no. 5, pp. 736–744, 1994.

[53] C. G. Moore and C. J. Harris, "Indirect adaptive fuzzy control," *Int. J. Control*, vol. 56, pp. 441–468, 1992.

[54] C. W. Xu, "Fuzzy system identification," in *Proc. IEE*, vol. 136, July 1989, pp. 146–150.

[55] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation and orthogonal least squares learning," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 807–814, 1992.

[56] K. Takana and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets and Syst.*, vol. 45, no. 2, pp. 135–156, 1992.

[57] S. G. Cao, N. W. Rees, and G. Feng, "Analysis and design for a class of complex control systems-part II: Fuzzy controller design," *Automatica*, vol. 33, pp. 1029–1039, 1997.

[58] Z. Huaguang and Q. Yongbing, "Modelling, identification and control of a class of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 2, pp. 349–354, 2001.

[59] M. Sugeno and Y. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, February 1993.

TH-801_02610204

[60] A. Cipriano, M. Ramos, and F. Montoya, "A new method for fuzzy model identification," in *Proc. IEEE 21st Int. Conf. Indus. Electron., Control and Instrumentation (IECON)*, vol. 2, 1995, pp. 1514–1519.

[61] A. E. Gaweda and Z. M. Zurada, "Data-driven design of fuzzy system with relational input partition," in *Proc. 10th IEEE Int. Conf. Fuzzy Syst.*, vol. 2, no. 3, 2001.

[62] D. Saez and A. Cipriano, "Fuzzy models based economic predictive control for a combined cycle power plant boiler," in *Proc. IEEE Int. Symp. on Intell. Control/Intell. Syst. and Semiotics*, Cambridge, MA, September 1999, pp. 417–422.

[63] A. L. Elshafei and F. Karray, "Fuzzy based sliding manifolds for identification of a class of nonlinear systems," in *Proc. 9th IEEE Int. Conf. Fuzzy Syst.*, vol. 2, 2000, pp. 841–846.

[64] D. Saez and A. Cipriano, "Design of a supervisory predictive controller based on fuzzy models," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2001, pp. 1004–1007.

[65] M. Y. Chen and D. A. Linkens, "A systematic method for fuzzy modeling from numerical data," in *Proc. IEEE Int. Conf. Syst., Man and Cybern.*, 2001, pp. 28–33.

[66] A. Flores, D. Saez, J. Araya, M. Berenguel, and A. Cipriano, "Fuzzy predictive control of a solar power plant," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 1, pp. 58–68, February 2005.

[67] J. H. Holland, *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge MA, 1975.

[68] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall Inc., 1997.

[69] T. Kumon, M. Iwasaki, T. Suzuki, T. Hashiyama, N. Matsui, and S. Okuma, "Nonlinear system identification using genetic algorithm," in *Proc. 6th IEEE Conf. Indus. Electron. Society*, vol. 4, 2000, pp. 2485–2491.

[70] A. Akramizadeh, A. A. Farjami, and H. Khaloozadeh, "Nonlinear hammerstein model identification using genetic algorithm," in *Proc. IEEE Int. Conf. Artificial Intell. Syst. (ICAIS'02)*, 2002, pp. 351–356.

[71] J. G. Juang, "Application of genetic algorithm and recurrent network to nonlinear system identification," in *Proc. IEEE Conf. Control Appl.*, vol. 1, 2003, pp. 129–134.

TH-801_02610204

[72] D. H. Kim and H. Lee, "Intelligent control of nonlinear power plant using immune algorithm based multiobjective optimization," in *Proc. IEEE Int. Conf. Networking, Sensing & Control*, March 2004, pp. 1388–1393.

[73] K. R. Vazquez, C. M. Fonseca, and P. J. Fleming, "Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming," *IEEE Trans. Syst., Man and Cybern.-Part A: Syst. and Humans*, vol. 34, no. 4, pp. 531–545, July 2004.

[74] J. S. R. Jang and C. T. Sun, "Neuro-fuzzy modeling and control," *Proc. IEEE*, vol. 83, no. 3, pp. 378–406, March 1995.

[75] Y. Lin and G. A. Cunningham-III, "A new approach to fuzzy-neural system modelling," *IEEE Trans. on Fuzzy Syst.*, vol. 3, no. 2, pp. 190–198, May 1995.

[76] J. S. R. Jang, "Input selection for ANFIS learning," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, vol. 2, 1996, pp. 1493–1499.

[77] S. Chiu, "Selecting input variables for fuzzy models," *J. Intell. and Fuzzy Syst.*, vol. 4, no. 4, pp. 243–256, 1996.

[78] M. A. Denai, F. Palis, and A. Zeghbib, "ANFIS based modelling and control of nonlinear systems: A tutorial," in *Proc. of the IEEE Int. Conf. on Syst., Man and Cybern.*, vol. 4, 2004, pp. 3433–3438.

[79] ——, "Modelling and control of nonlinear systems using soft computing techniques," *Applied Soft Computing*, vol. 7, pp. 728–738, 2007.

[80] H. Ishibuchi, R. Fujioka, and H. Tanaka, "Neural networks that learn from fuzzy if-then rules," *IEEE Trans. on Fuzzy Syst.*, vol. 1, no. 2, pp. 85–97, May 1993.

[81] C. F. Juang and C. T. Lin, "An on-line self constructing neural fuzzy inference network and its application," *IEEE Trans. on Fuzzy Syst.*, vol. 6, no. 1, pp. 12–32, Feb 1998.

[82] E. G. Sanchez, J. M. C. Izquierdo, M. J. A. Bravo, Y. A. Dimitriadis, and J. L. Coronado, "Adaptive IMC using fuzzy neural networks for the control on nonlinear systems," in *Research Paper*. ESPRIT Project No. 22416 "MONNET", Leon, Spain.

TH-801_02610204

[83] Y. Li, X. Zhao, and L. Jiao, “A nonlinear system identification approach based on neurofuzzy networks,” in *Proc. of the ICSP.*, vol. 3, 2000, pp. 1594–1597.

[84] S. Kawaji and Y. Chen, “Soft computing approach to nonlinear system identification,” in *Proc. IEEE Conf. on Indus. Electron. Society*, vol. 3, 2000, pp. 1803–1808.

[85] S. Wu, M. Joo, and Y. Gao, “A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks,” *IEEE Trans. on Fuzzy Syst.*, vol. 9, no. 4, pp. 578–594, Aug 2001.

[86] Y. Gao and M. Joo, “Nonlinear identification and control using a generalized fuzzy neural network,” in *Proc. 41st IEEE Conf. on Decision and Control, Las Vegas, Nevada USA*, vol. 2, Dec 2002, pp. 1363–1368.

[87] M. Panella and A. S. Gallo, “An input-output clustering approach to the synthesis of ANFIS networks,” *IEEE Trans. on Fuzzy Syst.*, vol. 13, no. 1, pp. 69–81, Feb 2005.

[88] S. Thangavel, V. Palanisamy, K. Duraiswamy, and S. C. Pandian, “Fuzzy identification and modeling of an intelligent controller for adaptive control of reactive power in a utility system using anfis,” *The J. CPRI.*, vol. 3, no. 1, pp. 29–35, September 2006.

[89] A. Abraham and B. Nath, “A neuro-fuzzy approach for modelling electricity demand in victoria,” *J. Applied Soft Computing*, vol. 1, pp. 127–138, 2001.

[90] A. Kamia, S. J. Ovaska, R. Roy, and S. Kobayashi, “Fusion of soft computing and hard computing for large scale plants: A general model,” *J. Applied Soft Computing*, vol. 5, pp. 265–279, 2005.

[91] J. S. R. Jang, “ANFIS: Adaptive-network-based fuzzy inference systems,” *IEEE Trans. on Syst., Man and Cybern.*, vol. 23, no. 3, pp. 665–684, May/June 1993.

[92] Y. Tsukamoto, M. M. Gupta, R. K. Ragade, and R. R. Yager, “An approach to fuzzy reasoning method,” in *Advances in Fuzzy Set Theory and Application*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds., North-Holland, Amsterdam, 1979, pp. 137–149.

[93] J. S. R. Jang and C. T. Sun, “Functional equivalence between radial basis function networks and fuzzy inference systems,” *IEEE Trans. on Neural Netw.*, vol. 4, no. 1, pp. 156–159, January 1993.

TH-801_02610204

[94] A. Pinkus, "Approximation theory of the MLP in neural networks," *ACTA Numerica*, vol. 8, pp. 143–196, 1999.

[95] H. Ying, Y. Ding, and S. L. S. Shao, "Comparison of necessary conditions for typical Takagi-Sugeno and Mamdani fuzzy systems as universal approximators," *IEEE Trans. Syst. Man Cybern. Part A*, vol. 29, pp. 508–514, 1999.

[96] K. Liu, S. Subbarayan, R. R. Shoults, M. T. Manry, C. Kwan, F. I. Lewis, and J. Naccarino, "Comparison of very short term load forecasting techniques," *IEEE Trans. Power Syst.*, vol. 11, pp. 877–882, 1996.

[97] A. Khotanzad, R. A. Rohani, and D. Maratukulam, "ANNSTLF-Artificial neural network short term load forecaster-generation," *IEEE Trans. Power Syst.*, vol. 13, pp. 1413–1422, 1998.

[98] H. S. Hipert, C. E. Pedreira, and R. C. Souza, "Neural networks for short term load forecasting: a review and evaluation," *IEEE Trans. Power Syst.*, vol. 16, pp. 41–55, 2001.

[99] K. S. Swarup and B. Satish, "Integrated ANN approach to forecast load," *IEEE, Comput. Appl. Power*, vol. 15, pp. 46–51, 2002.

[100] S. Makiko and K. Yoshitsugu, "Demand forecasting by the neural network with fourier transform," in *Proc. IEEE Int. Joint Conf. on Neural Netw.*, vol. 4, 2004, pp. 2759–2763.

[101] N. Meade, "Forecasting with growth curves: The effect of error structure," *J. Forecast.*, vol. 7, no. 4, pp. 235–244, 2006.

[102] G. E. P. Box, G. M. Jenkins, and G. C. Reinsell, *Time Series Analysis : Forecasting and Control*, 3rd ed. Prentice-Hall Englewood, 1994.

[103] M. Adya, F. Callopy, J. S. Armstrong, and M. Kennedy, "Automatic identification of time series features for rule based forecasting," *Int. J. of Forecast.*, vol. 17, no. 10, pp. 143–157, 2001.

[104] R. Gao and L. Tsoukalas, "Neural-wavelet methodology for load forecasting," *J. Intell. and Robotic Syst.*, vol. 31, no. 1, 2001.

[105] L. D. S. Coelho and R. A. Krohling, "Nonlinear system identification based on B-spline neural network and modified particle swarm optimization," in *Int. Joint Conf. on Neural Netw.*, July 2006, pp. 3748–3753.

TH-801_02610204

[106] H. Saito and T. Abe, "A traffic forecasting method taking into account of outliers," *Electron. and Communications in Japan (part I:Communications)*, vol. 69, no. 10, pp. 85–92, March 2007.

[107] N. Amjady, "Short term hourly load forecasting using time series modeling with peak load estimation capability," *IEEE Trans. Power Syst.*, vol. 16, pp. 789–805, 2001.

[108] L. Mili, T. V. Cutsem, and M. R. Pavella, "Bad data identification methods in power system state estimation: A comparative study," *IEEE Trans. on Power Apparatus and Syst.*, vol. PAS-104, no. 11, pp. 3037–3049, November 1985.

[109] B. M. Zhang and K. L. Lo, "A recursive measurement error estimation identification method for bad data analysis in power system state estimation," *IEEE Trans. on Power Syst.*, vol. 6, no. 1, pp. 191–198, February 1991.

[110] A. Abur, "A bad data identification method for linear programming state estimation," *IEEE Trans. on Power Syst.*, vol. 5, no. 3, pp. 894–901, August 1990.

[111] G. Long and F. Ling, "A new complex system identification method and its application to echo canceller fast initialization," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP-90*, vol. 3, April 1990, pp. 1671–1674.

[112] A. Pankratz, *Forecasting with Univariate Box-Jenkins Models: Concepts and cases*. Wiley, 1983.

[113] A. D. Papalexopoulos and T. C. Hesterberg, "A regression based approach to short term system load forecasting," *IEEE Trans. Power Syst.*, vol. 5, pp. 1535–1547, 1990.

[114] J. G. D. Gooijer and K. Kumar, "Some recent developments in nonlinear time series modeling, testing and forecasting," *Int. J. Forecast.*, vol. 8, pp. 135–156, 1992.

[115] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," *Int. J. Forecast.*, vol. 14, pp. 35–62, 1998.

[116] P. A. Mastorocostas, J. B. Theocharis, and A. G. Bakirtzis, "Fuzzy modeling for short term load forecasting using the orthogonal least squares method," *IEEE Trans. Power Syst.*, vol. 14, pp. 29–36, 1999.

TH-801_02610204

[117] M. R. Khan, A. Abraham, and C. Ondrusek, "Short term load forecasting models in Czec. Republic using soft computing techniques," *Int. J. Knowledge-Based Intell. Engineering Syst.*, vol. 7, no. 4, pp. 172–179, 2003.

[118] M. Verleysen, E. D. Bodt, and A. Lendasse, *Engineering applications of Bio-inspired Artificial Neural Networks*. Springer Berlin / Heidelberg, 1999, vol. 1607.

[119] Z. J. Wang, Y. Zhao, C. J. Wu, and Y. T. Li, "Application of kernel smoothing to time series data," *ACTA Mathematicae Applicatae sinica*, vol. 22, no. 2, pp. 219–226, April 2006.

[120] R. T. Clemen, "Combining forecasts: a review and annotated bibliography," *Int. J. Forecast.*, vol. 5, pp. 559–583, 1989.

[121] K. H. Kim, H. S. Youn, and Y. C. Kang, "Short term load forecasting for special days in anomalous load condition using neural networks and fuzzy inference method," *IEEE Trans. Power Syst.*, vol. 15, pp. 559–565, 2000.

[122] A. Abraham and B. Nath, "A neural-fuzzy approach for modelling electricity demand in Victoria," *Applied Soft Computing*, vol. 1, pp. 127–138, 2001.

[123] F. C. Morabito and M. Versaci, "Fuzzy neural identification and forecasting techniques to process experimental urban air pollution data," *Neural Netw.*, vol. 16, no. 3–4, pp. 493–506, 2003.

[124] B. M. Zhang, S. Y. Wang, and N. D. Xiang, "A linear recursive bad data identification method with real time application to power system state estimation," *Trans. on Power Syst.*, vol. 7, no. 3, pp. 1378–1385, August 1992.

[125] M. Kandemir and J. Ramanujam, "Data relation vectors: A new abstraction for data optimizations," *IEEE Trans. on Computers*, vol. 50, no. 8, pp. 798–810, August 2001.

[126] S. A. Ahmed and J. R. Cruz, "Complex system identification methods for fast echo canceller initialization," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP-92*, March 1992, pp. 525–528.

[127] D. Erdogmus and J. C. Principe, "An error entropy minimization for supervised training of nonlinear systems," *IEEE Trans. on Signal Processing*, vol. 50, no. 7, pp. 1780–1786, July 2002.

TH-801_02610204

[128] V. I. Dedenko and A. L. Movchan, "Minimization of number of metrological parameters for data acquisition systems," *IEEE Trans. on Instrumentation and Measurement*, vol. 51, no. 1, pp. 88–91, February 2002.

[129] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. of the ACM, SIGKDD 02*, 2002.

[130] B. Babcock, S. Babu, M. Datar, and R. Motwani, "Chain: Operator scheduling for memory minimization in data stream systems," in *Proc. of ACM, SIGMOD 2003*, June 2003.

[131] D. Weekes and G. B. Fogel, "Evolutionary optimization, backpropagation and data preparation issues in QSAR modeling of HIV inhibition by HEPT derivatives," *Bio. Sys.*, vol. 72, pp. 149–158, 2003.

[132] T. Meekhof and R. B. Heckendorn, "Using evolutionary optimization to improve markov based classifcation with limited training data," in *Proc. of ACM GECCO'05*, 2005, pp. 2211–2212.

[133] J. Pekar and J. Stecha, "Identification and predictive control by p-norm minimization," *Acta Polytechnica*, vol. 46, no. 1, pp. 33–39, 2006.

[134] Y. Lii, J. Huang, and Q. Liu, "Improving statistical machine translation performance by training data selection and optimization," in *Proc. Joint Conf. on Empirical Methods in Natural Language Processing and Comput. Natural Language Learning*, June 2007, pp. 343–350.

[135] G. E. Dieter, *Engineering Design: A Material and Processing Approach*. McGraw-Hill Inc, 1991.

[136] A. Kohli and U. S. Dixit, "A neural-network-based methodology for the prediction of surface roughness in a turning process," *The Int. J. Advanced Manufact. Technology*, vol. 25, no. 1-2, pp. 118–129, January 2005.

[137] M. Stone, "Cross validation choice and assessment of statistical predictions," *J. Royal Statistical Society*, vol. 36 (B), pp. 111–147, 1974.

[138] C. L. Sabharwal, "An implementation of hybrid approach to indexing image data bases," in *The ACM Symp. on Applied Comput.*, Feb 28–Mar 2 1999, pp. 421–426.

TH-801_02610204

[139] P. Zhang, "Model selection via multifold cross validation," *Annals of Statistics*, vol. 21 (1), pp. 299–313, 1993.

[140] O. Nelles, *Nonlinear System Identification*. Springer-Verlag, Berlin Heidelberg, Germany, 2000.

[141] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 3rd ed. Addison Wesley Longman, Inc., 2000.

[142] L. C. Jain and N. M. Martin, *Fusion of Neural Networks, Fuzzy Sets and Genetic Algorithm: Industrial Applications*, Int. Series on Computational Intelligence ed. CRC Press, 1999.

[143] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control action," in *Proc. IFAC Symp. Fuzzy Inform., Knowledge Representation and Decision Analysis*, July 1983, pp. 55–60.

[144] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[145] S. L. Chiu, "A cluster estimation method with extension to fuzzy model identification," in *Proc. 3rd IEEE Conf. on Fuzzy Syst., World Congress in Comput. Intell.*, vol. 2, June 1994, pp. 1240–1245.

[146] M. H. Hayes, *Statistical Digital Signal processing and Modelling*. New York: John Wiley & Sons Inc., 1996.

TH-801_02610204

Related Publications by the Author

Journal Publication:

1. M. Buragohain and C. Mahanta, " A Novel Approach for ANFIS Modeling based on Full Factorial Design," *Applied Soft Computing*, vol. 8, Issue 1, Jan 2008, pp. 609-625.

2. M. Buragohain and C. Mahanta, "V-fold Technique based ANFIS Model for Complex Large Scale Systems," *International Journal of Artificial Intelligence*, special issue on Theory and Application of Soft Computing, vol. 1, Issue A08, Autumn 2008, pp. 34-55.

3. M. Buragohain and C. Mahanta, "Data Optimization based Modeling of Complex Large Scale Systems: A Comparative Analysis with Soft Computing and Statistical Techniques," *Journal of Systems Science and Engineering*, Vol. 17, No. 1, June 2008.

Communicated to Journals:

1. M. Buragohain and C. Mahanta, "ANFIS based Modelling of Thermal Power Plant Using Full Factorial Design," to *Engineering Application of Artificial Intelligence*.

Conference Publications:

1. M. Buragohain and C. Mahanta, " Data Optimization based Modeling of Complex Systems: A Comparative Analysis with Soft Computing and Statistical Techniques," in *Proceedings of National Systems Conference*, Manipal, Dec, 2007.

2. M. Buragohain and C. Mahanta, " Grid Partition based Neuro-Fuzzy Technique and Statistical Modeling of Complex Systems," in *Recent Trends in IT and Soft Computing(ITSC)*, IMT Nagpur, February 2007, pp. 47-58.

3. M. Buragohain and C. Mahanta, " Complex Systems Modeling with Grid Partitioning based Soft Computing Technique," in *Recent Trends in IT and Soft Computing(ITSC)*, IMT Nagpur, February 2007, pp. 89-97.

4. M. Buragohain and C. Mahanta, " ANFIS Modeling of Nonlinear System based on V-fold Technique," in *Proceedings of IEEE International Conference on Industrial Technology*, Mumbai, 15-17 Dec. 2006, pp. 2178-2183.

TH-801_02610204

5. M. Buragohain and C. Mahanta, “ ANFIS Modeling of Nonlinear Systems based on Combined FFD-V-fold Technique,” in *Proceedings of IEEE International Conference on Industrial Technology*, Mumbai, 15-17 Dec. 2006, pp. 2462-2467.

6. M. Buragohain and C. Mahanta, “ Full Factorial Design based ANFIS Model for Complex Systems,” in *Proceedings of IEEE Annual India Conference*, New Delhi, Sept. 2006.

7. M. Buragohain and C. Mahanta, “ ANFIS Modeling of Nonlinear Systems based on Subtractive Clustering and V-fold Technique,” in *Proceedings of IEEE Annual India Conference* , New Delhi, Sept 2006.

8. M. Buragohain and C. Mahanta, “ ANFIS Modeling of Nonlinear Systems based on Subtractive Clustering and Combined FFD-V-fold Technique,” in *Proceedings of IEEE Annual India Conference* , New Delhi, Sept. 2006.

9. M. Buragohain and C. Mahanta, “ Modeling of Complex Systems with Soft Computing Techniques,” in *Proceedings of National Systems Conference* , Goa, Nov 2006.

10. M. Buragohain and C. Mahanta, “ Statistical and Neuro-Fuzzy Technique based Modeling of Complex Systems ,” in *Proceedings of National Systems Conference* , Goa, Nov 2006.

11. M. Buragohain and C. Mahanta, “ Modeling of Thermal Power Plants Using Full Factorial Design based ANFIS ,” in *Proceedings of IEEE International conference on Cybernetics and Intelligent Systems*, Bangkok, June 2006.

TH-801_02610204

Bio-data of the Author

Mr. M. Buragohain received the Bachelor of Engineering degree in Electrical Engineering. in 1992 from Guwahati University, Guwahati. He obtained the Master of Engineering in High Voltage Engineering from Jadavpur University in 2000. He next joined the PhD program in Electronics and Communication Engg. in IIT Guwahati in 2003. His research interests lie in nonlinear system identification and soft computing. He is currently working as a Senior Lecturer in the Department of Electrical Engineering, Jorhat Engineering College, Assam.

TH-801_02610204