SENSOR VALIDATION SCHEMES USING NEURO-FUZZY LOGIC ESTIMATORS FOR NON-LINEAR PLANT WITH SLOW-FAST MOTION STUDY

A THESIS

Submitted by

KUMARESAN M

in partial fulfillment for the award of the degree of

DOCTOR OF PHILOSOPHY

Department Of Electrical And Electronics Engineering
FACULTY OF ENGINEERING & TECHNOLOGY

Dr.M.G.R EDUCATIONAL AND RESEARCH INSTITUTE,
UNIVERSITY
(u/s 3 of the UGC Act, 1956)
CHENNAI – 600095

MAY 2013
SENSOR VALIDATION SCHEMES USING NEURO-FUZZY LOGIC ESTIMATORS FOR NON LINEAR PLANT WITH SLOW-FAST MOTION STUDY

A THESIS

Submitted by

KUMARESAN M

in partial fulfillment for the award of the degree of

DOCTOR OF PHILOSOPHY

Department Of Electrical And Electronics Engineering
FACULTY OF ENGINEERING & TECHNOLOGY

Dr.M.G.R
EDUCATIONAL AND RESEARCH INSTITUTE,
UNIVERSITY
(u/s 3 of the UGC Act, 1956)
CHENNAI – 600095
MAY 2013

DECLARATION BY THE CANDIDATE

I declare that the thesis entitled “Sensor Validation Schemes Using Neuro-Fuzzy Logic Estimators for Non Linear Plant with Slow-Fast Motion Study” submitted by me for the degree of Doctor of Philosophy is a bonafide record of research work carried out by me during the period from November-08 to January-13 under the supervision of Dr. S. RAVI and has not formed the basis for the award of any degree, diploma, associate-ship, fellowship, titles in this or any other University or other similar institution of higher learning and devoid of any plagiarism.

I have also published my papers in International Journals (Scopus rated) as per list of publications in the Annexure.

Signature of the Research Scholar

[M. Kumaresan]
BONAFIDE CERTIFICATE

Certified that the thesis titled “Sensor Validation Schemes Using Neuro-Fuzzy Logic Estimators for Non Linear Plant with Slow-Fast Motion Study” is the bonafide work of Mr. M. KUMARESAN, who had carried out the research under my supervision and devoid of any plagiarism to the best of my knowledge. Certified further, that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or diploma was conferred on an earlier occasion on this or any other scholar.

Signature of the Supervisor

(Dr. S. Ravi)

Supervisor
ABSTRACT

Measurements are used to monitor and ultimately, to control and optimize processes. Difficulties in measuring quality (primary) variables inevitably mean poor or no control at all. Online sensors may be available but they may suffer from long measurement delays (e.g. gas chromatographs) or may be subject to factors that affect the reliability of the sensor (e.g. drifts and fouling). In this work, a nonlinear process control system (real time) assisted with soft estimator and a fuzzy controller, with an objective to model the relationship between a primary output and secondary outputs and inputs is developed, implemented and tested. The system response is obtained for different types of nonlinearities and for slow and fast motion inputs. The phase-plane portrait is obtained for each case and the system stability is evaluated. The construction of a parameter (or state) estimator can be basically considered as a function approximation problem. To design an estimator, it is first necessary, to obtain the training data set ‘G’ such that, this training data set contains as much information as possible about a system ‘g’. Once trained properly, the estimator will adaptively follow the slope of ‘g’ at all times. In this research, signals are processed in real time and combined with previous monitoring data to estimate, the process variable level in a nonlinear process control plant.

Key words: Nonlinear control, Sensor validation. Fuzzy Estimator, Neural Network, Fuzzy logic controller, Slow motion system, Fast motion system.
ACKNOWLEDGEMENT

I like to express my sincere thanks to Revered founder Mr. A. C. Shanmugam and President Er. A. C. S. Arun Kumar, Dr. M. G. R Educational and Research Institute, University, Chennai -95, for creating a conducive environment and providing obligatory infrastructure for development and implementation of this work.

I express my thanks to Dr. K. Meer Mustafa Hussain, Vice Chancellor, Dr. M. G. R Educational and Research Institute, University, Chennai -95, for his enthusiastic support and insightful ideas for this work to flourish.

I express my thanks to Dr. A. Thirunavukkarasu, Dean (Research), Dr. M. G. R Educational and Research Institute, University, Chennai -95, for his enthusiastic support and insightful ideas for this work to flourish.

I express my sincere thanks to Dr. Uma Rajaram, Dean (E & T), Dr. M. G. R Educational and Research Institute, University, Chennai -95, for the amicable support provided to carry out this work.

I extend my genuine thanks with gratitude to my guide Dr. S. Ravi, Professor and Head, Electronics and Communication Engineering, Dr. M. G. R Educational and Research Institute, University, Chennai -95, for his subtle and considerate approach in shaping, guiding and directing me towards effective culmination of this work.

I thank my parents, my wife, and my sons who offered unconditional love and has stood by me all along. I am grateful to my friends, in-laws, and sisters for their unwavering support.
M.KUMARESAN

TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER NO.</th>
<th>TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Abstract</td>
<td>iv</td>
</tr>
<tr>
<td></td>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td></td>
<td>List of Tables</td>
<td>xiii</td>
</tr>
<tr>
<td></td>
<td>List of Symbols and Abbreviations</td>
<td>xiii</td>
</tr>
</tbody>
</table>

1. INTRODUCTION

1.1 Control Methods

1.1.1 Desired Features of A Feedback Control Algorithm 1
1.1.2 PID Controller 1
1.1.3 Conventional PID Controller 2
1.1.4 Adaptive Control 4

1.1.4.1 Types of Adaptive System 5
1.1.4.2 Construction Steps of an Adaptive Controller 6
1.1.5 Model Reference Adaptive Control 6
1.1.5.1 Components of Model Reference Adaptive Controller 7
1.1.6 Hybrid Neuro Fuzzy Controller Model 8

1.2 Problem Statement and Proposed Solution 8

1.3 Focus of Research 12

1.4 Benefits of the Research Work 13

1.5 Goals of the Research Work 13

1.6 Thesis Contents 13

1.7 Objectives of the Present Work 14
2. LITERATURE SURVEY

2.1 Limitation of Conventional Constant Gain PI Controllers 15
2.2 Adaptive PI Controllers 19
2.3 Fuzzy Logic Controllers 22
2.4 Intelligent Controllers with Fuzzy Logic for State Estimation 28
2.5 Intelligent Controllers with Neural Network for State Estimation 30
2.6 Inference from Literature Survey 32

3. DESCRIPTION OF THE REAL TIME PLANT 34

3.1 Transfer Function of the Nonlinear Process 38
3.1.1 Modeling for the Cylindrical Portion of the Tank 39
3.1.2 Modeling for the Conical Portion of the Tank 39
3.2 Controllers and their Structure 40
3.3 Representation of Stable and Unstable Systems 44
3.4 Analog Controllers 45
3.5 Neural Networks and their Control Structures 46
3.6 Basic Neural Learning Model 47
3.7 ANN for System Identification 47
3.8 Forward Modeling 47
3.9 Two Layered Feed Forward Neural Network 48

4. NEURO-FUZZY LOGIC 50

4.1 Fuzzy Logic Controller 50
4.2 Configuration of FLC 53
4.2.1 Fuzzification and Membership Functions 53
4.2.2 Rules Creation and Inference 55
4.2.3 Defuzzification 55
4.3 Development Tools 55
4.3.1 Back Propagation Learning Algorithm
4.3.2 Training with Back Propagation Algorithm
4.4 Design of Fuzzy Logic Controller (FLC)
4.5 Training the Fuzzy Estimator
4.5.1 Tuning the Fuzzy Estimator Model
4.5.2 Data Pairs and Outliers Filtering
4.5.3 Number of Rules
4.5.4 Cluster Centers and Parameter ‘M’
4.6 Estimation Procedure
4.7 Plant with Sensor Validation Model Using Neural Estimator
4.8 Design of Estimator Using Neural Network
4.9 Summary

5. PERFORMANCE OF FUZZY AND NEURAL ESTIMATOR
5.1 Training Data for Neural Network
5.2 Experimental Response with the Neural Estimator
5.3 Performance of the Estimator
5.3.1 Step Response of the Estimator
5.3.2 Response of the Estimator to 0.5Hz Square Wave Input
5.3.3 Response of the Estimator to 0.05Hz Square Wave Input
5.4 Response to Low Frequency Sine Wave
5.5 Design of a Generic FLC
5.6 Functions Subroutine KB and RB Selection
5.7 Functions Subroutines for Tuning $\mu P_z$, $\mu ASO$, $\mu P_{IM}$, $\mu OUT$

6. RESULTS AND DISCUSSION
6.1 Response to Set Point Variations
6.2 Response to Variations in Load Variable Outflow
6.3 Online Acquired Plots
6.4 Comparison Results
6.5 Experimental Results 84
6.5.1 Slow Motion (SMS) and Fast Motion (FMS) Response 87

7. INTELLIGENT CONTROLLER WITH SENSOR NOISE ELIMINATION NETWORK 89
7.1 Independent Component Analysis (ICA) 91
7.1.1 Principal Component Analysis (PCA) based whitening Algorithm 92
7.1.2 Separation process 93
7.1.2.1 Neural Network based Separation Algorithm 93
7.1.3 Choice of activation function ‘g(.)’ used in the Separation Process 94
7.2 Multiple Sensor Coordination Model (with sensor noise elimination scheme and feedback sensor validation network) 94
7.3 Results 97
7.4 Performance of Sensor Noise Cancellation Network 97
7.4.1 Non Changing Environment with High Noise 97
7.4.2 Non Changing Environment with Low Noise 98
7.4.3 Changing Environment with Low Noise 99
7.4.4 Changing Environment with High Noise 101
7.5 Discussion on the performance of the Sensor Noise Elimination Network 102
7.6 Hardware Implementation Results 105

8. CONCLUSION AND FUTURE SCOPE 109
8.1 Conclusion 109
8.2 Future Scope 110
REFERENCES 111
APPENDIX – I
LIST OF PUBLICATIONS
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO.</th>
<th>TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Block diagram of PID controller</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Block diagram of Adaptive System</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Model Reference Adaptive Controller</td>
<td>7</td>
</tr>
<tr>
<td>3.1</td>
<td>Geometrical Cross-Section of the Tank Used in the Mathematical Modelling</td>
<td>34</td>
</tr>
<tr>
<td>3.2</td>
<td>P&amp;I Diagram of Nonlinear Hopper Type Tank Chosen for Experimental Study MV-Manually operated Valve and FT-Flow Transmitter</td>
<td>35</td>
</tr>
<tr>
<td>3.3</td>
<td>Laboratory Set Up of the Plant Model</td>
<td>37</td>
</tr>
<tr>
<td>3.4</td>
<td>Real Time implementation of the Nonlinear Tank</td>
<td>38</td>
</tr>
<tr>
<td>3.5</td>
<td>3.5(a) Generic Controller</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>3.5(b) Proportional Controller</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>3.5(c) PI Controller</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>3.5(d) Modified I-D Controller</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>3.5(e) Derivative Feedback with PI Controller</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>3.5(f) Derivative Feedback with Delay and PI Control</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>3.5(g) PID Controller</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>3.5(h) Derivative Feedback with Delay and P Control</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>3.5(i) P-I Control with Feedback</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>3.5(j) Sample PI Control</td>
<td>44</td>
</tr>
<tr>
<td>3.6</td>
<td>System Stability in 2-d Quadrant</td>
<td>44</td>
</tr>
<tr>
<td>3.7</td>
<td>Analog Controller PI</td>
<td>45</td>
</tr>
<tr>
<td>3.8</td>
<td>Analog Controller (PID)</td>
<td>46</td>
</tr>
<tr>
<td>3.9</td>
<td>Forward Modeling Neural Network</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Two Layer Feed forward Neural Network</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>Basic Configuration of Fuzzy Logic Controllers</td>
<td>52</td>
</tr>
</tbody>
</table>
4.2 Triangular Membership Functions 54
4.3 Gaussian Membership Functions 54
4.4 Trapezoidal Membership Functions 54
4.5 Back Propagation Network 58
4.6 Design of FLC 60
4.7 Membership Function for e(k),ce(k) and Change in Control Variable 61
4.8 Parameter Estimator usign Data Driven Model 63
4.9 Data Preprocessing 64
4.10 Plant Model with Neural Estimator to Take Care of Feedback Sensor Failure 66
4.11 Process with “u” Control Input, “d” Disturbance and “y” Measured Output 68
4.12 Neural Network during Training with Weights Adaptive 68
4.13 Trained Neural Network as Fault Tolerant System 68
4.14 Neural Network Architecture with Input Layer, Hidden layer and Output Layer 69

5.1 Performance Analysis of the Designed Estimator 73
5.2 Result of the Fuzzy Estimator with Step Input (solid line = estimator response, dotted = truth model response) 73
5.3 Result of the Fuzzy Estimator with a 0.5Hz Square Wave Input with Initial Level of the Tank at 1 Meter (solid line = estimator response, dotted = truth model response) 74
5.4 Result of the Fuzzy Estimator With a Square Wave Input with Initial Condition of h(0) = 0.5 (solid line = estimator response, dotted = truth model response) 74
5.5 Result of the Fuzzy Estimator with a 1 Hz sine Wave Input (solid line = estimator response, dotted = truth model response) 75
5.6 FL Control 76
5.7 KB and RB Initialization 77
5.8 Tuning of Coefficients 78
6.1 Set point Variations Chosen for the Experimental Study 81
6.2 Measured Variations of Manipulated Variable Inflow ( %) 82
6.3 Measured Variations of Load Variable Outflow (%)

6.4 Actual and Neural Network Estimated Level Variations of the Real time Plant with Neural Estimator

6.5 Measured and Neural Estimated Level Variations of the Real time Plant with Fuzzy Controller in response to perturbations in Load Variable Outflow.

6.6 On-line Acquired Plots Showing the True Process Value of Outflow (1st quadrant), Neural Network Estimated Value (2nd quadrant) and Estimator o/p Value (4th quadrant). (Plot obtained during regulatory tracking)

6.7 On-line Acquired Plots Showing the true Process Value of level (1st quadrant), On-line Acquired Plots Showing the True Process Value of Outflow (1st quadrant), Neural Network Estimated Value (2nd quadrant) and Fuzzy Estimator O/P Value (4th quadrant). (Plot obtained during regulatory tracking) Neural Network Estimated Value (2nd quadrant) and the Fuzzy Estimator O/P Value (4th quadrant). (Plot obtained during set point tracking)

6.8 Plant response with Dead Zone Nonlinearity

6.9 Plant response with ON/OFF Nonlinearity

6.10 Plant response with Linear Dead Zone Nonlinearity

6.11 Slow Motion System (SMS) Response

6.12 Fast Motion System (FMS) Response

7.1 Adaptive Noise Cancellation Network for Stationary x(k)

7.2 Adaptive Noise Cancellation Network to include Non-stationary x(k)

7.3 Adaptive Noise Cancellation Network to include Non-stationary x(k) with no secondary Noise Source.

7.4 ICA Network used for Removing the Noise from the Sensor Readings

7.5 Plant Model with Sensor Noise added to Inflow and Outflow Sensors

7.6 Multiple Sensor Coordination model with the ICA Network for Sensor Noise Elimination and Neural Network or Fuzzy Trained
Estimator for Feedback Sensor Validation in a Multivariable Interacting Systems. $G_c$ is the Controller and FT is the Flow Transmitter

7.7 Non Changing High Noise Environment Plot 100
7.8 Non Changing Low Noise Environment Plot 101
7.9 Changing Low Noise Environment Output 103
7.10 Changing High Noise Environment Output 104
7.11 Initial Tracking of Neural Estimator 107
7.12 Neural Estimator Setting to first Environment Value and Successfully Tracking first Change in Environment 108
7.13 Neural Estimator Setting early after First Change 109
7.14 Neural Estimator Successfully tracking Second Change 110
LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO.</th>
<th>TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Effect of Controller Parameters $K_P$, $K_I$ and $K_D$</td>
<td>3</td>
</tr>
<tr>
<td>4.1</td>
<td>Network Category Architecture And Typical Application</td>
<td>56</td>
</tr>
<tr>
<td>4.2</td>
<td>Generic Fuzzy Rule Base</td>
<td>60</td>
</tr>
<tr>
<td>5.1</td>
<td>Input Training Data</td>
<td>71</td>
</tr>
<tr>
<td>6.1</td>
<td>Normalized MSE</td>
<td>85</td>
</tr>
<tr>
<td>6.2</td>
<td>Performance Measure</td>
<td>90</td>
</tr>
<tr>
<td>7.1</td>
<td>Normalised ISE</td>
<td>105</td>
</tr>
<tr>
<td>7.2</td>
<td>Normalised MSE</td>
<td>106</td>
</tr>
<tr>
<td>7.3</td>
<td>Normalised MSE values</td>
<td>106</td>
</tr>
<tr>
<td>7.4</td>
<td>Normalised ISE values</td>
<td>106</td>
</tr>
</tbody>
</table>

LIST OF ABBREVIATIONS

PI : Proportional Integral
PID : Proportional Integral Derivative
ICA : Independent Component Analysis
PCA : Principal Component Analysis
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLC</td>
<td>Fuzzy Logic Controllers</td>
</tr>
<tr>
<td>FOPDT</td>
<td>First Order Plus Dead Time model</td>
</tr>
<tr>
<td>SMS</td>
<td>Slow Motion Study</td>
</tr>
<tr>
<td>FMS</td>
<td>Fast Motion Study</td>
</tr>
<tr>
<td>FIS</td>
<td>Fuzzy Inference System</td>
</tr>
<tr>
<td>MF</td>
<td>Membership Function</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Networks</td>
</tr>
<tr>
<td>PE</td>
<td>Processing Element</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Controller’s design and development procedures have its root on analytical study of dynamic systems response for various inputs. The external input to a system is called the reference. Conventional scenario associated with the controller projects one or more output variables pursuing specific benchmark output. The controller delivers expected output by manoeuvring the inputs. The customary goal of a control design is to estimate solution for the appropriate curative action from the controller that upshot system stability, that is, the system will clutch the set point and not fluctuate around it. Differential equations are used as the analytical tool to relate inputs and outputs of a control system. The development of a control system involves many tasks such as modeling, design of a control law, implementation and validation.

1.1 CONTROL METHODS

1.1.1 DESIRED FEATURES OF A FEEDBACK CONTROL ALGORITHM

The desired features of a feedback control algorithm are,

i. Zero offset

ii. Insensitivity to modeling parameters errors

iii. Wide applicability

iv. Simple Calculations

1.1.2 PID Controller

"PID" is an acronym for “Proportional, Integral, and Derivative”. A (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an "error" value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs.
Adaptive control system is one which can adjust its parameters automatically in such a way as to compensate for variations in the characteristics of the process it controls. The various type of adaptive control systems differ only in the way the parameter of the controller is adjusted. The types are Gain Scheduling, Model Reference Adaptive Controller, Self-Tuning Regulator and Dual Controller or Stochastic Model.

1.1.3 Conventional PID Controller

A PID controller is a controller that includes elements with those three functions as shown in figure 1.1. In the literature on PID controllers, acronyms are also used at the element level: the Proportional element is referred to as the “P element,” the Integral element as the “I element,” and the Derivative element as the “D element”. The three goals of PID controller design are,

i. Controlled variable performance
ii. Model error robustness
iii. Manipulated variable behaviour

The three elements of the PID controller produce outputs with the following nature

P element refers Proportionality to the “Present” error, error at the instant \( t \).
I element refers Proportionality to the “Past” error, Integral of the error up to the instant \( t \).
D element Proportionality to the “Future” error, Derivative of the error at the instant \( t \).

The PID controller can be understood as a controller that takes the present, the past, and the future of the error into consideration. After digital implementation was introduced, a certain change of the structure of the control system was proposed and has been adopted in many applications.

\[ (1.1) \]

\( K_P \) - Proportional Gain
\( T_I \) - Integral Time
\( T_D \) - Derivative Time
The four major characteristics of closed loop step response are,
Rise Time: The time it takes for the plant output \( y \) to rise beyond 90% of the desired level for the first time.
Overshoot: The normalized magnitude difference between actual peak level to steady state.
Settling Time: The time output takes for the system to converge to its steady state.
Steady-State Error: The difference between the steady-state output and the desired output.

The effects of increasing each of the controller parameters \( K_P \), \( K_I \) and \( K_D \) can be summarized in table 1.1.

**Table 1.1 Effect of Controller Parameters \( K_P \), \( K_I \) and \( K_D \)**

<table>
<thead>
<tr>
<th>Response</th>
<th>Rise Time</th>
<th>Overshoot</th>
<th>Settling Time</th>
<th>S-S Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_P )</td>
<td>Decrease</td>
<td>Increase</td>
<td>Nil effect in tuning</td>
<td>Decrease</td>
</tr>
<tr>
<td>( K_I )</td>
<td>Decrease</td>
<td>Increase</td>
<td>Increase</td>
<td>Eliminate</td>
</tr>
<tr>
<td>( K_D )</td>
<td>Nil effect in tuning</td>
<td>Decrease</td>
<td>Decrease</td>
<td>Nil effect in tuning</td>
</tr>
</tbody>
</table>
Typical steps for designing a PID controller are

i. Assessment of characteristic improvement scope of the system

ii. Use $K_P$ to decrease the rise time and $K_I$ to eliminate the steady-state error

iii. Use $K_D$ to reduce the overshoot and settling time.

### 1.1.4 ADAPTIVE CONTROL

An adaptive controller is one whose parameters are automatically adjusted to meet corresponding variations in the parameters of the process being controlled in order to optimize the response of the control loop. Adaptive control involves automatically detecting the changes that occur in the gain or dead time (period) of the process and readjusting the PID control mode setting, thereby adapting the loop to the changing condition. An adaptive control system can be thought of as having two loops. One loop is a normal feedback with the process and the controller. The other loop is the parameter adjustment loop. The parameter adjustment loop is often slower than the normal feedback loop. A block diagram of an adaptive system in figure 1.2.

![Figure 1.2 Block diagram of Adaptive System](image-url)
1.1.4.1 TYPES OF ADAPTIVE SYSTEM

Adaptive system is described in four types. Those are

( i ) Gain Scheduling

It is possible to find auxiliary variables in systems which correlate well with the changes in process dynamics. It is then possible to eliminate the influences of parameter variations by changing the parameters of the regulator as functions of the auxiliary variables. This method of eliminating variations in process dynamics is called Gain Scheduling.

( ii ) Model Reference Adaptive Systems (MRAS)

Model Reference Adaptive System (MRAS) was originally proposed to solve a problem in which the performance specifications are given in terms of a reference model. This model tells how the process output ideally should respond to the command signal. The controller can be thought of as consisting of two loops. The inner loop is an ordinary feedback loop composed of the process and the controller. The outer loop adjusts the controller parameters in such a way that the error, which is the difference between process output $y$ and model output $y_m$ is small.

( iii ) Self-Tuning Regulators (STR)

The controller automatically tunes its parameters to obtain the desired properties of the Closed-loop system. This type of controller called as Self-tuning Regulator (STR). It has the two loops. The inner loop consists of the process and an ordinary feedback controller. The outer loop consists of a recursive parameter estimator and a design controller.

( iv ) Dual Control (or) Stochastic Model

Regulator structures such as MRAS and STR are based on heuristic arguments. It would be appealing to obtain the regulators from a unified theoretical framework. This can, in principle, be done by using nonlinear stochastic control theory. The system and its environment are then described by a Stochastic Model (or) Dual control. The criterion is formulated so as to minimize the expected value of a loss function which is a scalar function of states and controls.

1.1.4.2 CONSTRUCTION STEPS OF AN ADAPTIVE CONTROLLER

The construction of an adaptive controller thus contains the following steps:

Characterize the desired behavior of the closed-loop system.
Determine a suitable control law with adjustable parameters.
Find the mechanism for adjusting the parameters.
Implement the control law.

The model-reference adaptive control (MRAC) is an important controller for designing a controller as an intelligent. It may be regarded as an adaptive servo system in which the desired performance is expressed in terms of a reference model, which gives the desired response to a command signal. This is a convenient way to give specification for a servo problem. The performance of the controller is depends on the controller parameters. The control parameters are found by adaptive MIT algorithm.

1.1.5 MODEL REFERENCE ADAPTIVE CONTROL (MRAC)

Model Reference Adaptive Control (MRAC) has been developed and implemented to control a non-linear system as shown in figure 1.3. The idea of the MRAC is based on forcing the plant to follow the reference model, i.e. the adaptive controller has to decrease the error vector between the reference model and plant to zero. This method of MRAC has been implemented in the feedback loop to improve the performance of the process. The MRAC has an ordinary feedback loop composed of the process and the controller and another feedback loop that changes the controller parameters. The parameters are changed on the basis of feedback from the error, which is the difference between the output of the system and the output of the reference model.
1.1.5.1 COMPONENTS OF MODEL REFERENCE ADAPTIVE CONTROLLER

(i) Reference Model

It is used to specify the ideal response of the adaptive control system to external command. It should reflect the performance specifications in control tasks. The ideal behavior specified by the reference model should be achievable for the adaptive control system.

(ii) Controller

It is usually parameterized by a number of adjustable parameters. In this paper two parameters are used to define the controller law. The control law is linear in terms of the adjustable parameters (linear parameterization). Adaptive controller design normally requires linear parameterization in order to obtain adaptation mechanism with guaranteed stability and tracking convergence.

(iii) Adaptive Mechanism

It is used to adjust the parameters in the control law. Adaptation law searches for the parameters such that the response of the plant which should be same as the reference model. It is designed to guarantee the stability of the control system as well as convergence of tracking error to zero.

1.1.6 HYBRID NEURO FUZZY CONTROLLER MODEL
Recently hybrid neuro-fuzzy HNF approach has gained considerable attention in fields like control, pattern recognition, process industry, image processing, etc.

The FLC can be a better choice when accurate mathematical formulations are not possible. Other merits of FLC include but not limited to are

(i) It can operate with less precise inputs.

(ii) It needs less information storage in the form of membership functions and rules than conventional look up table for non-linear controllers.

(iii) It is more robust than other non-linear controllers.

The hybrid neuro-fuzzy system includes

(i) Fuzzy Controller

(ii) Simulink model with FLC

(iii) Train the collected data with generated FIS upto a particular no. of Epochs.

The focus of this work is to design an estimator and controller using Fuzzy and neural network for continuous/discrete-time nonlinear control systems which guarantees desired transient performances in the presence of plant parameter variations and unknown external disturbances. The proposed design methodology allows providing effective control of nonlinear systems in the presence of uncertainty. A distinctive feature of the designed control system is that, two-time-scale motions are artificially forced in the closed-loop system. The advantage of the discussed schemes is that controller with additional low pass filtering can be found for nonlinear systems, where controller parameters depend explicitly on the specifications of the desired output behavior.

1.2 PROBLEM STATEMENT AND PROPOSED SOLUTION

Nonlinear control systems require efficient regulatory structures of various kinds to enhance their quality of operation. System such as adaptive PI controllers and fuzzy logic controllers were reported to be effective in specific and well-defined environments. The main requirements for these systems is that they required regulatory (synonym) generated by certain functional blocks.

The effectiveness of the system is demonstrated effectively for systems with high degree of instability. This constrain is represented by a system which does entail feedback parameters through endorsing regulative algorithms to provide control input for the same. The restoration function can also be accomplished in different ways such as, augmenting the capabilities of remaining system
paths, detouring around the points of damage and providing system with whole new paths for feedback. The benefit of this system is that the percent of false trigger is low and the necessary system requirements are inexpensive.

The specific features associated with the sensor signal (reflecting the variations) is extracted and mapped into another set of control signals that can be eventually used to control devices. Unfortunately, Artifacts (Undesirable Potentials of non-cerebral origin) can modify the value of a signal and result in an unintentional control of the device. Therefore there is a need to avoid reject or remove artifacts from the instantaneously acquired signal. Linear filtering is useful for removing only for those artifacts located in certain frequency bands that do not overlap with the region of interest and this method fails when the signals and artifacts overlap or lie in the same frequency band.

Alternatively, using a linear combination of the constant gain PI's is the most common technique to offer control signal from output. The linear combination technique provides expected output with reduced bias. The problem lies with the estimation of bias value and conventionally least square criterion is used to estimate the value of bias. The disadvantages associated with this scheme is whether the value of bias should be calculated separately for each type of output and for the different scenarios. The disadvantages increase for slowly varying command inputs. As they have no reference values.

ICA is a method that blindly separates mixtures of independent source signals, forcing the components to be independent. Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non gaussian and mutually independent and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA. ICA is superficially related to principal component analysis and factor analysis. ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail completely.

The data analyzed by ICA could originate from many different kinds of application fields, including digital images, document databases, economic indicators and psychometric measurements. In many cases, the measurements are given as a set of parallel signals or time series; the term blind source separation is used to characterize this problem. Typical examples are mixtures of simultaneous
speech signals that have been picked up by several microphones, brain waves recorded by multiple sensors, interfering radio signals arriving at a mobile phone, or parallel time series obtained from some industrial process.

Independent component analysis (ICA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent only if the data set is jointly normally distributed. ICA is sensitive to the relative scaling of the original variables.

ICA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data. If a multivariate dataset is visualised as a set of coordinates in a high-dimensional data space (1 axis per variable)

In the area of process engineering, process design and simulation, process supervision, control and estimation, and process fault detection and diagnosis rely on the effective processing of unpredictable and imprecise information. In such situations, the neural network, which can achieve the sophisticated level of information processing the brain is capable of, can excel. The neural networks are generally viewed as process modeling formalism and given the appropriate network topology; they are capable of characterizing nonlinear functional relationships. Furthermore, the structure of the resulting neural network based process model may be considered generic, in the sense that little prior process knowledge is required in its determination. The knowledge about the plant dynamics and mapping characteristics is implicitly stored within the network.

Many control applications require cooperation of two or more independently designed separately located, but mutually affecting, subsystems. In addition to good behavior of each of the subsystem, effective coordination of these subsystems is very important to achieve the desired overall system performance. Such coordination can permit the use of commercially designed subsystems to perform more sophisticated tasks than originally intended, and hence improve the operational reliability of a plant. However, such a co-ordination is very difficult to accomplish mainly due to the
lack of availability of precise system models and/or dynamic parameters. A new multiple-sensor coordinator based sensor validation scheme that combines the techniques of intelligent control and neural networks is presented in this thesis. The basic idea is to improve the operational reliability of a plant by integrating the data from multiple sensors, (which are commonly present in most industrial processes for monitoring various states and environmental conditions) and using it for estimating the principal output. Since intelligent control does not depend only on mathematical analysis and manipulation, it is an attractive candidate to deal with complex control system problems.

Conventional PI and PID controllers are popular partially due to their functional simplicity that allows process engineers to operate them in a simple and straightforward manner. To implement such a controller, three parameters must be determined from the mathematical model of the process, but the drawback is, that the mathematical model of the system can change due to a shift in the operating point and sudden changes in environment. In these situations, the controllers have to be retuned on-line and manual retuning is tedious. The parameter-adaptive structure of intelligent controller overcomes this drawback. The performance-adaptive structure is motivated by human expert control and/or human cognition ability and attempts to control a system directly with the intelligent controller.

In this work an algorithm that improves the system performance and help in extending the range of operation of the conventional control algorithms with respect to sensor validation is developed. The main idea behind this approach is to monitor and approximate any off-nominal behavior in the dynamical system by using on-line approximation structures (e.g., neural networks). The sensor validation algorithms are important, since the designs of adaptive tuning methods are based on multiple measurements. If a measurement does not truly represent the process conditions due to a sensor failure, the resulting tuning constants could be far from the proper values, leading to poor or even unstable performance.

In a multiple sensor coordination network formed with the help of two or more separately located, independently designed sensors, it is desirable for each of the sensors to have identical characteristics with respect to noise rejection capability. This rigorous constraint cannot be satisfied in actual practice, and hence, it becomes necessary, to use noise elimination algorithm and remove the noise present in the individual sensors. In the present work, an Independent Component Analysis (ICA) network is
incorporated into the existing system to extract the true value of the measured states from noisy sensor readings.

The algorithms developed in this work, are tested directly on a nonlinear process, since practical systems are not precisely linear and even when represented as linearised models around a nominal operating point, the real system characteristics need not be reflected due to variations in process parameters. A nonlinear (level) process is considered for simulation and real time implementation. In the process considered, sensors are used for measuring inflow, outflow and process variable level. A shift in the operating point from the bottom to the top of the tank will alter the time constant and static gain drastically. Thus the tank process using multiple sensors with vast amount of highly correlated data, process dependent time constant and gain becomes suitable for the present work.

1.3 FOCUS OF RESEARCH

The reported limitations in noise removal schemes and control units are addressed in this research through a method referred to as ‘Statistical correlation Selection method’.

The overall objective of this research is (i) to develop a system that is resistant or tolerant of sensor failures (ii) describes the use of an ICA based noise cancellation filter to detect a sensor failure by using the data from plant sensors. Once a failure in a sensor is detected, the processing hardware will be “re-designed” based on evolvable hardware principles to take the average of the remaining sensors. The system thus continues to operate with a reduced number of sensors. The filter estimates the process value by using a form of feedback control.

1.4 BENEFITS OF THE RESEARCH WORK

(i) To study the existing artifact removal schemes and their limitations
(ii) Implement an efficient noise removal scheme, tested against sensor noise

1.5 GOALS OF THE RESEARCH WORK

The goals of this research include, proposing, implementing a fault tolerant control system and validating the sensor noise cancellation network

(i) In non changing environment with low and high noise
1.6 THESIS CONTENTS

Chapter 1 summarize the introduction, design of an estimator and control schemes for nonlinear systems with guaranteed desired transient performance in the presence of plant parameter variations and unknown external disturbances.

Chapter 2 summarize literature review for constant gain PI controllers, adaptive PI controllers, fuzzy logic controllers, intelligent controllers with fuzzy logic for state estimation and intelligent controllers with neural network for state estimation.

Chapter 3 provides the description of the real time plant with derived transfer function of the nonlinear process, representation of stable and unstable systems, analog controllers, neural networks and their control structures, basic neural learning control and layered feed forward neural network.

Chapter 4 summarizes the fusion of ideas from fuzzy control and neural networks. Fuzzy logic has proven effective for complex, nonlinear and imprecisely defined systems. The common bottleneck in fuzzy logic is the derivation of fuzzy rules and the parameter tuning for the controller. The neural networks have powerful learning abilities, optimization abilities and adaptation. The fuzzy logic and neural networks can be integrated to form a connectionist adaptive network based. This integrated adaptive system can modify the characteristics of the rules, topology of fuzzy sets and/or the structure of control system.

Chapter 5 narrates a comprehensive training data for neural work for assessing response of the proposed neural estimator. This chapter proposes performance analysis scheme for the estimator to measure step response of the estimator, estimator response to 0.5Hz square wave, estimator response to 0.05Hz square wave and response to low frequency sine wave.

Chapter 6 presents the response to set point variations, variations in load variable outflow along with online acquired plots, experimental results and comparison study for both slow and fast motion response.

Chapter 7 describes intelligent controller with sensor noise elimination network using independent component analysis suited for a multiple sensor coordination model.

Chapter 8 summarizes the conclusion and future scope.
1.7 OBJECTIVES OF THE PRESENT WORK

(i) To design a parameter-adaptive intelligent controller using online tuning techniques and performance-adaptive fuzzy controller for a nonlinear multivariable (level and flow) process and compare the performance of the two controllers by simulation and experimental study.

(ii) To process the plant signals in real time and combine with previous monitoring data to design an on-line estimator to detect and accommodate faults in nonlinear dynamical systems. In the presence of a sensor failure, the function represented by the on-line approximate is used as an estimate, thereby providing a natural framework for fault identification and accommodation. In this work, two types of estimator models to estimate the process variable level is proposed by the author, one using fuzzy logic and the other using neural network.
CHAPTER 2

LITERATURE REVIEW

2.1 LIMITATIONS OF CONVENTIONAL CONSTANT GAIN PI CONTROLLERS

The limitations of PI controller for a nonlinear process designed using some of the well known linearisation techniques was discussed by Rugh (1987). He showed that the best performance of such type of controllers is guaranteed only for slowly varying command inputs.

It was reported by Lee, et al (1990) that for processes approximated as a first order plus dead time model (FOPDT), the PI or PID controllers designed using conventional tuning methods gave poor performance in some cases. He also suggested that the system performance in such situations shall be improved by using a second order plus dead time model instead of the FOPDT model.

The limitations of the ZN-PID controllers were reported by Shen and Yu (1994). They presented that even with proper tuning, the system with a ZN-PID controller can become unstable for underdamped systems. It was concluded that the system response with PID controllers is generally poor for an integrating process. In such cases, Lee and Sung, (1996) proposed that two independent PID controllers shall be used; one for set point tracking and the other for disturbance rejection. However, the interaction effects and other nonlinearities were not considered.

Nikolaou and Misra (2001) reported that most of the chemical processes are nonlinear, and for several of them, linear feedback control is adequate. They concluded that techniques are required to quantify the nonlinearity of a process to assess whether conventional constant gain controllers alone are sufficient.


Krishnaswamy and Rangiah (1994) A simple method for determination of second order plus dead time model parameters from process transients has been presented. The estimation is based entirely on the reaction curve
Lee and Sung (1996) brought out the limitations of the PID controllers. The transfer function models have been alone used.


Wright and Kravaris (2003) the authors have focused on synthesizing nonlinear decoupling controllers for multivariable nonlinear systems represented by a state-space model, in presence of dead times associated with the sensors and actuators. Simulation study alone has been done.

KokKiong Tan et al (2006) presented an iterative learning control (ILC) approach towards closed loop automatic tuning of PID controllers. The modification is not tested on a nonlinear system.

Wen Tan et al (2006) have compared several well known PID tuning formulas and observed that the robustness measure should lie between 3 to 5 to have a good compromise between performance and robustness. Z-N method has the best compromise between robustness and performance for the process considered

Padhy et al (2006) brought out the relay based PI-PD design for stable and unstable first order plus time delay (FODPT) processes. Only simulation results are available.

Wright and Kravaris (2003) the authors have focused on synthesizing nonlinear decoupling controllers for multivariable nonlinear systems represented by a state-space model, in presence of dead times associated with the sensors and actuators. Simulation study alone has been done.

Da Zhang (2006) concluded when the output of a proportional-integral (PI) controller is saturated, performance degradation or even instability occurs. Authors proposed three new anti-windup algorithms for a digital PI-speed controller to improve the control performance of variable-speed motor drives. These designs are implemented in a Field Programmable Gate Array (FPGA) device and stochastic theory is employed to enhance the computational capability of FPGA. The proposed scheme delivers large dynamic range, easy digital design, minimal scaling of digital circuits, reconfigurability, and direct hardware implementation, while maintaining high control performance. The developed controllers are applied to the speed control of a field-oriented controlled induction motor drive using a hardware-in-the-loop test bench.

Wang, B. (2006) discussed multible integral processes with dead time controlled by a PI controller. Author implemented a normalised system that involves only two free parameters. The
proposed scheme characterized control parameters region to ensure system stability and determines gain and/or phase margin [GPM].

Lennartson, B. (2009) presented an analytical PID design method based on lambda turning approach. This general controller evaluation scheme taking both performance and robustness in different frequency regions into account. The analytical method introduced includes two tuning parameters, one that guarantees a specified stability margin for the given model, and one that is also able to adjust the control activity to a desired level. The suggested method, called robust internal control based model (IMC) for a specific second-order non-minimum phase plant model to control both mid and high-frequency robustness. extended evaluation procedure also illustrates how efficiently PI and PID controllers including a Smith predictor (SP) can control time delayed plants. More specifically, it is shown to be more profitable to provide a PI controller with derivative action than with a SP for plants with long time delays.

Lin-linOu (2009) considered the problem of stabilizing a single-input-single-output (SISO) linear time-invariant (LTI) plant with known time delay using a low-order controller, such as a Proportional (P), a Proportional-Integral (PI), or a proportional-integral-derivative (PID) controller. Necessary and sufficient conditions for the stability of LTI systems with time delay were presented. By employing an extended Hermite-Biehler Theorem applicable to quasi-polynomials. Author proposed analytical algorithms to compute the stabilizing sets of P, PI and PID controllers. The resulting characterizations of the stabilizing sets for P, PI and PID controllers are analogous to the Youla parameterization of all stabilizing controllers for plants without time delay.

Xinyu Du (2010)Research results on type-2 (T2) fuzzy control that has started to emerge in the literature over the past several years. None of these results, however, are concerned with the explicit input-output mathematical structure of a T2 fuzzy controller. As the literature on type-1 (T1) fuzzy control has demonstrated, revealing such structure information is important as it will deepen the precise understanding of how and why T2 fuzzy controllers function in the context of control theory and lay a foundation for more rigorous system analysis and design. In this paper, the mathematical structure of two Mamdani interval T2 fuzzy-proportional-integral (PI) controllers that use the following identical elements are derived: two interval T2 triangular input fuzzy sets for each of the two input variables, four singleton T1 output fuzzy sets, a Zadeh and operator, and the center-of-sets type reducer. Onecontroller employs the popular centroid defuzzifier, while the other employs a new defuzzifier that we propose, which is called the average defuzzifier. The advantages of using the latter defuzzifier are given, which include the fact that the derivation method originally developed by us in
previous papers for the T1 fuzzy controllers can be directly adopted for the T2 controller, and the results are general with respect to the design parameters. This is not the case for the other T2 controller, for which we have developed a novel derivation approach partially depending on numerical computations. The results prove explicitly both controllers to be nonlinear PI controllers with variable gains (i.e., the expressions are different). The gain-variation characteristics are analyzed and extend the findings are extended to the corresponding T2 fuzzy-proportional-derivative (PD) controllers. The results are consistent with the relevant structure results on the T1 fuzzy-PI and PD controllers in the literature and contain them as special cases.

Chien-Hung Liu (2010) a self-tuning proportional-integral (PI) controller in which the controller gains are adapted using the particle swarm optimization (PSO) technique is proposed for a static synchronous compensator (STATCOM). An efficient formula for the estimation of system load impedance using real-time measurements is derived. To demonstrate the effectiveness of the proposed PSO self-tuning PI controller for a STATCOM, experimental results for a system under different loading conditions are presented. Results from the self-tuning PI controller are compared with those from the fixed-gain PI controllers.

Teo J. (2010) Show that stabilizing tracking proportional-integral (PI) controllers can be constructed for minimum-phase nonaffine-in-control systems. The constructed PI controller is an equivalent realization of an approximate dynamic inversion controller. This equivalence holds only for the time response when applied to the unperturbed system. Even when restricted to unperturbed minimum-phase linear time invariant systems, their closed loop robustness properties differ. This shows that in general, properties that do not define the equivalence relation for systems/controllers are not preserved under such equivalence transformations.

2.2 ADAPTIVE PI CONTROLLERS

An on-line tuned PI controller for a nonlinear level process, by considering a conical tank, was presented by Astrom, et al, (1995). The controller was tested on transfer function model only. The system response was much better with the adaptive PI controller compared to the conventional constant gain ZN-PI controller. The controller was designed by choosing the PI controller parameters in terms of the nonlinear process
characteristics, the process gain and the process time constant. However, no experimental study was carried out. The effect of dead time was not considered in the study. Also, in some cases, it may not be as easy as reported in the work, to determine the controller parameters as a function of the measured variables.

An adaptive PI controller was designed for a bio-methanization process by Bastin G., et al, (1983). The closed loop dynamics of the system were found to be acceptable for both set point tracking and disturbance rejection. However, the design was tested on transfer function models only, without considering the effect of modeling uncertainties and noise.

Efficient control strategies for a nonlinear pH process using the velocity-form of linearisation were reported by Nystrom, et al (2002). It was shown, that the theory of correctly choosing the adaptive controller parameters for a nonlinear process, still remains at a stage of infancy, and trial and error method remains to be the most effective design approach till date.

J.K.Harvey (1993) presented a self-tuning controller for a wheel torque regulation. The effect of noise is not explicitly brought out.

Bamieh and Giarre (1999) Considered least square type identification problems for LPV systems with polynomial dependence on the parameters. Once the LPV model based on the Moore and Greitzer nonlinear model of compressors is identified, the design of robust gain scheduling predictive controller can be implemented. Simulation study alone has been done.

Nystrom et al (2002) Presented that the theory of ‘correct’ gain scheduling still remains at a stage of infancy. Concludes that gain-scheduling parameter should be slowly varying. Simulation study. The problem is still not rigorously solved and trial and error remains the most effective design approach.


Kerrigan E.C., et al (2000) Presented a fault tolerant control of a ship propulsion system using model predictive control. The ability of MPC to handle higher level control objectives during failures such as speed control and efficiency optimization is not done.

Kristiansson, B. (2006) this article presented some easily understood and applied methods for close-to-optimal tuning PI and PID controllers. By optimal it means good midfrequency robustness and the best possible tradeoff between output performance and control activity. For plants with all
poles on the negative real axis, a simple step response can provide adequate plant knowledge. For plants with integral action, an impulse response or a relay experiment can be used. In all PID cases, the controller zeros can be fixed, the control activity can be varied by the filter factor, and, finally, the integral gain can be adjusted to the required damping of a step response for the closed-loop system. With this strategy, tuning a PID controller is as easy as tuning a PI controller, the difference being that the PID solution gives additional freedom for selecting slightly higher control activity which significantly improves the output performance. When the situation demands HF rolloff of the controller, the PI or PID controller can be augmented by an additional lowpass filter. In these cases, the inclusion of derivative action is recommended. Four of the five controller parameters are then easily found, and the remaining gain can be manually tuned to obtain a desired tradeoff between output performance and damping (MF robustness).

Jong-Woo Choi (2009) proposed a new scheme for handling integrator windup. A new antiwindup strategy for PI speed controller was suggested to handle large set-point changes. When the speed control mode is changed from P control to PI control, an appropriate initial value for the integrator is assigned. This value then restricts the overshoot. In addition, the proposed method guarantees the designed performance independent of the operating conditions, i.e., different set-point changes and load torques, and can be easily implemented with existing PI controllers. In SIMULINK/MATLAB-based comparative simulations and experiments for a permanent-magnet synchronous motor speed controller, the proposed method shows a superior control performance compared with the existing well-known antiwindup methods, such as conditional integration and tracking back calculation.

Luo. Y. (2011). In this study, a fractional order (PI)$^\lambda$ controller is developed and implemented to improve the flight control performance and robustness of a small fixed-wing unmanned aerial vehicle (UAV). The decoupled roll-channel control is realised under certain conditions and tested using the designed controllers in this study. The inner closed-loop system of the roll-channel is approximately identified as a first-order plus time delay model using the flight test data. For comparison purpose, an integer-order PI controller is designed following the modified Ziegler-Nichols (MZNs) tuning rule, based on this identified roll-channel control model. According to three design pre-specifications, the integer-order proportional integral derivative (PID), fractional-order PI$^\lambda$ and (PI)$^\lambda$ controllers are designed for the roll-channel flight control system of a small fixed-wing UAV. These three designed controllers share the same gain crossover frequency and phase
margin settings for fair comparisons. From both simulation and real flight experiments, the two designed fractional order controllers outperform the MZNs PI and the designed integer-order PID controllers. The designed (PI)$^\lambda$ controller achieve even better performance than the designed PI$^\lambda$ controller.

Almeida, G.M. (2011) Adaptive multiprocessor systems are appearing as a promising solution for dealing with complex and unpredictable scenarios. Given the large variety of possible use cases that these platforms must support and the resulting workload variability, offline approaches are no longer sufficient because they do not allow coping with time changing workloads. This presented a novel approach based on the utilization of PI and PID controllers, widely used in control automation, for optimizing resources utilization in Multiprocessor System-on- frequency Chip (MPSoC). Several architecture characteristics such as response time during changing, noise and perturbations are modeled and validated in a high-level model and results are compared to information obtained on a homogeneous MPSoC platform prototype. Power and energy consumption figures are discussed and two controllers are proposed: 1) PI and 2) PID-based controllers. Results show the system capability in adapting to disturbing conditions while ensuring application performance constraints and reducing energy consumption.

Hensel, B. (2012) Event-based sampling allows saving energy in the sensor transmitter by avoiding unnecessary messages. One important application is room temperature control with wireless sensors. Optimizing the controller parameters of a PI controller for this application is a difficult task, because usually no process model is available and challenging issues like actuator saturation have to be taken into account. Adaptive controllers offer the possibility to tune themselves automatically. An adaptive PI controller based on pattern recognition is proposed, designed for room temperature control, sensor energy efficiency, and level-crossing sampling. The implementation is much easier than that of most other adaptive controllers and robustness to disturbances and noise is high. The focus of this paper lies rather on the basic idea, simulations and practical issues than on theoretical investigations.

2.3 FUZZY LOGIC CONTROLLERS (FLC)

Several of the un-modeled disturbances and nonlinearities present in a process can be taken care by the FLC. Processes with dead time can also be controlled by using FLC with minimum number of rules in the rule base,
and this was demonstrated by Aoki, et al, (1990) by designing a simple FLC for controlling the temperature in a glass melting furnace. Fermentation process with significant measurement delays were compensated by using a FLC, to control the product concentration in a batch-fed fermentor by Chidambaran and Nyttle (1994). However, experimental study was not carried out.

Fuzzy tuned PI and PID controllers have been reported for control of nonlinear processes (Zhao, et al, 1993). The operational simplicity of PI or PID controllers can be retained, and the performance can be improved by using fuzzy tuning techniques. An integrated fuzzy predictive control was also reported in the control of complex chemical processes (Kim, et al, 1995).

Tuning of fuzzy controllers using heuristic algorithms such as Genetic Algorithms for minimum ISE criterion were used by Homifair, et al (1996). Tuned controllers were found to exhibit good set point tracking and disturbance rejection capability. The limitation observed was that the tuning techniques are offline and can be time consuming when used online.

Chidambaran and Nyttle (1994) Developed a rule based fuzzy logic controller to compensate for significant measurement delays in the control of product concentration in a batch-fed fermentor. Prediction of future variable is based on derivative of present variable and present change in input variable.

Kim and Kim (1995) A combined fuzzy and predictive control was proposed for a multi input- multi output system (MIMO). The fuzzy control mentioned has not handled the interaction effects in the considered MIMO system.

Zhao et al (2000) The feature patterns of a typical chemical process with, large dead time have been analyzed and a pattern based fuzzy predictor (PFP) has been developed. Significant nonlinearity and variations of the dead time and the stability of the control system after incorporating the PFP scheme have not been addressed.

Osofisan et al (2007) Proposed a simple fuzzy logic controller for controlling the fluidized catalytic cracking unit in petrochemical industry and a well defined relationship between the vital variables through fuzzy logic control scheme is also determined. The fuzzy model design is capable of managing the characteristic uncertainties and imprecision normally associated with catalytic cracking process in FCCU.


Wen Li (2007) An approach is proposed for vibration suppression in a two-inertia system using an integration of a fractional-order disturbance observer and a single neuron-based PI fuzzy controller. The former is used to obtain disturbance estimate and generate compensation signal, and the latter is utilized to realize outer loop control. Fractional-order disturbance observer has a wider range to select a suitable tradeoff between robustness and vibration suppression, because introduction of fractional calculus makes universe of relative degree of Q-filter is expanded from integer domain to real-number domain. For the single neuron-based PI fuzzy controller, a single neuron makes up a PI controller and such a controller is embedded in each cell of the fuzzy control table. Thus, the fuzzy control table is changed into a controller matrix and it constructs a nonlinear adaptive controller with parameter self-tuning property. Experimental results illustrate that the integration of fractional-order disturbance observer and single neuron-based PI fuzzy controller can improve the performance of disturbance attenuation and system robustness.

Shahnazi, R. (2008) A position control of a class of servomotors is addressed in this paper via a novel adaptive fuzzy PI sliding mode control. The premise and the consequence parts of the fuzzy rules are tuned with adaptive schemes. To attenuate chattering effectively, the discontinuous control is approximated by an adaptive PI control structure. Moreover, the bound of the discontinuous control term is assumed to be unknown, and an adaptive mechanism is used to estimate this bound. All adaptive laws are derived via Lyapunov synthesis method, thereby guaranteeing the closed-loop stability. The proposed approach has the added advantage that, for external disturbances, it only requires a bound to exist, without needing to know the magnitude of this bound. The
proposed controller is applied to control a model of uncertain induction servomotor subject to significant disturbances and a model of DC servomotor with unknown parameters and uncertainty in load condition. The analysis of simulations reveals the effectiveness of the proposed method in controlling servomotors in terms of significant reduction in chattering while maintaining asymptotic convergence.

Shahnazi, R.(2008) Design of controllers for uncertain systems is inherently paradoxical. Adaptive control approaches claim to adapt system parameters against uncertainties, but only if these uncertainties change slowly enough. Alternatively, robust control methodologies claim to ensure system stability against uncertainties, but only if these uncertainties remain within known bounds. This is while, in reality, disturbances and uncertainties remain faithfully uncertain, i.e., may be both fast and large. In this paper, a PI-adaptive fuzzy control architecture for a class of uncertain nonlinear systems is proposed that aims to provide added robustness in the presence of large and fast but bounded uncertainties and disturbances. While the proposed approach requires the uncertainties to be bounded, it does not require this bound to be known. Lyapunov analysis is used to prove asymptotic stability of the proposed approach. Application of the proposed method to a second-order inverted pendulum system demonstrates the effectiveness of the proposed approach. Specifically, system responses to fast versus slow and large versus small disturbances are considered in the presented simulation studies.

Wahyunggoro.O.(2008) Direct Current (DC) servomotors are widely used in robot manipulator applications. Servomotors use feedback controller to control either the speed or the position or both, and the basic continuous feedback control is PID controller. This discusses the modelling and simulation of DC servomotor control built using MATLAB/Simulink, and the analysis of controller performance, namely a fuzzy-scheduled PID controller and a fuzzy-logic-based self tuning PI controller with a variation of scaling factor. The singleton fuzzification is used as a fuzzifier: three membership functions for both input and output of fuzzy logic controller. The center average is used as a defuzzifier. Two control modes are applied in sequential to the plant: speed control, and then position control. Simulation results show that fuzzy-logic-based self tuning PI controller has better performance in terms of percent OS and settling time compared to fuzzy-scheduled PID controller for the speed control of DC servomotor.

Nagarajan.R.(2008) Presented a simple adaptive predictive fuzzy logic controller is developed for the attitude control of a micro-satellite; its performance is compared with a PID controller. PID controller is the most widely used among the conventional controllers. The gain of proportional,
integral and derivative control has to be tuned and fixed throughout the control simulation. APFLC is introduced in order to reduce the effect of unpredictable time delays and large uncertainties. The design schemes of modeling APFLC has the following: basic fuzzy logic controller, predictive FLC and adaptive predictive FLC. All the proposed models have been analyzed with the same level of noise and external disturbances. From the simulation results, it is observed that the performance of the proposed APFLC has an edge over the conventional PID controller.

Al-Zubi, A.A. (2010) presented paper deals with the problems of fuzzy controllers design. The controllers are components of control systems for various controlled objects. The analysis of design methods influencing efficiency of solving various compromise problems related to minimization of hardware complicity of fuzzy digital controllers and ensuring the desired quality coefficients of the control systems is given. The various approaches for increasing efficiency of fuzzy controllers design processes based on their structural-parametric optimization are discussed.

Jung, J.-W. (2011) This study proposed a robust fuzzy PI-type current controller for a permanent magnet synchronous motor (PMSM). The proposed current control law consists of two control terms: a decoupling control term and a fuzzy PI control term. The decoupling controller accounts for the non-linearity of a PMSM model, and the fuzzy PI controller stabilises the decoupled dynamics. Based on Kharitonov's theorem, the stability condition, which can guarantee the asymptotic stability of the closed-loop system, is derived. Simulation and experimental results are presented to demonstrate the feasibility of the proposed control scheme using a prototype 1 HP PMSM servo system.

Shun-Chung Wang (2011) Based on the redevelopment of control rule base, two modified PI-like fuzzy logic controllers with output scaling factor (SF) self-tuning mechanism are proposed and verified in this paper with application to switched reluctance motor (SRM) drive system. The motivation of this paper is to simplify the program complexity of the controller by reducing the number of fuzzy sets of the membership functions (MFs) without losing the system performance and stability via the adjustable controller gain. For both types of controllers, the output SF of the controller can be tuned continuously by a gain updating factor, whose value is derived from the fuzzy logic reasoning, with the plant error and the error change ratio as the input variables. The rule bases are created based on the knowledge of the SRM's dynamic behavior and practical experience. Various aspects of the design considerations about the MF, rule base, and gain tuning strategy are described in detail. Experimental results, carried out on a four-phase 8/6-pole SRM based on the dSPACE DS1104 platform, are given to show the feasibility and effectiveness of the devised
methods. A performance comparison of the proposed controllers with their conventional counterpart is also included.

Jiuxiang Dong (2010) Considered the output feedback control problem for nonlinear discrete-time systems, which are represented by a type of fuzzy systems with local nonlinear models. By using the estimations of the states and nonlinear functions in local models, sufficient conditions for designing observer-based controllers are given for discrete-time nonlinear systems. First, a separation property, i.e., the controller and the observer can be independently designed, is proved for the class of fuzzy systems. Second, a two-step procedure with cone complimentary linearization algorithms is also developed for solving the \( H_\infty \) dynamic output feedback (DOF) control problem. Moreover, for the case where the nonlinear functions in local sub models are measurable, a convex condition for designing \( H_\infty \) controllers is given by a new DOF control scheme. In contrast to the existing methods, the new methods can design output feedback controllers with fewer fuzzy rules as well as less computational burden, which is helpful for controller designs and implementations. Lastly, numerical examples are given to illustrate the effectiveness of the proposed methods.

Chairez, I. (2013) In general, output-based controller design remains an important research area in control theory. Most of the existing solutions use a state estimation algorithm to reconstruct a plausible approximation of the real state. Then, one can apply a nonlinear controller, based on fuzzy logic, for example, to enforce the system trajectories to a desirable stable equilibrium point. Nevertheless, the aforementioned method may not be suitable for uncertain systems affected by external noises. State observers based on the system's structure cannot be applied in those cases. However, some sort of adaptive estimation may be developed. This paper deals with a fuzzy controller that was designed using the state observer solution when the dynamic model of a plant contains uncertainties or it is partially unknown. Differential neural network (DNN) approach is applied in this uninformative situation. A new learning law, containing an adaptive adjustment rate, is suggested to enforce the stability condition for the observer's free parameters. On the other hand, nominal weights are adjusted during the preliminary training process using the least mean square method.

Huo, B. (2012) This study developed an adaptive fuzzy control method for accommodating actuator faults in a class of uncertain multi-input and multi-output non-linear systems in strict-feedback form and without the requirement of their states being available for controller design. The considered faults are modelled as both loss-of-effectiveness and lock-in-place (stuck at unknown place). With the help of fuzzy logic systems to approximate the unknown non-linear functions,
a fuzzy adaptive observer is developed for estimating the unmeasured states. Combining backstepping technique with non-linear tolerant-fault control theory, a novel adaptive fuzzy fault-tolerant control approach is constructed. It is proved that the proposed control approach can guarantee that all the signals of the resulting closed-loop system are bounded, and also the tracking errors between the system outputs and the reference signals converge to a small neighbourhood of zero by appropriate choice of the design parameters. Simulation results are provided to show the effectiveness of the control approach.

Chang-WooPark (2004) A parameter estimation scheme with an appropriate adaptive law for updating the parameters is designed and analyzed based on the Lyapunov theory for the general MIMO Takagi-Sugeno (T-S) fuzzy models. The parameters of the Takagi-Sugeno fuzzy models can be estimated by observing the behaviour of the system and with the online parameter estimator, any type of fuzzy controllers works adaptively to the parameter perturbation. In order to show the applicability of the proposed estimator, an existing fuzzy state feedback controller is adopted and indirect adaptive fuzzy control design with the proposed estimator is shown. From the numerical simulations and experiments, it is shown that the derived adaptive law works for the estimation model to follow the parameterized plant model and the overall control system has robustness to the parameter perturbation.

Huo, B., Li, Y., Tong, S. (2012) This study developed an adaptive fuzzy control method for accommodating actuator faults in a class of uncertain multi-input and multi-output non-linear systems in strict-feedback form and without the requirement of their states being available for controller design. The considered faults are modelled as both loss-of-effectiveness and lock-in-place (stuck at unknown place). With the help of fuzzy logic systems to approximate the unknown non-linear functions, a fuzzy adaptive observer is developed for estimating the unmeasured states. Combining backstepping technique with non-linear tolerant-fault control theory, a novel adaptive fuzzy fault-tolerant control approach is constructed. It is proved that the proposed control approach can guarantee that all the signals of the resulting closed-loop system are bounded, and also the tracking errors between the system outputs and the reference signals converge to a small neighbourhood of zero by appropriate choice of the design parameters. Simulation results are provided to show the effectiveness of the control approach.

2.4 INTELLIGENT CONTROLLERS WITH FUZZY LOGIC FOR STATE ESTIMATION
The motivation for the use of fuzzy logic for modeling and state estimation was given by L.X.Wang, et al, (1992), by showing that fuzzy systems are universal function approximators. He presented a detailed review on the above theory.

A fuzzy model reference learning control was implemented by J.R.Layne and K.M.Passino (1993), for a cargo ship steering problem. However, the controller was not tested on nonlinear process.

The application of fuzzy based self-learning controllers to nonlinear process was reported by Jyh-Shing and R.Jang (1992). Simulation results obtained from the classical inverted pendulum model was also presented. However, the reported work has no experimentation.

Some of the techniques, using which fuzzy systems can be trained for use as estimators, were presented by E.G.Laukonen, et al, (1993). The use of fuzzy logic to estimate the parameter, altitude, in the control of a spacecraft was demonstrated by S.Daley and K.F.Gill (1987). Some of the robustness issues, such as considering the effects of noise, etc. in the design of estimators, were presented by Patton, R.J., et al (1993).

R.Tanscheit, et al (1988) Presents a rule-based self-organizing controller for robotics applications. Pointed out that, if the process has poles near the origin in the left plane, then the responses may be sluggish enough to be unacceptable. Consequently, disturbance attenuation, which is one of the major problems of process control, was not discussed. Works well for the case of model mismatch being very small.

S.R.Morris (1994) Investigated the use of fuzzy estimators for failure detection and identification. He concluded that the estimator is consistent if the differences between the plant and the model are sufficiently small. Experimental results not presented. Noise rejection capability of the estimator not demonstrated.


LoftiA.Zadeh (1973) The founder of fuzzy logic. He has outlined the new approaches to the analysis of complex systems and decision processes.

S.Daley and K.F.Gill (1987) Presented the design of a self-organizing fuzzy logic controller to estimate the parameter altitude in a spacecraft control. The transfer function model has alone been used. Experimental study with noise effects has not been considered.
S.Abe and M.S.Lan (1995) Presented a study on extracting fuzzy rules directly from numerical data for functional approximation. Useful in deriving rules for applications where rich input-output data is available. Not much suited for ill-defined models.


Prakash J. et al,(2007) Presented the design of a Fuzzy observer based nonlinear model predictive controller for continuous stirred tank reactor. The fuzzy control mentioned has not handled the interaction effects in the considered system.


Perry. A.G. (2007) proposed a novel design procedure of proportional and integral (PI)-like fuzzy logic controller (FLC) for DC-DC converters that integrates linear control techniques with fuzzy logic. The design procedure allows the small signal model of the converter and linear control design techniques to be used in the initial stages of FLC design. This simplifies the small signal design and the stability assessment of the FLC. By exploiting the fuzzy logic structure of the controller, heuristic knowledge is incorporated in the design, which results in a nonlinear controller with improved performance over linear PI controllers.

Milanes. V. (2012) proposed twofold: on the one hand, to describe a comparative study of two intelligent control techniques-fuzzy and intelligent proportional-integral (PI) control, and on the other, to try to provide an answer to an as yet unsolved topic in the automotive sector-stop-and-go control in urban environments at very low speeds. Commercial vehicles exhibit nonlinear behaviour and therefore constitute an excellent platform on which to check the controllers. This described the design, tuning, and evaluation of the controllers performing actions on the longitudinal control of a car-the throttle and brake pedals-to accomplish stop-and-go manoeuvres. They are tested in two steps. First, a simulation model is used to design and tune the controllers, and second, these controllers are implemented in the commercial vehicle-which has automatic driving capabilities-to check their
behaviour. A stop-and-go manoeuvre is implemented with the two control techniques using two cooperating vehicles.

2.5 INTELLIGENT CONTROLLERS WITH NEURAL NETWORK FOR STATE ESTIMATION

The scope of using neural networks as estimators in process engineering was studied in detail by M.J.Willis, et al (1990). They also designed a neural network for estimating the biomass concentration in an industrial fermentation system.

The possibility of neural network being used as an emulator for a nonlinear process was demonstrated by D.H.Nguyen, et al (1990). Simulation results were also presented by considering a “truck backer-upper” model. However, the capability of the network to function effectively in the presence of obstacles were not demonstrated.

A neural network to estimate the proper timing for condition based maintenance of an airport ground transportation vehicle was presented by A.E.Smith, et al, 2002. However, the effects of noise were not considered in the study.

An intelligent controller for a level process control loop was presented by J.Zumberge, et al, 1996. The intelligent controller was equipped with estimators to estimate the process variable level. However, the effect of sensor noise was not considered in the study.

M.M.Gupta et al (1993) Presented the design of dynamic neural units as controller for unknown nonlinear systems. Presents techniques to study the adaptation capability of a neural network to variations in input signal and perturbations in load variable. Experimental results of this technique in the presence of noise have not been mentioned.

P.A.Minderman et al (1990) Presented the techniques for modeling chemical process systems via neural computation. Effect of model uncertainties is not analyzed. Examples on CSTR are presented. Real time implementation not carried out.

K.M.Passino (1995) Presented an intelligent controller for autonomous systems. Simulation results on a flexible link robot presented. Illustrates the idea that estimators need to be used as an integral part of intelligent controllers for improved performance.
Grimble and Ordys (2001) Presented a review of some well known predictive control algorithms like dynamic matrix control, predictive functional control and generalized predictive control. The combination of predictive control and linear quadratic Gaussian control will provide improved robustness together with adequate speed of response. No experimental works have been carried out.

Olga Chibirova et al (2008) Presented the design of dynamics of firing patterns in evolvable hierarchically organized neural networks Experimental results of this technique in the presence of noise have not been mentioned.

Caliskan F.(2007) presented a NN identification of icing parameters in an A340 aircraft and a reconfiguration technique to keep the A/C performance close to the performance prior to icing. Hardware implementation was carried out.

Youmin Zhang et al (2005) Presented a fault tolerant control system design technique and analyzed for managing performance degradation in the presence of multiple faults in actuators. Only simulation results are obtained.


Rubaai. A.(2011) presented the development of a fuzzy-neural-network (FNN) proportional-integral (PI)-proportional-derivative (PD)-like controller with online learning for speed trajectory tracking of a brushless drive system. The design implements the novel use of the extended Kalman filter (EKF) to train FNN structures as part of the PI/PD-like fuzzy design. The FNN structure has two parallel FNN PI/PD-like controllers, each with four internal layers. EKF trains each FNN by modifying the weights and the membership function parameters. Thus, the proposed EKF-based architecture presents an alternative to control schemes employed so far. The objective is to replace the conventional PI-derivative (PID) controller with the proposed FNN PI/PD-like controller with EKF learning mechanism. Comparisons of the algorithm performances provide evidence of improvement of the FNN PI/PD-like controller over PID control. A test bench enables design implementation in the laboratory on hardware using a dSPACE DS1104 DSP and MATLAB/Simulink environment. Experimental testing results show that the proposed controller learns and robustly responds to a wide range of operating conditions in real time.

2.6 INFERENCE
In most of the analogous works reported so far, some of the factors that should have been given more emphasis in the design of an intelligent system, (particularly for a process plant) have been found to be lacking. The works cited are mostly on simulation studies using transfer function models. Testing the algorithm on a simulated process does not always reflect the actual system performance and can neglect the effects of model mismatch, modeling uncertainties, sensor noise, etc. Though, a system is operated at a certain operating point, the model obtained at that point may not reflect the real system characteristics due to variations in the process parameters. Thus, any proposed intelligent control algorithm should be tested for satisfactory performance on a real time nonlinear prototype model.

Most of the real time processes are nonlinear, and is subjected to both slow motion and fast motion inputs i.e. SMS and FMS systems and has mutually interacting process variables with a vast amount of highly correlated data, and this good correlation need to be exploited in the design of intelligent system algorithms. Such exploitations are found to be missing in the literature survey done so far. However, in this work, a unified approach towards using intelligent control strategies by integrating data from mutually interacting, independently designed, separately located sensors and testing the algorithms on a real time nonlinear process is presented.

Practical system uses multiple sensors for monitoring various states and environmental conditions. Using this, effective multi-sensor signal processing algorithms with minimum human intervention are presented here, for extending the range of operation of a system with respect to online tuning and sensor validation.
CHAPTER 3

DESCRIPTION OF THE REAL TIME PLANT

The prototype model constructed for experimental study consists of the cylindrical tank with a conical bottom open to the atmosphere at the top end. The experimental model is to be used, to study the performance of the proposed intelligent control algorithms by obtaining the servo and regulatory response, in the presence of disturbances, feedback sensor failure and sensor noise. Suitable signals are given to a pneumatic operated control valve to regulate the manipulated variable inflow. Disturbances in the form of random variations in outflow (measurable) and/or changes in outflow coefficient are considered to enter the process. The schematic diagram of the plant is shown in figure 3.1. The process and instrumentation diagram (P&I) of the non-linear (hopper type tank) liquid level process is shown in figure 3.2. The process variable level is sensed by means of an RF capacitance probe and using suitable electronics circuitry, a voltage output is obtained. The analog voltage is converted into digital form using an 8-bit A/D converter. The inflow and outflow rates are measured using suitable flow transmitters. The details of the different plant components are given in table-3.1. The laboratory set up is shown in figure 3.1.

![Figure 3.1 Geometrical cross-section of the tank used in the mathematical model](image-url)
Figure 3.2 P&I diagram of the nonlinear hopper type tank chosen for experimental study

Table 3.1

Tank dimensions:

- Tank cylindrical portion height: 50.5 cms
- Tank conical portion height: 32 cms
- Conical portion angle: 1.88 degrees
- Outer diameter of tank: 32 cms
- Inner diameter of tank: 31.5 cms
- Outer circumference: 100 cms

Valve details:

- Bypass valve: ¼ “ ball valve
- Drainage valve: ¼ “ ball valve
I to P converter

Input : 4-20mA
Output : 3-15psi
Pressure span : 760mm of Hg to 1kg/cm²
Power supply : 24vd.c.

Level Sensor

Type : RF capacitance probe
Maximum distance of probe : 150cms
Resolution : 1pF
Repeatability : better than +/-1%
Zero and span range : 0 to 2000pF
Supply voltage : 14 to 40vd.c.
Figure 3.3 Laboratory set up of the plant model shown in figure 3.2
3.1 TRANSFER FUNCTION OF THE NONLINEAR PROCESS

The mathematical model of the nonlinear hopper type tank is developed, by considering the process as a combination of (i) a cylindrical geometry and (ii) a conical geometry. The plant transfer function is obtained in terms of the process characteristics, process gain and the process time constant. The dead time $t_d$ is neglected.
3.1.1 MODELING FOR THE CYLINDRICAL PORTION OF THE TANK

The cylindrical portion of the hopper type tank (figure 3.1) is considered with outflow rate proportional to the square root of level. The mass balance equation governing the system is given by

\[ \frac{dA}{dt} + A = f \]  \hspace{1cm} (3.1)

\[ \frac{d\sqrt{h}}{dt} + \sqrt{h} = \frac{f}{\sqrt{A}} \]  \hspace{1cm} (3.2)

Where \( A = \pi R^2 \)

The transfer function relating the height \( h \) and the inflow rate \( f \) with parameters \( (k, \tau) \) is derived as:

\[ \frac{\sqrt{h}}{f} = \frac{1}{k} e^{-\tau \frac{d}{dt}} \]  \hspace{1cm} (3.3)

The nominal transfer function

\[ \frac{\sqrt{h}}{f} = \frac{1}{k^0} e^{-\tau^0 \frac{d}{dt}} \]  \hspace{1cm} (3.4)

where \( k^0 \) and \( \tau^0 \) are evaluated at a nominal height \( h_0 \).

3.1.2 MODELING FOR THE CONICAL PORTION OF THE TANK

The conical portion of the hopper type tank (figure 3.1) is considered with outflow rate proportional to the square root of level. The mass balance equation governing the system is given by

\[ \frac{dA(h)}{dt} + A(h) = f \]  \hspace{1cm} (3.5)

\[ \frac{d\sqrt{h}}{dt} + \sqrt{h} = \frac{f}{\sqrt{A(h)}} \]  \hspace{1cm} (3.6)

where \( A(h) = \)  \hspace{1cm} (3.6)

When the operating point is in the conical region, we obtain a similar transfer function:
with the major difference that the area $A(h)$ is now a function of the height $h$.

### 3.2 CONTROLLERS AND THEIR STRUCTURE

The commonly employed control structures are illustrated in the form of block diagram in figure 3.5 (a) to figure 3.5(j)

**Figure 3.5 (a) Generic Controller**

**Figure 3.5 (b) Proportional Controller**
Figure 3.5 (c) PI Controller

Figure 3.5 (d) Modified I-D Controller

Figure 3.5 (e) Derivative Feedback with PI Controller
Figure 3.5 (f) Derivative feedback with delay and PI control

Figure 3.5 (g) P-I-D Controller
Figure 3.5 (h) Derivative feedback with delay and P control

Figure 3.5 (i) P-I Control with feedback
3.3 REPRESENTATION OF STABLE AND UNSTABLE SYSTEMS

The representation of system stability in 2-d quadrant is shown in figure 3.6

Figure 3.5 (j) Sampled PI control

Figure 3.6 System stability in 2-d quadrant
3.4 ANALOG CONTROLLERS

The hardware implementation of the different controllers discussed previously is presented in this section using analog components are shown in figure 3.7 and figure 3.8.

![Figure 3.7 Analog Controller PI](image)

Figure 3.7 Analog Controller PI
3.5 NEURAL NETWORKS AND THEIR CONTROL STRUCTURES

In this section the possibilities of using neural networks for nonlinear control is reviewed. The neural networks are generally viewed as process modeling formalism or even a knowledge representation framework. The
knowledge about the plant dynamics and mapping characteristics is implicitly stored within the network. The nonlinear functional mapping properties of neural networks are central to their use in control applications. Training a neural network using input-output data from a nonlinear plant is considered as a nonlinear functional approximation problem.

3.6 BASIC NEURAL LEARNING MODEL

One type of neural structure used for learning and control has to cope with uncertainties regarding plant dynamics and its environment, the controller has to estimate the unknown information during its operation. If this estimated information gradually approaches the true information as time proceeds, then the controller approaches that of an optimal controller. Such a controller can be viewed as an adaptive controller, due to the gradual improvement of the estimated information. The controller learns the unknown information during operation, and this information, in turn, is used as an experience for future decision and controls. A control system is called a learning control system if the information pertaining to the unknown features of the plant or its environment is acquired during operation, and the obtained information is used for future estimation, recognition, classification, control or decision such that the overall system performance is improved. Once learning is complete the control system can compensate for large number of changes in the plant and its environmental conditions. The difference between adaptive and learning system lies in the fact that the former treats every distinct operating situation as novel, whereas the latter correlates the past experience with the present situations and accordingly adapts its behavior.

3.7 ANN FOR SYSTEM IDENTIFICATION

An obvious approach for system modeling is to choose the input-output structure of the neural network to be the same as that of the system.
If the output of the network is defined as \( y \) then the system relationship can be expressed as \( y(k) = f(y_p(k-1), \ldots, y_p(k-n); u(k), \ldots, u(k-m)) \). It is a well established fact that a feedforward network of the multilayer perceptron type can approximate arbitrarily well a continuous function (Cybenko 1988, Funahashi 1989). The different types of neural network architectures used in system identification are (i) forward modeling (ii) inverse modeling (iii) model reference control (iv) internal model control (v) optimal decision control and (vi) predictive control.

3.8 FORWARD MODELING

The neural network is trained to represent the forward dynamics of a system. The ANN model is placed in parallel with the system (Jordan, et al, 1991) and the error (the prediction error) between the system and network outputs is used as the network training signal as shown in figure 3.9. To train the network, target values have to be provided directly in the output coordinate system of the learner.

![Figure 3.9 Forward modeling neural network](image)

3.9 TWO LAYERED FEED FORWARD NEURAL NETWORK
A layered feedforward neural network consists of Adalines connected together as shown in Figure 3.10. A layer of Adalines is created by connecting a number of Adalines to the same input vector. Many layers can then be cascaded, with output of one layer connected to the inputs of the next layer, to form a network. It has been proven that a network consisting of only two layers of Adaline can implement any nonlinear function $X, d(X)$ given enough Adalines in the first layer (S. Miyake et al., 1988). The idea is that each Adaline in the first layer can take a small piece of the function relating $X$ to $d(X)$ and make a linear approximation to that piece. The second layer then adds the pieces together to form a complete approximation to the desired function.

Figure 3.10 Two layer feed forward neural network
CHAPTER 4

NEURO – FUZZY LOGIC

The fusion of ideas from fuzzy control and neural networks had acknowledged a significant role in improving controller performances. Fuzzy logic has proven effective for complex, non-linear and imprecisely defined systems. The common bottleneck in fuzzy logic is the derivation of fuzzy rules and the parameter tuning for the controller. The neural networks have powerful learning abilities, optimization abilities and adaptation. The fuzzy logic and neural networks can be integrated to form a connectionist adaptive network based fuzzy logic controller. This integrated adaptive system modifies the characteristics of the rules, topology of fuzzy sets and/or the structure of control system.

4.1 FUZZY LOGIC CONTROLLER

In the real world, data that characterizes a system can be both numeric as well as linguistic in nature. Fuzzy rules and fuzzy reasoning are the backbone of fuzzy inference system, which are most important modeling tool based on fuzzy set theory. They have been successfully applied to wide range of areas.

The fuzzy If – THEN rules also known as fuzzy rules assumes the form,

\[
\text{If } x \text{ is } A \text{ then } y \text{ is } B.
\]

Where A and B are the linguistic values defined by fuzzy sets on universe of discourse x and y respectively. Often ‘\(x \text{ is } A\)’ is called the antecedent or premise, while "\(y \text{ is } B\)" is called the consequence or conclusion.

Examples of fuzzy if – then rules are,

- If pressure is high, then volume is small
- If the speed is high, then apply the brake a little.

Fuzzy Inference system (FIS) is a popular computing framework based on the concept of fuzzy set theory and fuzzy IF – THEN rules.

The basic structure of a fuzzy inference system consists of three conceptual components: a rule base, which contains a selection of fuzzy rules; a database which defines the membership
functions used in the fuzzy rules; and a reasoning mechanism, which performs the inference procedure upon the rules and given facts to derive a reasonable output or conclusion.

The system can take either fuzzy inputs or crisp inputs, but the outputs it produces are almost always fuzzy sets. Sometimes it is necessary to have a crisp output, especially in a situation where a fuzzy inference system is used as a controller. Therefore, we need a method of defuzzification to extract a crisp value that best represents a fuzzy set. With a crisp input and outputs a fuzzy inference system implements a nonlinear mapping from its input space to output space. This mapping is accomplished by a number of fuzzy IF–THEN rules.

Fuzzy Logic control (FLC) has proven effective for complex, non–linear and imprecisely defined processes for which standard model based control techniques are impractical or impossible. Fuzzy Logic, unlike Boolean or crisp logic, deals with problems that have vagueness, uncertainty and use membership functions with values varying between 0 and 1. Fuzzy Logic tends to mimic human thinking that is often fuzzy in nature.

In fuzzy logic a particular object has a degree of membership in a given set, which is in the range of to 1. The essence of fuzzy control algorithms is a conditional statement between a fuzzy input variable A and a fuzzy output variable B. This is expressed by a linguistic implication statement such as.

IF A THEN B

In general a fuzzy variable is expressed through a fuzzy set, which in turn is defined by a membership function \( \mu \).

4.2 CONFIGURATION OF FLC

The basic configuration of an FLC developed by Lofti A. Zadeh (1984) is given in figure 4.1. It comprises of four principal components:

1. A fuzzification interface
2. A knowledge base
3. A decision – making logic and
4. A defuzzification interface
1. The fuzzification interface involves the following functions.
   (a) measures the values of input variable.
   (b) performs a scale mapping that transfer the range of values of input variable into corresponding universe of discourse.
   (c) Performs the function of fuzzification that converts input data into suitable linguistic values.
2. The knowledge base consists of data base and a linguistic control rule base.
   a) The database provides necessary definitions, which are used to define linguistic control rules.
      (b) The rule base characterize the control goals and control policy of the domain experts by means of a set of linguistic control rules.
3. The decision – making logic is the kernel of an FLC. It has the capability of simulating human decision – making based on fuzzy concepts and of inferring fuzzy control actions employing fuzzy implication and the rules of inference in fuzzy logic.
4. The defuzzification interface performs the following function.

(a) A scale mapping, that converts the range of values of output variables into corresponding universe of discourse.

(b) Defuzzification, which yields a non–fuzzy control from an inferred fuzzy control action.

Thus the idea behind the FLC is to fuzzify the controller inputs, and then infer the proper fuzzy control decision based on defined rules. The output is then produced by defuzzifying this inferred control decision.

### 4.2.1 FUZZIFICATION AND MEMBERSHIP FUNCTIONS

Fuzzification is a process of transferring the crisp (real) control variables to corresponding fuzzy variables. Selection of the control variables relies on the nature of the system and its desired output. The FLC input and output signals are interpreted into a number of linguistic variables. The number of linguistic variables varies according to the application. Increasing the number of linguistic variables results in a corresponding increase in the number of rules. Each linguistic variable has its fuzzy membership function. The membership function maps the crisp values into fuzzy variables.

A fuzzy set $A$ in $X$ is defined as

$$A = \{(x, \mu_A(x))/x \in X\}$$

where $\mu_A(x)$ is called membership function (MF). The membership grade of each element of $X$ is in the range 0-1.

Some of the one–dimensional membership functions commonly used are:

1. Triangular membership functions.
2. Gaussian membership functions.
3. Trapezoidal membership functions.

The general description of fuzzy classes of triangular MF's is shown in figure 4.2. It is specified by three parameter $\{a, b, c\}$ with $(a < b < c)$ are the $x$ – coordinates of the three corners of the MF.
The class of Gaussian membership functions is shown in Figure 4.3. It is given by
\[ \mu_{Ai}(x_i) = \frac{(c_i - x_i)^2}{2\sigma_i^2} \]
where \( c_i \) and \( \sigma_i \) are the center and width of the \( i^{th} \) fuzzy set \( A_i \), respectively.

The Trapezoidal membership function is specified by four parameters \((a, b, c, d)\). The parameters \( \{a, b, c, d\} \) with \( a < b \leq c < d \) determine the \( x \) – coordinates of the four corners of the underlying trapezoidal MF. The membership function description of trapezoidal family is described in figure 4.4.
4.2.2 RULES CREATION AND INFESSION

In general, fuzzy systems map input fuzzy sets to output sets. Fuzzy rules are relations between input/output fuzzy sets. The modes of deriving fuzzy rules are based either of the following.

- Expert experience and control engineering knowledge.
- Operators control actions.
- Learning from the training examples

The general form of the fuzzy control rules in learning from training examples case is

\[
\text{IF } x \text{ is } A_i \text{ AND } y \text{ is } B_i \text{ THEN } z = f_i(x, y)
\]

where \( x, y \) and \( z \) are linguistic variables representing the process state variables and the control variable respectively. \( A_i, B_i \) are the linguistic values of the linguistic variables, \( f_i(x, y) \) is a function of the process state variables \( x, y \) and the resulting fuzzy inference system (FIS) is called a first – order sugeno fuzzy model.

4.2.3 DEFUZZIFICATION

The function of the inference engine is to calculate the overall value of the control output variable based on the individual contributions of each rule in the rule base. (i.e) the defuzzification process. There is no systematic procedure for choosing defuzzification. In first – order sugeno fuzzy model each rule has a crisp output and overall output is obtained as weighted average thus avoiding the time consuming process of defuzzification required in a conventional FLC.

4.3 DEVELOPMENT TOOLS NETWORK CATEGORIES AND ARCHITECTURES

The majority of the variations between neural network architectures is due to the various learning rules and how those rules modify the network’s topology. Most applications of neural networks fall into five primary categories:

- prediction (regression)
- classification
- data association
- dataconceputalisation
- data filtering
Table 4.1 Network Category Architecture and Typical Application

<table>
<thead>
<tr>
<th>Network category</th>
<th>Architecture</th>
<th>Typical use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction (Regression) function approximation</td>
<td>Feedforward backpropagation  Direct random search  Self organizing map into backpropagation  General regression NN  Mixture density networks</td>
<td>Use input to predict some output (typical regression type problems)</td>
</tr>
<tr>
<td>Classification</td>
<td>Learning vector quantization  Counter propagation  Probabilistic neural network  Mixture density network</td>
<td>Use input values to determine the classification</td>
</tr>
<tr>
<td>Data association</td>
<td>Hopfield  Boltzmann machine  Bidirectional associative memory  Spatio-temporal pattern recognition</td>
<td>Used as in classification networks but also recognize data containing errors</td>
</tr>
<tr>
<td>Data conceptualization</td>
<td>Adaptive resonance network  Self organizing feature map</td>
<td>Analyse inputs so that grouping relationships can be inferred</td>
</tr>
<tr>
<td>Data filtering</td>
<td>Recirculation</td>
<td>Smooth an input signal</td>
</tr>
</tbody>
</table>

Table 4.1 shows the differences between the five primary network categories and shows typical architectures and applications for each category (Anderson, 1992). However, when developing models for engine performance prediction, the interest is with architectures that can model continuous functions of input variables. Thus, the obvious network category is prediction/function approximation. In this category, the feed forward back-propagation network is the most popular due to its relatively simple implementation and it has been successfully used to solve many types of problems in a range of engineering applications. Therefore, for the problem of this study, the feed forward back propagation network architecture was chosen. The engine performance mapping problem can be thought of as a nonlinear regression problem. In the next section, nonlinear regression is briefly explored so as to give insight into the neural network architecture and functionality.
4.3.1 BACK PROPAGATION LEARNING ALGORITHM

Based on this algorithm the networks learns a distributed associative map between the input and output layers. What makes this algorithm different than the others is the process by which the weights are calculated during the learning phase of the network. In general, difficulty with multilayer perceptions is calculating the weights of the hidden layers in an efficient way that results in the least (or zero) output error; the more hidden layers there are; the more difficult it becomes. To update the weights, one must calculate an error. At the output layer this error is easily measured; this is the difference between the actual and desired (target) outputs. At the hidden layers however, there is no direct observation of the error, hence some other technique must be used to calculate error, as this is the ultimate goal.

4.3.2 TRAINING WITH BACK PROPAGATION ALGORITHM

During the training session of the network a pair of patterns is presented \((x_k, d_k)\), where \(x_k\) is the input pattern and \(d_k\) is the target or desired pattern. The \(x_k\) pattern causes output responses at each neuron in each layer and, hence actual output \(O_k\) at the output layer. At the output layer, the difference between the actual and target outputs yields an error signal. This error signal depends on the values of the weights of the neurons in each layer. This error is minimized, and during this process new values for the weights are obtained. The speed and accuracy of the learning process (i.e), the process of updating the weights also depends on factor known as the learning rate.

The basis for this weight update algorithm is simply the gradient – descent method as used for simple perception with differentiable units. For a given input-output training pair \((x_k', d_k')\) the back – propagation algorithm performs two phase of data flow. First the input pattern is propagated from the input layer to the output layer and , as a result of this forward flow of data, it produces an actual output \(y_k\). Then the error signals resulting from the difference between \(d_k\) and \(y_k\) are back – propagated from the output layer to previous layers for them to update their weights.
Let us consider 'm' PEs (Processing elements) in the layer 'l' PEs in the hidden layer and 'n' PEs in the output layer as shown in figure 4.5.

Consider an input - output pair $(x, d)$ a PE $q$ in the hidden layer receives a net input of

$$\text{net } q = \sum v_{qj} x_j = 1, \ldots, m \quad \text{-------------------(4.1)}$$

and produces an output of

$$z_q = a (\text{net } q) = a (\sum v_{qj} x_j) \quad \text{-------------------(4.2)}$$

The net input for a PE 't' in the output layer is then

$$\text{net } i = \sum w_{iq} z_q = \sum w_{iq} a (\sum v_{qj} x_j) = 1, \ldots, 1 \quad \text{-------------------(4.3)}$$

and it produces an output of

$$y_i = a (\text{net } i) = a (\sum w_{iq} z_q) = a (\sum w_{iq} a (\sum v_{qj} x_j)) \quad \text{-------------------(4.4)}$$

The above equations indicate the forward propagation of input signals through the layers of neurons, next, we shall consider the error signals and their back propagation.

$$E(w) = \frac{1}{2} \sum (d_i - y_i)^2 \quad I = 1, \ldots, n \quad \text{-------------------(4.5)}$$

Then according to the gradient – decent method the weights in the hidden to output connections are updated by

$$\Delta w_{iq} = -\eta \frac{\partial E}{\partial w_{iq}}$$
where $\delta_{oi}$ is the error signal and its double subscript indicates the $i^{th}$ node in the output layer. The error signal is defined by

$$\delta_{oi} = -\frac{\partial E}{\partial o_i} = -[\frac{\partial E}{\partial y_i}] [\frac{\partial y_i}{\partial \text{net } i}]$$

For the weight update on the input to hidden connections the chain rule with the gradient descent method is used and the weight update on the link weight connecting PE $i$ to PE $q$ in the hidden layer is obtained.

$$\Delta v_{iq} = -\eta [\frac{\partial E}{\partial v_{iq}}] = \eta \delta_{nq} x_i$$  (4.7)

where $\delta_{nq}$ is the error signal of PE $q$ in the hidden layer and is defined as

$$\delta_{nq} = -\frac{\partial E}{\partial \text{net } q}$$

where $\text{net } q$ is the net input to the hidden PE $q$. The error signal of PE in a hidden layer is different from the error signal of a PE in the output layer. Because of this difference, the above weight update procedure is called the generalized delta learning rule.

The process of computing the gradient and adjusting the weights is repeated until a minimum error is found. In practice, one develops an algorithm termination criterion so that the algorithm does not continue this iterative process forever.

In summary the error back-propagation algorithm can be outlined as

**Step 1:** Initialize all weights to small values.

**Step 2:** Choose an input – output training pair.

**Step 3:** Calculate the actual output from each neuron in a layer by propagating the signal forward through the network layer (forward propagation)

**Step 4:** Compute the error value and error signals for output layer.

**Step 5:** Propagate the errors back ward to update the weights and compute the error signals for the preceding layers.

**Step 6:** Check whether the whole set of training data has been cycled once, yes – go to step 7: Otherwise go to step 2.
Step 7: Check whether the current total error is acceptable; yes – terminate the training process and output the field weights otherwise initiate a new training epoch by going to step 2.

4.4 DESIGN OF FUZZY LOGIC CONTROLLER (FLC)

The block diagram of the plant with $e(k)$ and $ce(k)$ chosen as fuzzy variables (go and g1 are scaling blocks) along with the FLC and the sensor (i.e. estimated variable) is shown in figure 4.6.

Figure 4.6 Design of FLC

The set of rules of FLC using which the manipulated variable is generated is shown in table 4.2

<table>
<thead>
<tr>
<th>Table 4.2 GENERIC FUZZY RULE BASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting up, change the input, in response to the set point change</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Error not changing, change input accordingly</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Moving along; maintain input</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Condition</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>If $e_n$ is SN and $ce_n$ is SP then $\delta u_n$ is Z</td>
</tr>
<tr>
<td>If $e_n$ is LN and $ce_n$ is SP then $\delta u_n$ is Z</td>
</tr>
<tr>
<td>Getting worse, reverse input somewhat</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Error changing too fast, adjust input somewhat</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Reached equilibrium</td>
</tr>
<tr>
<td>Error is nil but changing, take action</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Error is nil and changing insignificantly, no action</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The membership function for the fuzzy variables $e(k)$, $ce(k)$ and $\Delta u(k)$ is shown in figure 4.7.
Figure 4.7 Membership function for $e(k)$, $ce(k)$ and change in control variable $\Delta u(k)$

4.5 TRAINING THE FUZZY ESTIMATOR

The order of the estimator needs to be specified. The term as employed here, refers to the length of the time histories to be used within the model structure. The use of time-histories imparts dynamic characteristics to the model, which is important if the estimated values are to be used for automatic feedback control. Using the input-output data is formed and used to train the fuzzy estimator as well as for comparison study. The controller output is chosen as $u(t) = \text{sgn}(t)$ i.e. both liquid pull in and pull out and the system is trained for both positive and negative inputs. Furthermore, the fuzzy estimation system needs to be trained with a “sufficiently rich” input. To satisfy these requirements, random noise input is chosen for all the training. This is shown in figure 4.8.
The algorithm for estimating the system model is as follows:

(i) Generate the input and output data.

(ii) Using White noise (limited to values of 5) as input gather training data. Samples shall be taken from the truth model at a low frequency (rate of 10Hz).

(iii) Once the input and output data are collected, create a regression vector grouping together the input and output. In this work, specifically, the following regression vector of length seven is chosen:

\[ x = [u(k-3), u(k-2), u(k-1), u(k), h(k-3), h(k-2), h(k-1)]^T \]

and the corresponding output is \( y = h(k) \). The regression vector length can be adjusted to produce the desired results.

(iv) Fuzzy clustering with optimal output pre-defuzzification is used to create an estimated system whose input/output characteristics will model the plant.

\[ \text{Figure 4.8 Parameter estimator using data driven model} \]

4.5.1 TUNING THE FUZZY ESTIMATOR MODEL

Several parameters are available for adjustment so as to obtain the best response, namely:

(i) the number of data pairs \( M \),

(ii) the number of rules \( R \),
(iii) the initial cluster centers and
(iv) the parameter ‘m’.

4.5.2 DATA PAIRS AND OUTLIERS FILTERING

To assist the tuning, i.e. to optimize the number of parameters used for training, heuristic knowledge is chosen to define the starting point. The size of the dataset needs to be fair and also not over fitted. In this work, a fairly large number is used initially, and then lowered to observe the effect on the estimation. For this simulation, $M = 250$ data sets were chosen and found to be quite sufficient. As the flow chart (Figure 4.9) shows, the first and most important thing to do is to rid the data of spurious/errant points or outliers. These can have significant impact on the model structure selection and estimator testing stages of the development cycle. Next, noise in the data should be attenuated as much as possible. In this work, the use of low phase shift filters (filters that have the least time-lags on the processed signals) is used.

![Flow Chart: Data Preprocessing](image-url)

**Figure 4.9 Data preprocessing**
4.5.3 NUMBER OF RULES

The number of rules is equally arbitrary, obviously limited to the $R < M$ and optimally twenty rules produced very good results for the system.

4.5.4 CLUSTER CENTERS AND PARAMETER ‘M’

The initial cluster centers were placed randomly between -1 and 1 to ensure that the initial centers are somewhat spread out. Other choices include randomly placed centers between the minimum and maximum value of the data points. This way, the initial centers will be spread out and also entirely within the data set space. However, the first method is chosen simply because of its ease of implementation. For this work $M = 4$ was used.

4.6. ESTIMATION PROCEDURE

The estimation procedure consists of training the input and output membership functions. The sampling period is 0.1 sec. Once all the initial parameters are determined, the estimation is done as follows:

1. The $(\text{for } i=1, 2, \ldots, M \text{ and } j=1, 2, \ldots, R)$ values are computed for each data set and each rule using the training data.
2. The new cluster centers, $(\text{for } j=1, 2, \ldots, R)$, are found using the
3. The convergence condition is chosen as
   
   If
   
   $(\Delta \text{updated centers} < \text{the specified (here, } \epsilon = 0.001) \text{ from the previous centers})$

   then
   
   (the training of cluster centers is complete)

   else
   
   repeat 1.

The convergence condition depends on where the initial centers were placed. Since we are randomly placing the centers, the number of iteration is bound to vary. In this work, once the training of the input membership function centers is complete, least squares estimation is used to place the output membership function centers. The complete estimator was programmed in Matlab using the steps described above.
4.7 PLANT WITH SENSOR VALIDATION MODEL USING NEURAL ESTIMATOR

The plant failure model with the neural estimator to take care of feedback sensor failure is shown in figure 4.10.

![Block diagram of plant model with Neural Estimator to take care of feedback sensor failure](image)

**Figure 4.10 Plant model with Neural Estimator to take care of feedback sensor failure**

In the model, the decision logic determines the feedback signal to be provided to the controller, by computing the deviation between the estimator output and the plant output, and comparing this deviation with a pre-defined threshold value. It is proposed to construct a neural state estimator to estimate a single parameter in the plant “g”. For this purpose, random excitation inputs were chosen to form the training data set. Excitation with random inputs was chosen, since, it had a better tendency to place the data points over a whole range of locations and also it is difficult to choose other inputs “u” that result in a better data set G.. A set of experiments were conducted with system “g” by varying the parameters fin(k) and fout(k) about its steady state values. The parameters were varied individually over a specified range of values to account for the possible failure scenarios.
the system might encounter. The random variations in inflow rate, outflow rate about its steady state values (that were used as excitation inputs for forming the training data set) and the resultant plant response obtained experimentally is shown in Appendix I (figure A and B). The parameters fin(k) and fout(k) were varied between –50% and +50% of its nominal value i.e. fin(k) and fout(k) [-0.5,+0.5], and the steady state deviations of the plant output was recorded.

4.8 DESIGN OF ESTIMATOR USING NEURAL NETWORK

Training a neural network using input-output data from a nonlinear plant is considered as a nonlinear functional approximation problem. A generic neural network estimator model, used to detect a sensor failure is shown in figure 4.11. Neural networks have effectively been used in many applications to predict performance degradation of operating systems in real-time. Neural networks are data driven models and data under a variety of conditions need to be obtained.

In the present work the experimental setup was used to gather data and the key measurable signals that were collected for training the network consisted of the inflow, outflow rate and the process value level. Different operating conditions were simulated and the change in inflow, outflow and the level were recorded. The data collected from the plant were pre-processed for normalization and fed to the Neural network for training. Data pre-processing was performed as the data obtained from the experiment in not ready to use for training directly. The first step in pre processing is to identify as samples that carry high leverage. Outliers can result from sensor failure, misreading from lab tests and other possible unknown upsets to the process.

A distinctive feature of outliers is that they have extremely large influence on the model. As a sequence, it is necessary to perform outlier detection and pretreatment before training the network. The presence of outliers in the present data set is identified, by observing the signals in frequency domain. The network was trained with the back propagation algorithm shown in figure 4.12.
Figure 4.11 Process with “U” control input, “d” disturbance and “y” measured output

Figure 4.12 Neural Network during Training with weights adaptive

Figure 4.13 Trained Neural Network as fault tolerant system
4.9 SUMMARY

In this chapter, the primary features of neural networks have been presented with the emphasis on the components that constitute a feed forward back propagation network used for nonlinear function approximation. Further, the chapter has laid the foundation for consistent discussion in the later chapters using the terminology associated with neural networks. The following is a point summary of the pertinent issues: Feed-forward neural networks have been successfully applied to many practical engineering problems due to their ability to learn by example and their capability of modeling complex functions and processes. This type of network is able to effectively address the curse of dimensionality problem besetting many of the standard modeling techniques used for engine performance mapping. A feed-forward network proposed in this study is a viable option for performance mapping of direct injection diesel engines. Neural network development exacts a number of specific conditions. The two primary conditions being that the training data characterizes the problem (function) being modeled, and that it is adequately sized to both train and test the network.

There is evidence that use of asymmetric activation functions in the hidden layers results in improved learning. Therefore, hyperbolic tangent activation functions are recommended for use. Also, although not conditional when sigmoid activation functions are used, trainable bias terms
should be used on the processing elements since there is evidence that network training is improved with their use (specifically on the output processing elements).
CHAPTER 5

PERFORMANCE OF FUZZY AND NEURAL ESTIMATOR

5.1 TRAINING DATA FOR NEURAL NETWORK

Sample values of input-output patterns obtained from the response and subsequently used for training the network is given in table 5.1. The training data should be spread over the input space uniformly to ensure that there is a regular spacing between points and not too many more points in one region than another [5]. This is essential to get a good coverage of the whole input space. The information as to how the mapping “g” is shaped in all regions should be implicitly presented as much as possible in the training data set.

Table 5.1 Input Training Data

<table>
<thead>
<tr>
<th>Fin (k)</th>
<th>fout(k) (in%)</th>
<th>average change in liquid level/sample (cms/sample)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>0%</td>
<td>0.10625</td>
</tr>
<tr>
<td>100%</td>
<td>20%</td>
<td>0.09</td>
</tr>
<tr>
<td>100%</td>
<td>40%</td>
<td>0.0812</td>
</tr>
<tr>
<td>100%</td>
<td>50%</td>
<td>0.06818</td>
</tr>
<tr>
<td>100%</td>
<td>65%</td>
<td>0.03846</td>
</tr>
<tr>
<td>100%</td>
<td>80%</td>
<td>0.02666</td>
</tr>
<tr>
<td>90%</td>
<td>0%</td>
<td>0.094176</td>
</tr>
<tr>
<td>90%</td>
<td>20%</td>
<td>0.081632</td>
</tr>
<tr>
<td>90%</td>
<td>40%</td>
<td>0.074</td>
</tr>
<tr>
<td>90%</td>
<td>50%</td>
<td>0.059259</td>
</tr>
<tr>
<td>90%</td>
<td>65%</td>
<td>0.053333</td>
</tr>
<tr>
<td>90%</td>
<td>80%</td>
<td>0.026229</td>
</tr>
<tr>
<td>80%</td>
<td>0%</td>
<td>0.0805</td>
</tr>
<tr>
<td>80%</td>
<td>20%</td>
<td>0.0744186</td>
</tr>
<tr>
<td>80%</td>
<td>40%</td>
<td>0.05927</td>
</tr>
<tr>
<td>80%</td>
<td>50%</td>
<td>0.05317</td>
</tr>
<tr>
<td>80%</td>
<td>65%</td>
<td>0.04</td>
</tr>
<tr>
<td>80%</td>
<td>80%</td>
<td>0</td>
</tr>
<tr>
<td>65%</td>
<td>0%</td>
<td>0.057142</td>
</tr>
<tr>
<td>65%</td>
<td>20%</td>
<td>0.04507</td>
</tr>
<tr>
<td>65%</td>
<td>40%</td>
<td>0.034408</td>
</tr>
<tr>
<td>65%</td>
<td>50%</td>
<td>0.02758</td>
</tr>
</tbody>
</table>
5.2 EXPERIMENTAL RESPONSE WITH THE NEURAL ESTIMATOR

The performance of the designed neural estimator was tested on the nonlinear hopper type tank by introducing feedback sensor failure at random time instants during the experimental run. The decision logic of figure 4.10 selects the neural estimator output as the feedback signal to the controller at those time instants when the deviation between the actual sensor value and the estimated value exceeds a set threshold. The actual plant response that would have been obtained with a faultless sensor was compared with the estimator response for different operating conditions such as variations in set point and outflow. These responses were obtained independently with fuzzy controller present in the forward path of the control loop. The ISE is calculated for both the servo and regulatory control with the estimator alone in the loop.

5.3 PERFORMANCE OF THE ESTIMATOR

Performing several process reaction curve experiments with the same input change would not provide the evidence to evaluate linearity. However, this would be useful data to evaluate the accuracy of the estimated parameters. Once the estimator design is complete, testing of the system in necessary to judge the performance of the system. The test flow is illustrated in figure 5.1. In this work, the step, square and sinusoidal test inputs were chosen to test the performance of the designed estimator.

<table>
<thead>
<tr>
<th>$F_{in}(k)$ (in %)</th>
<th>$F_{out}(k)$ (in %)</th>
<th>average change in liquid level/sample (cms/sample)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>20%</td>
<td>0.01758</td>
</tr>
<tr>
<td>50%</td>
<td>30%</td>
<td>0.013636</td>
</tr>
<tr>
<td>50%</td>
<td>40%</td>
<td>0.0056818</td>
</tr>
<tr>
<td>40%</td>
<td>0%</td>
<td>0.0125</td>
</tr>
<tr>
<td>40%</td>
<td>20%</td>
<td>0.01</td>
</tr>
<tr>
<td>40%</td>
<td>30%</td>
<td>0.006818</td>
</tr>
<tr>
<td>30%</td>
<td>0%</td>
<td>0.012</td>
</tr>
<tr>
<td>30%</td>
<td>10%</td>
<td>0.006</td>
</tr>
<tr>
<td>50%</td>
<td>20%</td>
<td>0.01758</td>
</tr>
<tr>
<td>50%</td>
<td>30%</td>
<td>0.013636</td>
</tr>
<tr>
<td>50%</td>
<td>40%</td>
<td>0.0056818</td>
</tr>
<tr>
<td>40%</td>
<td>0%</td>
<td>0.0125</td>
</tr>
<tr>
<td>40%</td>
<td>20%</td>
<td>0.01</td>
</tr>
<tr>
<td>40%</td>
<td>30%</td>
<td>0.006818</td>
</tr>
</tbody>
</table>
5.3.1 STEP RESPONSE OF THE ESTIMATOR

The resulting response (with initial condition of $h(0)=1$) is shown in figure 5.2. Also shown on this plot is the actual response of the system. It can be observed that the estimated system performs consistently.
5.3.2 RESPONSE OF THE ESTIMATOR TO 0.5 HZ SQUARE WAVE INPUT

Further testing the system, a square wave reference value with frequency of 0.5 Hz and magnitude 1.0 (and DC offset of 2) was input into the estimated system with initial condition $h(0)=1$. This is shown in figure 5.3. From this, one can see that the estimated system has a response that follows the actual system very well.

![Figure 5.3 Result of the Fuzzy estimator with a 0.5Hz square wave input with initial level of the tank at 1 meter (solid line = estimator response, dotted = truth model response)](image)

5.3.3 RESPONSE OF THE ESTIMATOR TO 0.05 HZ SQUARE WAVE INPUT

The result of 0.05Hz square wave input with initial condition of $h(0)=0.5$ is shown in figure 5.4.

![Figure 5.4 Result of the Fuzzy estimator with a 0.05Hz square wave input with initial level of the tank at 0.5 meter (solid line = estimator response, dotted = truth model response)](image)
Figure 5.4 Result of the Fuzzy estimator with a square wave input with initial condition of \( h(0) = 0.5 \) (solid line = estimator response, dotted = truth model response)

5.4 RESPONSE TO LOW FREQUENCY SINE WAVE

A sine wave of frequency 1 Hz was input into the estimated system with initial condition \( h(0)=1 \). The results are shown in figure 5.5. Also on this figure is shown the resulting response of the truth model to the same input. The estimated model performs very close to the actual plant.

Figure 5.5 Result of the Fuzzy estimator with a 1 Hz sine wave input (solid line = estimator response, dotted = truth model response)
5.5 DESIGN OF A GENERIC FLC

The software of the FLC system consists of main process and the subroutines. The main procedure includes initialization and function subroutines. The Flow chart of the FLC is shown in figure 5.6.

Figure 5.6 FL control
5.6 FUNCTIONS SUBROUTINE KB AND RB SELECTION

In the function subroutines initialization, first the knowledge base and rule base varieties are initialized as shown in figure 5.7 where in the $P_a$, $O_{as}$, $P_{im}$, $B_a$ are assigned with the $7 \times 7 \times 7$ rules are optimized.

![Flowchart of KB and RB Initialization]

Figure 5.7 KB and RB initialization
5.7 FUNCTIONS SUBROUTINES WAS TUNING $\mu P_Z, \mu ASO, \mu P_{IM}, \mu OUT$

The tuning of function $P_a$, $O_{as}$, $P_{im}$ and $B_a$ are explained in figure 5.8, where in the membership function was varied and system response was obtained. The choice of membership function width corresponding to minimum MAE was chosen.

Figure 5.8 Tuning of coefficients
CHAPTER 6

RESULTS AND DISCUSSION

6.1 RESPONSE TO SET POINT VARIATIONS

In the servo tracking experimental study on the real time plant, step signal with randomly varying magnitudes were used as the excitation input. The chosen variations of input signal $sp(k)$ in the interval $[-20, 20]$ is shown in figure 6.1, for the first 500 samples. The obtained servo and neural estimated response of the nonlinear plant with the fuzzy controller in the forward path of the control loop is shown in figure 6.1. The objective of this experimental study is to study the input signal adaptation capability of the designed neural estimator.

![Set point variations](image)

**Figure 6.1** Set point variations chosen for the experimental study
Figure 6.2 Measured variations of manipulated variable inflow (%)

Figure 6.3 Measured variations of load variable outflow (%)

[Graphs showing variations over time]
6.2 RESPONSE TO VARIATIONS IN LOAD VARIABLE OUTFLOW

The regulatory response of the plant estimated by the neural estimator during level sensor failure, with the fuzzy controller in the loop is shown in figure 6.5. The perturbations introduced in the load variable outflow were exactly the same.
Figure 6.5 Measured and neural estimated level variations of the real time plant with fuzzy controller in response to perturbations in load variable outflow.

6.3 ONLINE ACQUIRED PLOTS

The on line acquired plots for the process variable of level, outflow, estimated value for regulatory tracking and set point tracking are shown in figure 6.6 and 6.7.

Figure 6.6 On-line acquired plots showing the true process value of outflow (1st quadrant), Neural network estimated value (2nd quadrant) and fuzzy estimator o/p value (4th quadrant). (Plot obtained during regulatory tracking)
Figure 6.7 On-line acquired plots showing the true process value of level (1st quadrant), Neural network estimated value (2nd quadrant) and the fuzzy estimator o/p value (4th quadrant). (Plot obtained during set point tracking)

6.4 COMPARISON RESULTS

The experimental servo and regulatory response of the system with the two designed estimators were obtained for the following cases, and shown in table 6.1

1) Fuzzy estimator with fuzzy controller for servo tracking

2) Fuzzy estimator with fuzzy controller for regulatory response

3) Neural estimator with fuzzy controller for servo tracking

4) Neural estimator with fuzzy controller for regulatory response

Table 6.1 Normalized MSE

<table>
<thead>
<tr>
<th>Type of estimator /controller</th>
<th>Setpoint tracking</th>
<th>Regulatory response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy estimator with Fuzzy controller</td>
<td>404.37</td>
<td>360.03</td>
</tr>
<tr>
<td>Neural estimator with fuzzy</td>
<td>402.83</td>
<td>346.08</td>
</tr>
</tbody>
</table>
6.5 EXPERIMENTAL RESULTS

The realtime plant with the controller (FLC based) is shown in Appendix - I. The nonlinearity is due to the geometry of the process. Additionally, Dead Zone, ON/OFF and linear Dead Zone nonlinearities are introduced into the system and the output with the estimator is obtained. Figure 6.8 to figure 6.10 represents the output for three different nonlinearities (apart from the process geometry) i.e. (i) Dead Zone (ii) ON/OFF and (iii) Linear Deadzone nonlinearity. The phase plane plot for each case is also shown.
Figure 6.8 Plant response with Dead Zone nonlinearity
Figure 6.9 Plant response with ON/OFF nonlinearity
Figure 6.10 Plant response with Linear Dead Zone nonlinearity
6.5.1 SLOW MOTION (SMS) AND FAST MOTION (FMS) RESPONSE

Stability conditions imposed on the fast and slow modes, and a sufficiently large mode separation rate, can ensure that the full-order closed-loop system achieves desired properties: the output transient performances are as desired and they are insensitive to parameter variations and external disturbances. In this context, the performance of the system with the FLC and the estimator was studied for both slow motion system (SMS figure 6.11) and fast motion system (FMS figure 6.12). In the SMS (FMS) the rate of change was taken as a step variation of five units for every 25 (10) samples. The performance measure is tabulated in Table 6.2.

Figure 6.11 Slow Motion System (SMS) Response
Table 6.2 PERFORMANCE MEASURE

<table>
<thead>
<tr>
<th>Type of Nonlinearity</th>
<th>Performance</th>
<th>Figure No. in thesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dead Zone</td>
<td>Phase-Plane Plot stable</td>
<td>6.8</td>
</tr>
<tr>
<td>ON/OFF</td>
<td>Phase-plane plot stable</td>
<td>6.9</td>
</tr>
<tr>
<td>Linear Dead Zone</td>
<td>Nonstable equilibrium point exists</td>
<td>6.10</td>
</tr>
<tr>
<td>Slow Motion System</td>
<td>Phase-Plane Plot stable</td>
<td>6.11</td>
</tr>
<tr>
<td>Fast Motion System</td>
<td>Residual + Fixed Tracking Delay</td>
<td>6.12</td>
</tr>
</tbody>
</table>
CHAPTER 7

INTELLIGENT CONTROLLER WITH SENSOR NOISE ELIMINATION NETWORK

A review on the current literature on sensor noise elimination algorithms, reports the existence of several adaptive noise reduction strategies. The well known, Wiener filter, (B.Widrow and S.Stearns, 1985) which is an optimum filter, belongs to this category and removes the noise in the mean square error sense. The architecture of this adaptive filter is shown in figure 7.1. In this architecture, the noise $n_1(k)$ is estimated as $\hat{n}_1(k)$ using a secondary noise source $n_2(k)$ and is then subtracted from $s(k)$ to get an estimate of the desired signal. This scheme is suitable provided

(1) $x(k)$, $n_1(k)$ and $n_2(k)$ are stationary,

(2) $n_1(k)$ and $n_2(k)$ have known statistics and

(3) $n_2(k)$ is a noise signal that is correlated with $n_1(k)$ (M.Hayes, 1996)

\[
s(k) = x(k) + n_1(k)
\]

\[
\hat{x}(k) = x(k) - \hat{n}_1(k)
\]

**Figure 7.1 Adaptive noise cancellation network for stationary $x(k)$**

* $x(k)$ is the true value of the process variable
* $n_1(k)$ is the sensor noise and $n_2(k)$ is the secondary noise source

Extended adaptive filter techniques have also been reported for situations where $x(k)$, $n_1(k)$ and $n_2(k)$ can be nonstationary and no apriori statistical information about $x(k)$, $n_1(k)$ and $n_2(k)$ is available, except for the
consideration that $n_2(k)$ and $x(k)$ are uncorrelated. This architecture is shown in figure 7.2.

![Figure 7.2 Adaptive noise cancellation network to include non-stationary $x(k)$](image)

$x(k)$ is the true value of the process variable, $n_1(k)$ is the immeasurable sensor noise and $n_2(k)$ is the secondary noise source correlated with $n_1(k)$.

These techniques, however, need a secondary noise source $n_2(k)$ to estimate $n_1(k)$. The desired signal $x_1(k)$ is extracted as before, by subtracting the estimate from the observed signal $s(k)$.

A final class of noise elimination strategies, that are ideally suited for sensor noise cancellation in a real time environment favors a noise cancellation model in which the secondary noise source $n_2(k)$ is absent, while retaining the constraints that

(i) $x(k)$ and $n_1(k)$ can be nonstationary and

(ii) no apriori statistical information is required about $x(k)$ and $n_1(k)$.

The sensor noise removal algorithm to be presented in this chapter, and to be used along with the intelligent control algorithms discussed previously, belong to this class of noise elimination technique. This network architecture is shown in figure 7.3. The delay path serves to generate a reference signal, that can be used to estimate $x(k)$. It has the advantage, that, it can remove the noise by observing the mixed signal alone. No apriori statistical information about the desired signal is required to be given as input to the network. Also, the source signal $x(k)$ need not be stationary. This motivates the use of blind signal separation (BSS) algorithms to remove the sensor noise, and extract the true value of the process from the noisy sensor readings. A neural network based Independent Component Analysis (ICA) algorithm is incorporated into the existing system to remove the sensor noise.
The proposed intelligent controller, (discussed previously) along with this ICA algorithm for sensor noise elimination, can be used to handle plant exceptions, such as

(i) poor performance of controller due to faulty sensor readings and

(ii) reduced life of actuators due to frequent false actuating signals to the control valve, to regulate the manipulated variable even when the process is at steady state. This problem arises, since, at steady state $x(k)$ is stationary but the conventional control algorithms has access only to the noisy sensor readings $s(k)$.

By using the proposed sensor noise elimination algorithm (ICA network based) at the input of each of the separately located low cost sensors, the noise rejection capability of the sensor is improved at no extra hardware costs.

### 7.1 Independent Component Analysis (ICA)

The independent component analysis consists of three steps, as shown in figure 7.4

(i) Whitening
If the observed data vectors $x_k$ have a nonzero mean, then it is desirable to remove the mean. This process is called whitening. The effects of second-order statistics can be removed by using the whitening transformation (J.F. Cardoso and B. Laheld, 1997). The whitening matrix is chosen so that the covariance matrix of the whitened vectors $\{v_k, v_k^T\}$ equals the unit matrix $I_M$. After whitening, the components of the whitened vectors $v_k$ are mutually uncorrelated with zero mean and unit variance.

Uncorrelatedness is a necessary prerequisite for the stronger independence condition, so that after pre-whitening, the noise removal task from the observed mixed data usually becomes easier. There exist infinitely many solutions for whitening the input data, provided that the number of observations of the noisy sensor readings exceeds or atleast equals the number of source signals.

7.1.1 Principal Component Analysis (PCA) based whitening algorithm

In this work, the standard PCA (C. Jutten and J.F. Cardoso et al 1997 and J. Karhunen et al 1997) based approach is used for whitening. This method has the advantage that it can simultaneously compress information optimally in the mean-square error sense and filter possible noise.
The following steps are used in the PCA based pre-whitening process:

1) Remove the mean value from the noisy sensor readings \( s[k] \).
2) Form the covariance matrix of the observed signal \( E\{s_k s_k^T\} \).
3) Extract the matrices \( D \) and \( E \), where \( D = \text{diag}[\lambda(1), \ldots, \lambda(M)] \) is a \( M \times M \) diagonal matrix, \( E = [c(1), \ldots, c(M)] \) is a \( L \times M \) matrix, \( \lambda(i) \) is the \( i_{th} \) largest eigenvalue of the data covariance matrix \( E\{s_k s_k^T\} \) and \( c(i) \) is the respective \( i_{th} \) principal eigenvector.
4) Form whitening matrix \( V \) using \( V = D^{-1/2} \cdot E^T \).
5) Pre-whiten the inputs using the transformation \( v[k] = V^* s[k] \).

### 7.1.2 Separation process

The second stage in the ICA network is the separation of the source signals. This can be achieved by using suitable higher-order statistics. The separation process can be modeled as a single-layer neural network with an equal number of input and output nodes, where the coefficient \( w_{ij} \) of the separation matrix \( W \), are simply the weights from the input to output nodes. In the present work, the noise is removed from the measured sensor readings in this stage.

Several types of separation algorithms have been in use, such as the HJ algorithms (C.Jutten, et al, 1991), PFS algorithm (J.F.Cardoso, et al, 1996), Bell’s and Sejnowski’s algorithm (A.Bell and T.Sejnowski, 1995), bigradient algorithm (L.Wang and Karhunen, 1995), nonlinear principal-component-analysis (PCA) subspace learning rule (Oja, et al, 1991), etc. The nonlinear PCA subspace learning algorithm is used in the present work, and has the advantage that it can be realized using a simple modification of the one-layer standard symmetric PCA network (J.Karhunen, et al, 1995).

#### 7.1.2.1 Neural network based separation algorithm

The pseudo-code of the algorithm is given below.

**Step 1:**

```plaintext
while(count < number of training epochs)
{
    while(i < number of observed samples)
    {
```
assign \( y[k] = W^T \ast v[k] \)
assign \( \mu = 1/(\gamma/\mu[k-1] + |v[k]|^2) \)
assign \( W[k+1] = W[k] + \mu[k] \{ v[k] - W[k] \ast g(y[k]) \} g(y^T[k]) \)
i++
}
count++
}

where \( \gamma \) is a constant called the forgetting factor and is chosen as 0.6,
\( \mu \) is the learning parameter
\( W \) is the separation matrix and
\( g(.) \) is a suitable nonlinearity and is discussed in section 7.1.3.

**Step 2:**
Estimate the source signals using \( y[k] = W^T \ast v[k] \) \((\text{number of observed samples})\)

### 7.1.3 Choice of activation function ‘\( g(.) \)’ used in the separation process

A proper choice of the activation function \( g(.) \) is important for effective removal of noise present in the sensor readings. The activation function at the output nodes is used for the training mode only and plays a central role in blind signal separation (BSS). Its nature is defined by objective or contrast or score function. The maximum likelihood approach, (H.H.Yang, 1999) defines the score function as

\[
\text{score} = \log(p_s(u_i) - p'_s(u_i))
\]

where \( p_s(u_i) \) and \( p'_s(u_i) \) are the probability density function (pdf) and its derivative, respectively, of the source signals. In this work, a **sigmoidal** activation function of the form \( \tanh(.) \) is chosen.

### 7.2 Multiple Sensor Coordination Model (with sensor noise elimination scheme and feedback sensor validation network)

The nonlinear plant with the noise added to both the inflow and out flow measuring sensors is shown in figure 7.5. The multiple sensor coordination model proposed for use in an
interacting multivariable nonlinear system, to accommodate single sensor fault and noise in the measured states, that may occur independently or simultaneously, is shown in figure 7.6. The two independent ICA networks are used to remove the noise present in the inflow and outflow sensors, while the neural network or fuzzy trained estimator takes care of the feedback sensor failure.

![Figure 7.5 Plant model with sensor noise added to inflow and outflow sensors](image)

**Figure 7.5** Plant model with sensor noise added to inflow and outflow sensors

Unknown Transfer function
Figure 7.6 Multiple sensor coordination model with the ICA network for sensor noise elimination and neural network or fuzzy trained estimator for feedback sensor validation in a multivariable interacting systems. $G_c$ is the controller and FT is the flow transmitter.
7.3 RESULTS

The focus of this section is to demonstrate the capability of the designed noise elimination algorithm, to remove the sensor noise and extract the true value of the process, independent of the ensemble statistics of a signal. The distribution of the noise model is important since it is a representative of the physical source from where the noise originated (D.Middleton, (1977), L.M.Garth and H.V.Poor, (1994) and R.J.Adler and et al, (1998)). In the present work, the sensor noise was created during each of the trial, by filtering noise samples initially drawn from a Gaussian distribution. Though the proposed algorithm is designed to remove noise without regard to a particular distribution, the noise was assumed to have evolved from a Gaussian distribution, since it is sensible in terms of the central limit theorem (A.Papoulis, 1991 and T.T.Kadota, 1988).

Samples of the nonstationary signals inflow, outflow and level of sufficiently long duration was captured for study and many trials were performed. The samples were quantised with an 8-bit resolution. The noisy sensor readings were oversampled, and presented to the noise elimination network of figure 7.3. It was assumed throughout the period of study that the autocorrelation sequence of the signal can decay to zero at a slower rate than that of the noise. This assumption is reasonable and is typical of the one that would occur in a real time process control plant. The whitening and separation algorithms (section 7.1) were used, and an estimate of the true value was obtained. The mean square error (MSE) criterion was used as a measure of performance index in each case. The normalized MSE was calculated as

\[
\text{MSE} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \hat{x}_n)^2
\]

where \( N \) is the number of samples and is chosen as 400.

7.4 PERFORMANCE OF SENSOR NOISE CANCELLATION NETWORK

7.4.1 NON CHANGING ENVIRONMENT WITH HIGH NOISE

The following data and figure 7.7 show the variation of input signal and compares it with the average estimator output for a non changing environment. Since the simulations are of a high noise environment, the graph shows that the estimator takes a longer time for tracing. S1,S2,S3 is noise variance and T represents time at every 25 sample intervals.

Noise variance
The values between these time intervals remain correct even if a certain sensor fails in between. This is because, the average of other sensors are taken into computation.

![Graph](image_url)

**Figure 7.7 Non changing high noise environment plot**

### 7.4.2 NON CHANGING ENVIRONMENT WITH LOW NOISE

The following data and figure 7.8 show the variation of input signal and compares it with the average estimator output for a non changing environment. Since the simulation is of a low noise environment, the graph shows the estimator takes a shorter time for tracing.

**Noise variance**
7.4.3 CHANGING ENVIRONMENT WITH LOW NOISE

The following data and figure 7.9 show the variation of input signal and compares it with the average estimator output for a changing environment. Since the simulation is of a low noise environment, the graph shows the estimator takes a shorter time for tracing. It is also important to note the way the filter adapts to changing environment immediately without needing too many samples to track.
Noise variance of sensor
S1: 10
S2: 12
S3: 10

Estimator Response
T1  98.639648
T2  99.313988
T3  99.773521
T4  99.555588
T5  54.916523
T6  53.093877
T7  52.301666
T8  51.719868
T9  120.38886
T10 121.99090
T11 122.92529
T12 123.30020
T13  74.036057
T14  74.373108
T15  74.780304
T16  74.690163
T17  151.89450
T18  151.00962
T19  150.75369
T20  150.59066
7.4.4. CHANGING ENVIRONMENT WITH HIGH NOISE

The following data and figure 7.10 show the variation of input signal and compares it with the average estimator output for a changing environment. Since the simulation is of high noise environment, the graph shows the estimator takes a longer time for tracing. It is also important to note the way the estimator adapts to changing environment immediately without needing too many samples to track.

**Noise variance of sensor**
- S1: 20
- S2: 25
- S3: 15

**Estimator Response**
- T1: 100.31990
- T2: 100.64579
- T3: 100.30774
- T4: 100.43841
- T5: 42.013359
7.5 DISCUSSION ON THE PERFORMANCE OF THE SENSOR NOISE ELIMINATION NETWORK
The proposed sensor noise elimination scheme was implemented and tested for satisfactory performance on the real time plant. The true sensor readings were corrupted with Gaussian noise, and presented to the sensor noise elimination network. The estimated value is compared with the true value, by computing the mean square error deviation (MSE). From the obtained MSE value, it can be concluded that the estimated values matches well with the true values of the measured states. In particular, the ability of the network to process the noisy level sensor readings and give a good estimate of the true value during the servo tracking response is significant.

The proposed sensor noise elimination algorithm can be incorporated with ease, into any of the existing process control loops, at no extra hardware costs. This permits the use of a commercially available low cost sensor, with the sensor noise elimination taken care by the proposed algorithm. The algorithm can be implemented easily using the unsupervised neural learning architecture. The algorithms are designed to remove the sensor noise in a manner ideally suited for a real time environment, where the sensor noise elimination process need to be performed independent of the ensemble statistics of the signal without regard to a particular class of signal model. The MSE values obtained for the different cases, is summarized in table 7.1 to table 7.4.

However, it should be mentioned here, that in the area of blind signal separation, the theoretical properties, range of applicability and mutual comparisons still remains largely unexplored.

<table>
<thead>
<tr>
<th>Type of control</th>
<th>Set point tracking</th>
<th>Outflow change from 50% to 40%</th>
<th>Outflow coefficient change from 0.5 to 0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant parameter PI</td>
<td>7.88</td>
<td>43.44</td>
<td>4.33679x10^6</td>
</tr>
<tr>
<td>Fuzzy controller</td>
<td>2.663</td>
<td>6.186</td>
<td>0.99954x10^6</td>
</tr>
<tr>
<td>Adaptive PI</td>
<td>2.734</td>
<td>9.641</td>
<td>0.99954x10^6</td>
</tr>
</tbody>
</table>
Table 7.2 Normalized MSE

<table>
<thead>
<tr>
<th>Type of estimator/controller</th>
<th>Set point tracking</th>
<th>Regulatory response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy estimator with fuzzy controller</td>
<td>402.83</td>
<td>402.83</td>
</tr>
<tr>
<td>Neural estimator with fuzzy controller</td>
<td>404.37</td>
<td>404.37</td>
</tr>
</tbody>
</table>

Table 7.3 Normalized MSE values

<table>
<thead>
<tr>
<th>Sensor to which noise is added</th>
<th>Without noise elimination network</th>
<th>With noise elimination network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level sensor (servo tracking)</td>
<td>0.985</td>
<td>0.3345</td>
</tr>
<tr>
<td>Level sensor (Regulatory response)</td>
<td>1.0015</td>
<td>0.0461</td>
</tr>
</tbody>
</table>

Table 7.4 Normalized ISE values

<table>
<thead>
<tr>
<th>Sensor to which noise is added</th>
<th>With noise elimination network</th>
<th>Without noise elimination network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow</td>
<td>0.0503</td>
<td>1.045</td>
</tr>
<tr>
<td>Outflow</td>
<td>0.3522</td>
<td>0.92799</td>
</tr>
</tbody>
</table>
### 7.6 HARDWARE IMPLEMENTATION RESULTS

![Figure 7.11 Initial tracking of Neural estimator](image)

<table>
<thead>
<tr>
<th>t_v_aud3k</th>
<th>58.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_v_aud5a2</td>
<td>4</td>
</tr>
<tr>
<td>t_v_aud7a2</td>
<td>67</td>
</tr>
<tr>
<td>t_v_aud12</td>
<td>177</td>
</tr>
<tr>
<td>t_v_aud2</td>
<td>310</td>
</tr>
<tr>
<td>t_v_audcount</td>
<td>469</td>
</tr>
<tr>
<td>t_v_aud1</td>
<td>8</td>
</tr>
<tr>
<td>t_v_aud18</td>
<td>62.33</td>
</tr>
<tr>
<td>t_v_aud2</td>
<td>55.23</td>
</tr>
<tr>
<td>t_v_aud3</td>
<td>87.1</td>
</tr>
</tbody>
</table>

**Figure 7.11 Initial tracking of Neural estimator**

- **t_v_aud3k**: 58.1
- **t_v_aud5a2**: 4
- **t_v_aud7a2**: 67
- **t_v_aud12**: 177
- **t_v_aud2**: 310
- **t_v_audcount**: 469
- **t_v_aud1**: 8
- **t_v_aud18**: 62.33
- **t_v_aud2**: 55.23
- **t_v_aud3**: 87.1

The chart shows the initial tracking results of the Neural estimator, indicating the performance metrics over different time intervals.
Figure 7.12 Neural estimator settling to first environment value and successfully tracking first change in environment
Figure 7.13 Neural estimator settling early after first change.
CHAPTER 8

CONCLUSION AND FUTURE SCOPE

In this work, estimates of the primary output (the process variable change in level) are generated at the (faster) sampling rate of the secondary outputs and inputs, while adaptation occurs only at the (slower) sampling rate of the primary variable, or whenever the primary variable becomes available. Thus the estimator serves simply as a software based sensor (soft sensor). Future direction of study shall focus on obtaining the degree of time scale separation between SMS and FMS and accordingly tune the control law. In this research, signals are processed in real time and combined with previous monitoring data to estimate, using the neural network, the process variable level in a nonlinear process control plant. Neural Estimator and Fuzzy Estimator are designed for hopper type tank process. Experimental results were carried out for servo and regulatory problems, and improved results were obtained with Neural Estimator.

8.1 CONCLUSION

In this thesis, techniques that can be used to improve the operational range of a plant with minimum human intervention and at no extra hardware costs are presented. The design algorithms and techniques presented here, focuses on providing additional information about a plant, which can be used by the control algorithms for improved control. A rule-based real time intelligent controller with an estimator, designed by coordinating the data from multiple sensors, (that are commonly present in a process plant for monitoring various states and environmental conditions) is used, to handle exceptions such as sensor faults or extreme situations incorrectly handled by the less sophisticated conventional controllers.

In most of the previous analogous works surveyed so far, some of the factors that should have been given more emphasis in the design of an intelligent system, (particularly for a process plant) has been found to be lacking. For example, most of the real time processes are nonlinear, and has mutually interacting process variables with a vast amount of highly correlated data, and this good correlation need to be exploited in the design of intelligent system algorithms. Similarly, in a real
time environment, the measured states are noisy, and only these noisy sensor readings and nothing else, is available for processing.

In this work, the intelligent control algorithms have been designed and tested on a real time nonlinear plant for satisfactory performance, with due emphasis to include all the above missing factors. A multiple sensor coordination model, integrating the data from two or more independently designed, mutually interacting and separately located sensors is developed. It can be incorporated into any existing interacting multivariable process control loops with slight modifications in the algorithm, at no extra hardware costs. The fact that no extra hardware costs are involved implies that, with the proposed algorithms, the existing configuration need not be altered, but less sophisticated controllers can be made to perform more complicated tasks than it is intended to. The algorithms presented here can be used to assist, existing non-fault-tolerant conventional control strategies. Two types of sensor validation algorithms were proposed utilizing the vast amount of highly correlated data available, which is true in most of the industrial processes. A nonlinear (hopper type tank) multivariable interacting (level and flow) process was chosen for testing the different algorithms. The simulated and experimental servo and regulatory responses were obtained for different operating conditions and the ISE criterion was used as a measure of performance index. Comparison study between the simulated and experimental responses is also presented.

8.2 FUTURE SCOPE

Future direction of study shall focus on embedding the proposed control algorithms into hardware units for online use. Also, the intrinsic use of sensor noise control strategies can give improved results in real time implementation.
APPENDIX

DEAD-ZONE AND SATURATION

\[
\begin{align*}
Y &= K(x - D/2) ; 0 \leq \omega t \leq \alpha \\
&= K(S - D/2) ; \beta \leq \omega t \leq (\pi - \beta) \\
&= K(x - D/2) ; (\pi - \beta) \leq \omega t \leq (\pi - \alpha) \\
&= 0 ; (\pi - \alpha) \leq \omega t \leq \pi
\end{align*}
\]

where \( \alpha = \sin^{-1} \frac{D}{2X} \) and \( \beta = \sin^{-1} \frac{S}{X} \)

\[x(t) = x \sin \omega t\]

at \( \omega t = \alpha \); \( x(t) = D/2 \)

\[D/2 = x \sin \alpha \quad \text{or} \quad \alpha = \sin^{-1} \frac{D}{2X}\]

\[x(t) = x \sin \omega t\]

at \( \omega t = \beta \); \( x(t) = S \)

\[S = x \sin \beta \quad \text{or} \quad \beta = \sin^{-1} \frac{S}{X}\]

\[B_1 = 0 \]

\[\frac{\pi}{2}\]

\[A_1 = \frac{4}{\pi} \int y \sin \omega t \, d(\omega t)\]

\[0\]

\[= \left( K/\pi \right) \left[ 2X(\beta - \alpha) - X(\sin 2\beta - \sin 2\alpha) + 4[(D/2)(\cos \beta - \cos \alpha) + (S - D/2) \cos \beta] \right] \]

\[= \left( KK/\pi \right) \left[ 2(\beta - \alpha) + (\sin 2\beta - \sin 2\alpha) \right] \]

Therefore the describing function is given by

\[\begin{cases}
0 & ; X < D/2; \alpha = \beta = \pi/2 \\
\end{cases}\]
\[
K_N(X)/K = 1 - (2/\pi)(\alpha + \sin \alpha \cos \alpha) \quad ; \quad D/2 < X \leq S; \beta = \pi/2
\]

\[
(1/\pi)[2(\beta - \alpha) + (\sin 2\beta - \sin 2\alpha)] \quad ; \quad X > S
\]

Two special cases immediately follow

**Case I: Saturation nonlinearity** \((D/2 = 0, \alpha = 0)\)

\[
K_N(X)/K = \begin{cases} 
1 & ; \quad X < S \\
(2/\pi)(\beta + \sin \beta \cos \beta) & ; \quad X > S \\
= (2/\pi)[\sin^{-1}S/X + (S/X) \sqrt{1 - (S/X)^2}] & ; \quad X > S 
\end{cases}
\]

**Case I: Dead-zone nonlinearity** \((S \to \infty; \beta = \pi/2)\)

\[
K_N(X)/K = \begin{cases} 
0 & ; \quad X < D/2 \\
- (2/\pi)(\alpha + \sin \alpha \cos \alpha) & ; \quad X > D/2 \\
= 1 - (2/\pi)[\sin^{-1}(D/2X) + (D/2X) \sqrt{1 - (D/2X)^2}] & ; \quad X > D/2 
\end{cases}
\]

**RELAY WITH DEAD ZONE AND HYSTERESIS**

\[
Y = \begin{cases} 
0 & ; \quad 0 \leq \omega t \leq \alpha \\
+M & ; \quad \alpha \leq \omega t \leq \beta(\pi - \beta) \\
0 & ; \quad (\pi - \beta) \leq \omega t \leq (\pi + \alpha) \\
-M & ; \quad (\pi - \alpha) \leq \omega t \leq (2\pi - \beta) \\
0 & ; \quad (2\pi - \beta) \leq \omega t \leq 2\pi 
\end{cases}
\]

where \(\alpha = \sin^{-1}D/2X\) and \(\beta = \sin^{-1}(D - 2H)/2X\)

\[
\pi
\]

\[
B_1 = (2/\pi) \int_0^\pi \cos \omega t \, d(\omega t)
\]

0

\[
\pi - \beta
\]
\[
\frac{1}{2} \int M \cos \omega t \, d(\omega t) = (2M/\pi)(\sin \beta - \sin \alpha) = (2M/\pi)(-H/X)
\]
\[
A_3 = (2/\pi) \int y \sin \omega t \, d(\omega t)
\]
\[
\pi - \beta
\]
\[
= (2/\pi) \int M \sin \omega t \, d(\omega t) = (2M/\pi)(\cos \alpha + \cos \beta) = (2M/\pi)\left[\sqrt{1-(D/2X)^2} + \sqrt{1-(D - 2H/2X)^2}\right]
\]
\[
\alpha
\]

Therefore
\[
K_N(X) = \begin{cases} 
\sqrt{[A_3/X]^2 + [B_3/X]^2} \tan^{-1} B_3/A_3 & ; X < D/2 \\
0 & ; X > D/2
\end{cases}
\]

**I. Ideal relay:**

Letting \(D = H = 0\) in eqn. (1),

\[
K_N(X) = 4M/\pi X
\]

**II. Relay with dead-zone:**

Letting \(H = 0\) in eqn. (2)

\[
K_N(X) = \begin{cases} 
0 & ; X < D/2 \\
(4D/\pi X)\sqrt{1-(D/2X)^2} & ; X > D/2
\end{cases}
\]

**III. Relay with hysteresis:**

Letting \(H = D\) in eqn. (1),

\[
K_N(X) = \begin{cases} 
0 & ; X < D/2 \\
(4H/\pi X) \sin^{-1} (H/2X) & ; X > H/2
\end{cases}
\]

**BACKLASH**

\[
x - b/2 \quad ; 0 \leq \omega t \leq \pi/2
\]
\[
\begin{cases}
X - b/2 & ; \pi/2 \leq \omega t \leq (\pi - \beta) \\
Y = x + b/2 & ; (\pi - \beta) \leq \omega t \leq 3\pi/2 \\
-x + b/2 & ; 3\pi/2 \leq \omega t \leq (2\pi - \beta) \\
x - b/2 & ; (2\pi - \beta) \leq \omega t \leq 2\pi
\end{cases}
\]

where \( \beta = \sin^{-1} (1 - b/X) \);

\[
B_1 = \frac{-X/\pi \cos^2 \beta}{\sin \beta}
\]

\[
A_1 = \frac{X/\pi}{\sin \beta} [\frac{\pi}{2} + \beta + \frac{1}{2} \sin 2\beta]
\]

Therefore,

\[
K_N(X) = \begin{cases}
0 & ; X < b/2 \\
\sqrt{\left\{A_1/X\right\}^2 + \left\{B_1/X\right\}^2} \tan^{-1} B_1/A_1 & ; X > b/2
\end{cases}
\]
REFERENCES


• Hong, Y., Yang, O.W.W., 2006, ‘Design of an adaptive PI rate controller for streaming media traffic based on gain and phase margins Communications’, *IEE Proceeding*, vol.153, no.1, pp. 5 - 14

• Huo, B. 2012, ‘Fuzzy adaptive fault-tolerant output feedback control of multi-input and multi-output non-linear systems in strict-feedback form’, *Control Theory & Applications, IET*, vol.6, no.17, pp.2704-2715


• L. S. Iliadis, 2005. ‘A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation,’ Environmental Modelling & Software. vol. 20, no.5, pp 613-621.


• Tavazoei, M.S. 2012, ‘From Traditional to Fractional PI Control: A Key for Generalization From Traditional to Fractional PI Control: A Key for Generalization’, Industrial Electronics Magazine, IEEE, vol.6, no.3, pp. 41 - 51


• Zhao F., Ou J. and Du W. 2000, ‘Pattern-based fuzzy predictive control for a chemical process with dead time’, *Engineering Applications of Artificial Intelligence*, vol. 13, pp. 37-45

LIST OF PUBLICATIONS

