

1-1-2006

A computer simulator for steel plant electrical arc furnace regulator

Behzad (George) Jorjani
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Heat Transfer, Combustion Commons](#)

Recommended Citation

Jorjani, Behzad (George), "A computer simulator for steel plant electrical arc furnace regulator" (2006). *Theses and dissertations*. Paper 458.

A COMPUTER SIMULATOR FOR STEEL PLANT ELECTRICAL ARC FURNACE REGULATOR

By
Behzad (George) Jorjani
Bachelor of Electrical Engineering
Shiraz University
Shiraz, Iran, 1992

A thesis
Submitted to Ryerson University
in partial fulfillment of the
requirement for the degree of

Master of Applied Science

in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2006

©Behzad (George) Jorjani 2006

UMI Number: EC53862

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform EC53862
Copyright 2009 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Author's Declaration

I hereby declare that I am the sole author of this thesis.



I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature



I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature



Abstract

A COMPUTER SIMULATOR FOR STEEL PLANT ELECTRICAL ARC FURNACE REGULATOR

Behzad (George) Jorjani, MAsC.

Department of Computer and Electrical Engineering
Ryerson University
Toronto, Ontario, Canada
Year 2006

The function of the simulator is to imitate the behavior of the regulator loop, which is the main component of the Electrical Arc Furnace (EAF) control systems. In the past, the use of artificial intelligence methods, and in particular, the Adaptive Neuro Fuzzy Inference System (ANFIS) were successfully applied in the modeling and control of the EAF components individually. This research expands the use of ANFIS in building the full closed loop computer simulator for the three-phase regulator loop. The ANFIS models inputs and outputs selected for this project were tried for the first time in this research. The simulator components were trained and verified by the use of plant recorded data in the open loop mode. The response of the closed loop simulator was tuned to follow the behavior of the plant EAF. Therefore the simulator works independent of the plant data or operation commands. The developed simulator, then, was used to measure the results of applying new controls in EAF such as fuzzy controllers, without disturbing the actual plant process.

Acknowledgments

First of all, I am grateful to my research supervisor Dr. Farrokh Sharifi for all his support and leadership in duration of this research. I would like to thank him for his understanding, and sharing of his knowledge and helpful advices through my graduate program.

Furthermore, I would like to thank the management in Gerdau Ameristeel Whitby (GAW) plant for their significant supportive role in this research. My special thanks to Mr. Wagner DaSilva, the manager of engineering department in GAW, for his encouragement and support during my graduate studies. I would also like to thank Mr. Dave Burn, the senior electrical engineer of the plant, for sharing his vast process expertise in this research. Moreover, I would like to thank Mahin and Shahriar in Can-Technologies for their assistant in the data collection phase of this research. As well, I would like to thank Materials and Manufacturing Ontario (MMO) for providing the grant for this research.

Our research team in Robotic and Manufacturing Automation Lab (RMAL), in particular Dr. Iraj Hassan Zadeh, provided the required technical assistance during the lab work of the research. As well, many thanks to my sister-in-law Mitra for her inputs and comments.

Both of my parents are retired teachers. They have placed significant emphasis on educational achievements and made that a priority in my life. I would not have made it here without their support and encouragement through my life.

Last, but certainly not the least, my dear wife Maryam has been incredible help and given her hundred percent support during my masters program. She was not only the mother but also a father for our children for the duration of my studies.

Dedications

I would like to dedicate this thesis to my wonderful wife Maryam and our two beautiful daughters, Tara and Telma.

Contents

Chapter 1 Introduction	1
1.1 The Process of EAF.....	2
1.2 EAF Regulator Control.....	4
1.3 Objective and motivations.....	7
1.4 Previous Work.....	7
1.5 The Research Contribution.....	10
1.6 Thesis Outline.....	10
Chapter 2 The Background Theory.....	12
2.1 Fuzzy Logic.....	13
2.1.1 Fuzzy Sets	13
2.1.2 Fuzzy Operations	14
2.1.3 Fuzzy Rules and Reasoning	15
2.1.4 Fuzzy Inference Systems	16
2.2 Fuzzy PID Controllers.....	20
2.3 Optimization.....	25
3.3.1 System Identification	25
2.3.2 Least-Square Estimator.....	26
2.3.3 Gradient Based Optimization Method.....	28
2.4 Adaptive Networks.....	29
2.4.1 Backpropagation	30
2.5 ANFIS.....	33
2.5.1 The Hybrid Learning Method	36
2.5.2 C-Means Clustering	37
2.6 MATLAB® ANFIS Toolbox.....	39
Chapter 3 Data Collection	43
3.1 Data Acquisition System.....	43
3.2 Collected Data Items.....	44
3.3 The Process Analysis Based on the Collected Data.....	46
Chapter 4 The Simulator Structure and Implementation.....	51
4.1 The Overview of the Structure and Components	51
4.2 Measured Values ANFIS Models	52
4.3 The Simulated Regulator Control	56
4.4 Power Consumption Calculation.....	58

4.5 The Discrete Event Generator	60
Chapter 5 Model Verification, Results and Discussions.....	64
5.1 Model Verification Procedure	64
5.2 Verification Process for the Simulator Components	64
5.3 Model Outputs Graphs.....	67
5.4 The Closed Loop Simulator Graphs	70
5.5 Replacing the Classic Adaptive PI Controller with a Fuzzy Gain Scheduling PID	72
Chapter 6 Conclusion and Future Work.....	75
6.1 Conclusion	75
6.2 Future Work.....	76
Bibliography	77
Publications.....	81
Appendix A: ANFIS models inputs, outputs and parameters.....	82

List of Figures

Figure 1: The schematic diagram for the power components in EAF.	3
Figure 2: The control schematic for the EAF regulator loop.....	6
Figure 3: Example of membership functions.....	14
Figure 4: Fuzzy inference process with two inputs and one output.....	17
Figure 5: The nonlinear mapping of input and output space.	18
Figure 6: The Fuzzy Gain Scheduling PID Controller.	22
Figure 7: Membership functions for error and changes on error signals.....	23
Figure 8: Membership functions for proportional and derivative coefficients.	23
Figure 9: Singleton membership functions for linear coefficient to relate integral and derivative time constants.....	24
Figure 10: A typical feedforward adaptive network.	30
Figure 11: An example of ANFIS general structure (Figure with slight modification has been borrowed from [36]).	35
Figure 12: MATLAB® ANFIS GUI.	40
Figure 13: The typical FIS structure generated by MATLAB® ANFIS toolbox.....	42
Figure 14: The FIS rule viewer generated by MATLAB® ANFIS toolbox.....	42
Figure 15: The data acquisition system network diagram.	44
Figure 16: Energy consumption during the melting process.	46
Figure 17: Voltage signals' behaviors during the melting process.....	48
Figure 18: Current and power factor signals behaviors during the melting process.	49
Figure 19: The graph of the electrode movement during the melting process.	50
Figure 20: Components of the simulator for one of the phases and their interconnections...	53
Figure 21: ANFIS models for EAF measured signals.	54
Figure 22: Phase 1B measured active current ANFIS model.	55
Figure 23: Phase 1B measured reactive current ANFIS model.	55
Figure 24: Phase1B measured voltage ANFIS model.	56
Figure 25: Phase 1B regulator Simulink® block overview.	57
Figure 26: Phase 1B regulator PI controller.	57
Figure 27: Power factors calculation block.	59
Figure 28: Active power calculation.....	60
Figure 29: Sequential event simulator.	62
Figure 30: Voltage reference lookup table for the first charge.	63
Figure 31: Voltage reference lookup table for the second charge.	63
Figure 32: Model verification average error bar graphs.	66
Figure 33: Model verification, RMS error bar graphs.	67
Figure 34: Active current, model output and plant measured value graphs.	68
Figure 35: Reactive current, model output and plant measured value graphs.....	69
Figure 36: Phase voltage, model output and plant measured value graphs.	69
Figure 37: Regulator output, model output and plant measured value graphs.	70

Figure 38: The closed loop simulator results without using the plant data.	71
Figure 39: Comparing the regulator output with classic PI and GS Fuzzy PID controller. ...	73
Figure 40: Comparing total melting time between regulator uses a classic PI and GS Fuzzy PID controller.....	74

List of Tables

Table 1: The threephase voltage reference lookup table. 6

Table 2: An example of fuzzy rules..... 18

Table 3: PID Gain Scheduling rules 24

Table 4: The important collected plant signals. 45

Table 5: The error summary for the set of the training data 65

Table 6: The error summary for the set of the verification data 65

Nomenclature

Acronyms:

B	Big
COA	Centriod Of Area
EAF	Electrical Arc Furnace
FCM	Fuzzy C-Means clustering
FIS	Fuzzy Inference Systems
GAW	Gerdau Ameristeel Whitby
LSE	Least Square Estimator
MF	Membership Function
MMO	Materials and Manufacturing Ontario
NB	Negative Big
NM	Negative Medium
NS	Negative Small
PB	Positive Big
PM	Positive Medium
PS	Positive Small
PID	Proportional Integral Derivative
PLC	Programmable Logic Controller
RBFN	Radial Based Function Network
S	Small
SPLC	Smart Predictive Load Control
ZO	Zero

Variables:

A, B, C	Fuzzy set
C_i	Center of group i
d_k	Desired output for element k of training data
$e(k)$	Error signal at discrete time k
E_p	Training error (cost function)
f_i	Sugeno consequent of rule linear polynomial
$g(\theta)$	Gradient of the cost function
G_i	Group i
I	Current
K_p	Proportional coefficient
K_i	Integral coefficient

K_d	Derivative coefficient
L	Number of layers
l	Layer index
m	Number of training data pairs
n	Number of system parameters
$O_{l,i}$	Output of node i in the layer l
P	power
PF	power factor
p, q, r	Sugeno consequent of rule linear polynomial coefficients
T_d	Derivative time constant
T_i	Integral time constant
T_s	Sampling time constant
$u(k)$	System output at discrete time k
V	Voltage
w_i	Rule strength of rule number i
x	Member of input space
$x_{L,k}$	Network actual output for element k of training data
X	Universe of discourse
y	Model output
Y	Actual system output
Z	Universe of discourse

Symbols:

μ	Membership function
α	The conversion for integral and derivative time constants
$\Delta e(k)$	The delta of error signal
θ	System parameter
$\hat{\theta}$	Optimum system parameter
θ_{now}	System parameter in this round of iteration
θ_{next}	System parameter in next round of iteration
η	Step size or learning rate
$\nabla E(\theta)$	Gradient of the cost function
α, β, γ	Node parameters
$\varepsilon_{l,i}$	Derivative of cost function with respect to output of node i in layer l

Chapter 1

Introduction

The Electrical Arc Furnace (*EAF*) is the main equipment in the recycling process of steel plants. EAF is usually among the highest electrical energy consumers in the power grid. The rising cost of energy puts pressure on the steel industry to improve their processes and conserve energy. A significant factor in energy consumption is optimizing the control strategy in the EAF. Practicing poor control methods in the EAF could be quite costly, while also damaging to people or equipment. Thus, in controlling the EAF, having a good computer model provides a great advantage in verifying and tuning the proposed solutions in the simulator environment.

The EAF includes many processes, and building the full simulation of the EAF to include all the processes and auxiliary systems is a very difficult task. This research is an attempt to build the MATLAB[®]-based computer simulator for the regulator loop of the EAF in Gerdau Ameristeel Whitby (GAW). The regulator loop will be explained in detail throughout this thesis. The regulator loop performance has a direct impact on the electrical consumption of the EAF. Hence, it is selected to be the focus of this research and also the focus of many other studies in the past. In this research, the ANFIS black box modeling method has been used to model the behavior of the most important measured signals from the plant. For the purpose of a quick implementation, the MATLAB[®] ANFIS GUI has been used to build the fuzzy structures needed for the project. Additionally, the entire simulator has been built in MATLAB[®] Simulink[®].

1.1 The Process of EAF

In this section, several technical terms are explained which are utilized throughout this report. EAF is the main equipment in any steel plant that recycles used steel – also known as *scrap* metals. It consists of a metal segmented body (shell) that is covered by the thick layer of refractory. The shell is closed from the bottom and has a roof that could be raised, lowered and swung by the use of hydraulic cylinders. As well, the entire furnace rests from one side on the large hydraulic cylinder that could tilt the furnace close to 45 degrees. On the opposite side of the tilt cylinder at the bottom of the EAF, there is a tap hole that serves to discharge the molten steel into the ladle when the EAF is tilted. The EAF roof has three holes in a triangular shape around its center to allow the graphite electrodes to ascend and descend. The electrodes are moving up/down individually by the use of hydraulic cylinders. The graphite electrodes are clamped into the arm connected to the power logs. The three power logs carry the voltage from the transformer cable to the electrodes. All the mechanical elements of EAF, including the furnace shell, roof, electrode clamps, power logs and cables, are water-cooled.

The electricity from the power utility with the high voltage level of 230 KV is transferred through the hydro grid into the plant. Then, the plant transformer brings the voltage level to 46 KV. Subsequently, another transformer close to the EAF, which is known as the *Furnace-Transformer*, brings the voltage to the lower level of 200-1650 V. In order to vary the voltage supply to the electrodes in different stages of melting, the furnace transformer always comes with a mechanical tap changer. Depending on the transformer design, the tap change could be done under load or off load. In the under load tap changing, due to the arcs, the tap changes are usually high in maintenance. Nonetheless, they save time in the process of melting. It takes anywhere between 300-1000 KVA of apparent power in the transformer to melt one ton of scrap. The graphite electrodes carrying the three phase voltage into the scrap pile are consumable material. There are three electrodes, one for each phase. The positions of the electrodes are controlled by the use of the hydraulic cylinders. The control signals used to adjust the direction and speed of the electrodes' movements are the heart of the EAF control system. The electrode movement controls the actual voltage (in the regulated current

system), as well as the electrical arc length between the electrodes and scrap pile. The flow of the hydraulic fluid into and from the electrode cylinders is controlled by the use of the proportional valves. The control signals sent to the hydraulic proportional valves of the three phases come from the *Regulator Loop*. The regulator loop is explained with more details in the next section. The schematic diagram from the power and control components of EAF is illustrated in Fig. 1.

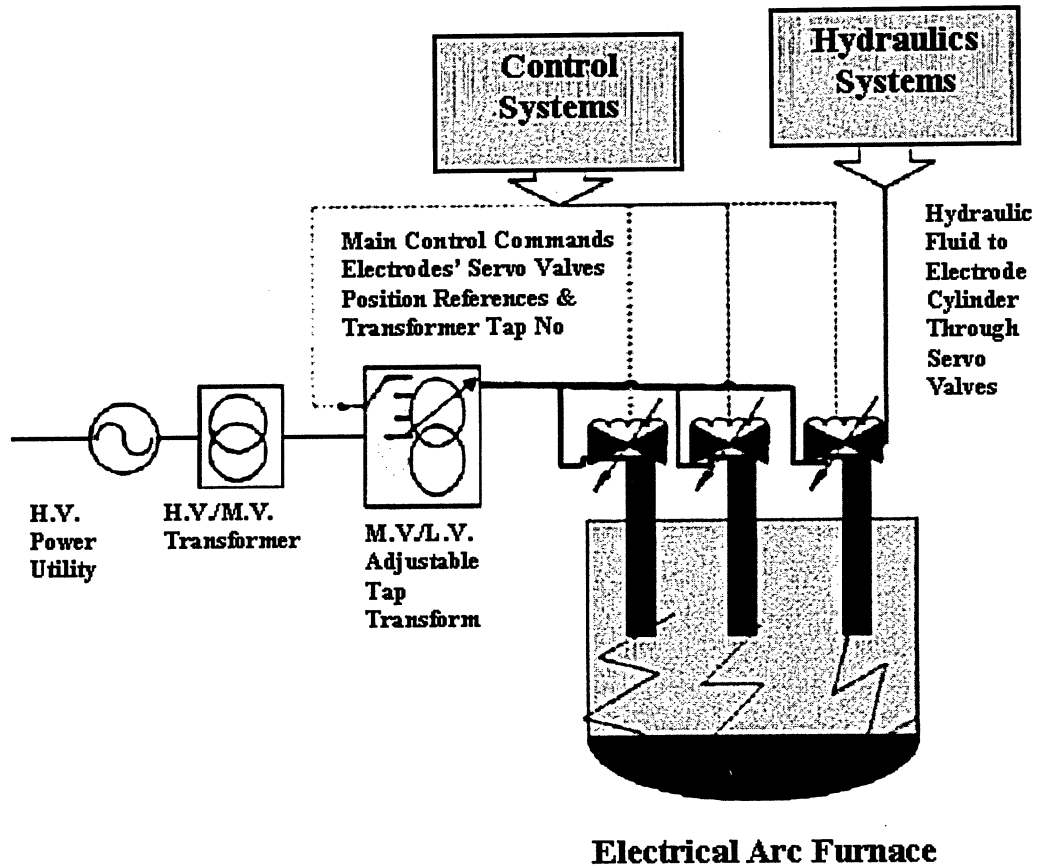


Figure 1: The schematic diagram for the power components in EAF.

The scrap metals occupy more volume in comparison to liquid molten steel. Therefore, to maximize the tonnage capacity of the EAF, the scrap metals are usually dumped into the EAF in two batches. Each batch is known as a *Charge*. The idea is to melt the first Charge and make room to receive more scrap metals in the second Charge. As well, in the second Charge, the chemistry of steel is made based on the desired steel grade, and then the batch

moves to the next process. One run of the molten steel with two or more charges is known as a *Heat*. As a result of variations in metal scrap types and sizes, and the differences in the mixed alloys, even the two Heats of steel made based on the same recipe are never identical in terms of metallurgical properties. For this reason, each Heat will be tracked through the rest of the steel-making and rolling process for quality control purposes.

Although electricity is the main form of energy for generating heat in the EAF, it is important to know that chemical energy contributes significantly to the melting process. The chemical energy comes from two different sources: *Burners* and *oxidization reactions*. The burners are located around the furnace shell. The flame, which is the result of combusting natural gas, would help melting the scrap metals around the burners. The flow of the gas and oxygen are controlled by using the flow control valves. Another source of chemical energy is the heat generated by oxidization reactions. The different impurities, carbon and lime, in the scrap react with the oxygen in different temperatures and generate heat. The heat generated by the burners and oxidization is known as *Alternative Energy* to electrical energy.

During the melting process, Carbon and other alloys are dumped into the EAF to create a desired grade of steel. Carbon and oxygen are also injected into the EAF from the burners to refine the impurities in a steel batch. By means of injecting oxygen and carbon, a layer known as Foamy Slag is formed on top of the molten steel. Due to heat isolation and other properties, forming the right thickness of *Foamy Slag* is a critical part of the process. Although the formation of Foamy Slag and the Alternative Energy control are very important, they are not the interest of this research. We hope to study those in future projects.

1.2 EAF Regulator Control

The main task of the control system inside the EAF is to position the electrodes in the process of melting. This task in the EAF is performed by the part of the system that is traditionally called the Regulator Loop. As previously mentioned, the arcs generated from

graphite electrodes carrying voltage (200-1200V) are the main source of the heat generation in the EAF. In the EAF, which is installed in GAW with the use of Smart Predictive Load Control (SPLC), the supply of current to the electrodes is regulated to be kept below the adjustable high limit. Thus, the EAF could always work with the same transformer tap and a constant voltage supply. On the other hand, the impedance inside the furnace and during the meltdown is always variable. Since the current is always kept limited by the SPLC, positioning the electrodes is the only way of achieving the voltage reference (set point). Each electrode is utilized with a servo valve that could get a ± 10 V as a reference set-point. Electrodes go up or down depending on the polarity of the voltage reference and with the speed proportional to the magnitude of voltage reference. Fig. 2 shows the PI control schematic for the regulator loop. The PI control in this case is a classic adaptive. This means that the proportional and integral coefficients are changing slightly through the different stages of the meltdown. The proportional and integral coefficients are stored in the lookup table, and the meltdown percentage is used as the pointer to them.

Furthermore, the reference set-points for the current and voltage changes in different stage of the melting process. The existing control system uses a lookup table, built based on the experience of the power engineer of the plant. Again, the meltdown percentage is the pointer to this lookup table. The meltdown percentage is calculated according to the total weight of scrap charged into the furnace and the assumed KWh amount that it takes to melt every ton of the scrap metal. Then, during the process, the actual KWh is measured and translated to melt down percentage dynamically. Table 1 shows the example of the data in the regulator program lookup table per phase. In this table every step is indicated by the number and also a name.

At present, the set-points of the voltage and current for all of the phases are the same. With the use of the new version of SPLC, in the near future, the control system would be able to vary each phase current reference independently. Thus, the regulator is programmed to allow individual reference set-points for each phase.

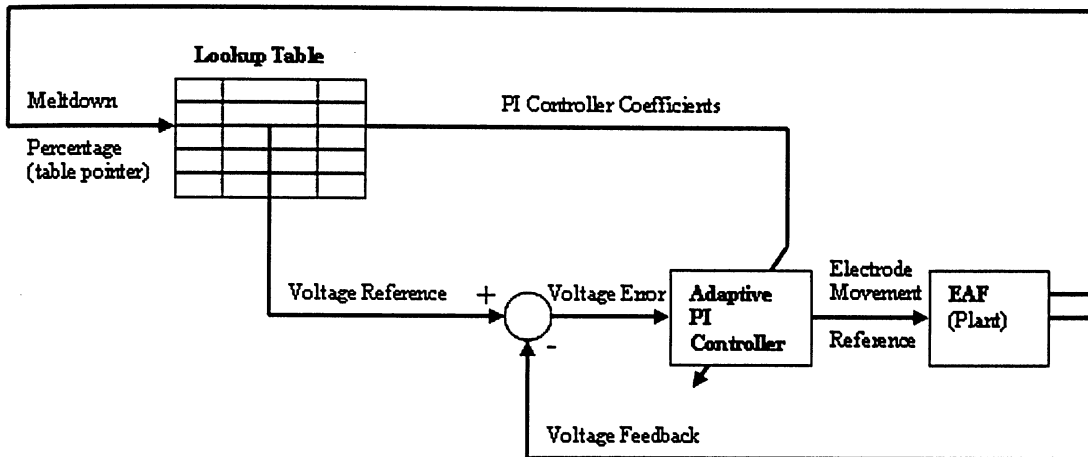


Figure 2: The control schematic for the EAF regulator loop.

Table 1: The threephase voltage reference lookup table.

Step	Step name	Phase1		Phase2		Phase3	
		Volts Value	KAmps Value	Volts Value	KAmps Value	Volts Value	KAmps Value
0	Soft Start	200	49	200	49	200	49
1	Deep Bore in	450	53	450	53	450	53
2	Meltdown 1	535	55	535	55	535	55
3	Meltdown 2	560	55	560	55	560	55
4	Meltdown 3	540	55	540	55	540	55
5	Refining 1	535	55	535	55	535	55
6	Refining 2	535	55	535	55	535	55
7	Manual Tap-out	530	55	530	55	530	55

1.3 Objective and motivations

The objective of this thesis is: to achieve the full closed loop simulator of the EAF regulator loop in Matlab® Simulink® environment. The developed simulator serves as a trail platform for evaluating the control improvement scenarios before they are tried in the real EAF.

The main motivations for this research are:

- 1- EAF is a major consumer of energy in the power grid. Therefore, every little improvement in the process will translate to a substantial cost saving.
- 2- Experiment with the actual EAF is too risky and, if the modification has not been evaluated properly, could be very costly and even dangerous for operation.
- 3- Building the simulator involves cascading the intelligent and conventional components in the closed loop format. The use of such a platform could be expanded in solving similar problems in future.

1.4 Previous Work

EAF, due to its significant potential of energy saving and its impact on power quality, has been the subject of many research. At the beginning, and for the duration of this research, a literature survey was conducted to discover the extent of these studies and defined the contribution of this research.

Both conventional and Artificial Intelligent (AI) based system identification methods were utilized in building the computer models for the components of the EAF. In the conventional methods [1]-[5], the equivalent R-L circuits in combination with the current voltage (I-V) characteristic curves of EAF, are generally put in use. Essentially, a set of dominating equations is required to simulate the time response of the system [1]-[3]. As well, there have been researches to model the thermodynamic relationship of the EAF [4]. The most

significant disadvantage of the conventional models is their lack of adaptability with changes in the process conditions. In the meantime, the EAF is a complex nonlinear system with variable load during the melting process. Even certain process conditions vary (such as, scrap type, refractory and electrode conditions) from batch to batch. As a result, adaptability of the computer model is rather important. There have been assumptions as to how to deal with the uncertainty and nonlinearity of the process for the majority of the mathematical models of the EAF. While these assumptions might be tolerable for the models with specific study purposes, typically the system becomes oversimplified for a general purpose EAF simulator. Throughout the past two decades, the AI techniques, such as Neural Networks (NN), Fuzzy Logics and Neuro Fuzzy systems, have been successfully applied in the modeling of the industrial systems and processes. The process of EAF, due to its nonlinearity, uncertainty and noisy environment, has been an attractive application for using the AI techniques [6]-[12]. Initially, King and Nyman proposed a feedforward NN to model the dynamics of EAF. The proposed study focused on intelligent modeling and adjustment of input voltage and current. It consisted of three layers, and utilized the previous five history states of current and voltage as inputs to obtain the voltage in its current state. Their experiment was based on a single phase small EAF. This model failed to take into account the important factor of interaction between the phases in the actual three phase system.

A series of research has been conducted by Sadeghian and Lavers to apply AI techniques in the modeling of EAF. To begin with, they used NN in the form of multi-layer Perceptron and Radial Based Function (RBF) network to model the dynamic behavior of EAF voltage [7]. In this research, the RBF is designated as an adequate solution because of its fast learning and flexibility. Furthermore, the study conveyed that the increase in the number of nodes would reduce the error but ultimately lead to an increased training time. Sadeghian and Lavers conducted another research for the use of fuzzy systems and adaptive fuzzy systems as a natural replacement of RBF network [8], [9]. In these fuzzy systems, the suggested inputs were current, the current derivative in respect to time, and the previous state of voltage. The output of the fuzzy system was the new state of voltage. From this research, Sadeghian and Lavers concluded that the fuzzy systems were trained faster; however,

because of their trial and error nature, they had less accuracy in comparison with RBF network. Therefore, they provided an intermediate solution in using the Neuro-Fuzzy structure to achieve both fast training and reasonable accuracy. As well, Sadeghian and Laver used the Recurrent Neuro Fuzzy systems as a prediction method for the voltage-current characteristic of the EAF [10], [12]. The ANFIS structure employs the past history of input variables, current and voltage, to predict the future state of these variables. The study demonstrated, for a relatively extensive range of predictions for the system states, more history states and fuzzy rules were required. Moreover, increasing either variable would subsequently increase the training time.

There are also numerous studies in the applying of AI techniques in the control systems of the EAF [13]-[15]. The AI techniques utilized in the EAF were inclusive of NN in the regulator control [13]. As well, there was an attempt to replace the PID controller in the EAF regulator with the Event-based NN and fuzzy controller [14]. In order to make use of these intelligent controls in the fast response systems of the regulator control, fast data collection and the implementation of the logic in the firmware level is necessary. As the technology of embedded systems and faster data acquisitions progresses, we will see more of the AI techniques used in power applications such as the EAF. Moreover, the mix of AI techniques has been recently more successfully practiced [15] in control of the slower EAF sub-systems, such as the formation of the foamy slag.

There are also several, commercially available, intelligent controllers for the regulator loop of the EAF. The methodology and details of these conclusions were never revealed in any scientific publication; therefore, the literature survey has excluded these solutions.

1.5 The Research Contribution

- Data and process analysis of the EAF regulator loop in order to come up with the main components of the systems.
- Implementing the ANFIS models of the process variables involved in the regulator loop with completely new set of inputs and outputs.
- Adapting some of the conventional power calculation and closed loop control as they are implemented in the real EAF.
- Developing a sequence event generator to simulate the operator manual commands.
- Integrating and verifying the entire system in MATLAB® Simulink® environment.
- Implementing the fuzzy gain scheduling PID control, instead of the conventional gain scheduling PI control in the EAF simulator, and measuring the improvement results.

1.6 Thesis Outline

The following chapters of this thesis are organized as:

Chapter 2 is a short description of some of the background theory used in this research. In addition, at the end of this chapter MATLAB® ANFIS toolbox, which is widely used in the implementation of this research, is explained.

Chapter 3 describes the data collection network used in this research, as well as the list of most important data items collected in this study. Also, graphical reports and discussions from the collected plant data illustrate some of efforts that were put in the system analysis phase of this research.

Chapter 4 explains the overview of the simulator, and provides more details regarding some of the MATLAB® Simulink® blocks that were developed for this research.

Chapter 5 discusses the models' verification process and summarizes the verification results in the form of tables and graphs. Also in this section the results of using fuzzy gain scheduling PID controller in the regulator control has been shown.

Chapter 6 concludes the contributions of this study and indicates the prospective work for this research.

Chapter 2

The Background Theory

In this chapter some of the theories and techniques used in this research are briefly reviewed. In order to understand ANFIS, which is coming from bonding of fuzzy systems and neural network, some background knowledge of both of these AI techniques are reviewed. Then in order to understand the training methods used in ANFIS to minimize the errors of the models, some basic concepts from optimization techniques are reminded. At the end of this chapter the MATLAB[®] ANFIS tool box which is used in implementation of the models is briefly explained.

The purpose of this chapter is not to explain these concepts in complete. It is just for the review purposes and is written for somebody already familiar with these fields. Most of the theories and the symbols presented in the formulas of this chapter have been captured from [36]-[37]. As a practical example of fuzzy logics in section 2.2, the application of fuzzy logic in gain scheduling of the PID controllers have been selected from [17]. The reason of choosing this particular application is because of its use in this research. The suggested technique of gain scheduling was applied in regulator loop and the results are illustrated in Chapter 6.

2.1 Fuzzy Logic

2.1.1 Fuzzy Sets

Fuzzy set is a term that is used in contrast with *crisp set*. A crisp set is based on the traditional definition of the set that draws clear borders in the input space between the members belonging to the set and the members excluded from the set. As an example R^+ , $R^+ = \{x \mid x \geq 0\}$, which are all the real numbers greater than or equal to zero constitute the Crisp sets.

If we use X as a space of objects and x as a member of this space, a classic set of A could be defined as $A \subseteq X$. In the space of X , we could define all the members of A as pairs such as: $(x,1)$ where $x \in A$ and a pair $(x,0)$ where $x \notin A$.

If we start positioning numbers between 0 and 1 to show the partial belonging of some space members of X to A , we could define a new but very common sense concept – fuzzy set - as it is shown by pairs like:

$$A = \{(x, \mu_A(x) \mid x \in X\} \quad (1)$$

In these pairs, $\mu_A(x)$ is called *Membership Function* (MF) and maps the space of X , denoted by *universe of discourse*, into membership value between 0 and 1.

Membership functions could have different shapes and even be custom-made to have the best mapping for solving a particular problem. MATLAB[®] has the most popular membership functions. Fig. 3 shows some of the most popular membership functions in MATLAB[®]. In the whole of membership functions, the vertical axis is a number between 0 and 1.

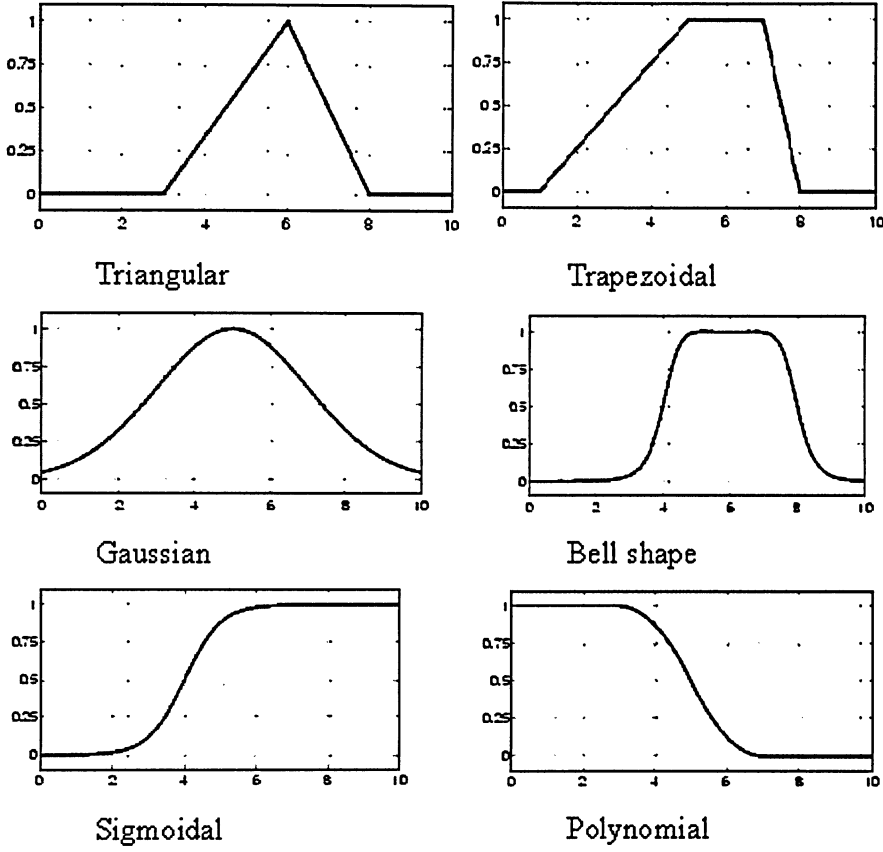


Figure 3: Example of membership functions.

2.1.2 Fuzzy Operations

By using fuzzy operations we could combine two or more fuzzy sets and conclude a new fuzzy set. The most popular fuzzy operations are as follows.

The union of two fuzzy sets A and B with the membership functions of $\mu_A(x)$ and $\mu_B(x)$ is a new fuzzy set C with the membership function of $\mu_C(x)$ as:

$$C = A \cup B = A \text{ OR } B \text{ where } \mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x)$$

(2)

Similarly, the intersection of two fuzzy sets A and B with the membership functions of $\mu_A(x)$ and $\mu_B(x)$ is a new fuzzy set C with the membership function of $\mu_C(x)$ as:

$$C = A \cap B = A \text{ .AND. } B \text{ where } \mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \quad (3)$$

Also the complement of fuzzy set A with a membership function of $\mu_A(x)$ is defined as:

$$\bar{A} = \neg A = \text{NOT. } A = \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (4)$$

2.1.3 Fuzzy Rules and Reasoning

One of the main advantages of the fuzzy system is in its use of the linguistic values, which are defined by fuzzy sets in the universe of discourse. The linguistic variables and values are what we use in everyday conversation. The linguistic variable could take values such as: soft, tall, fast, dark, poor, far, etc...

The purpose of fuzzy rules is to mimic the decision making process that the brain undergoes with its existing knowledge. The fuzzy rules are the core of any fuzzy inference system, which decodes the decision making process based on human expertise, science or scientific observation. In its simplest form, the fuzzy rule looks like the following If, then phrase:

If x is A then y is B ,

where A and B are linguistic values defined in universe of discourse of X and Y . The part between 'if' and 'then' is called *antecedent* or *premise* and the part after 'then' is called *consequence* or *conclusion*. In fuzzy reasoning, by applying the current input data to a set of fuzzy rules we want to reach to a conclusion. Usually the inputs to a fuzzy system are non-fuzzy or crisp and the output(s) are expected to be crisp, as well.

2.1.4 Fuzzy Inference Systems

Fuzzy inference systems is a systematic way of using stored rules and membership functions in calculating the output(s) of the fuzzy systems based on it's current input(s). The most popular fuzzy inference systems are Mamdani and Sugeno. Both of these systems are common in most of the fuzzy inference process steps. In general, the rules in fuzzy systems are nonlinearly mapping the fuzzy regions in premise parts (input space) to the fuzzy regions specified in consequent parts (output space). The order of the rules in fuzzy system does not matter. The system's final output is calculated when all of the rules are applied. The effect of a rule in the final output is called the *weight* of the rule. Unless specified otherwise by the system designer, the effects of every rule in inference process are equal. In this case, some of the rules with heavier weight could have more effect on calculating the overall system output.

In *Mamdani* fuzzy models, if the inputs are crisp, they apply to their associate membership functions. The result is another value between 0-1. This step is called *Fuzzification*. In the next step, all of the premises of the rules will be solved based on the defined *Fuzzy Operations* (AND, OR, NOT). The result of fuzzy operation in each rule indicates the firing strength of the rule. The firing strength (between 0-1) is used in the *Implication Operation*, which is another operation to qualify the consequent MF. The result of the qualified MF in consequents of the rules will then be combined in the *Aggregation Operation*. The result of the aggregation operation is one overall output for the system. In most of fuzzy systems, especially the fuzzy control system, the result should be in crisp logic. The last operation, known as *Defuzzification Operation*, will conclude a crisp number from overall fuzzy output.

In order to visualize all of the steps in the fuzzy inference, simple fuzzy systems with two inputs and one output are shown on Fig. 4. In this example, input1 is using three Gaussian and Sigmoid MFs, input2 is using two trapezoidal MFs, and the output is using two triangular MFs. The fuzzy operation is an AND (minimum) in all of the premises of the rules.

The implication operation is Minimum and aggregation operation is Maximum. The defuzzification method is Centriod, which will be explained shortly. Based on the six rules listed in Table2, by applying the crisp inputs of [2, 4] the crisp output of 5.03 has been calculated. The nonlinear mapping of inputs into the output space is shown in Fig. 5.

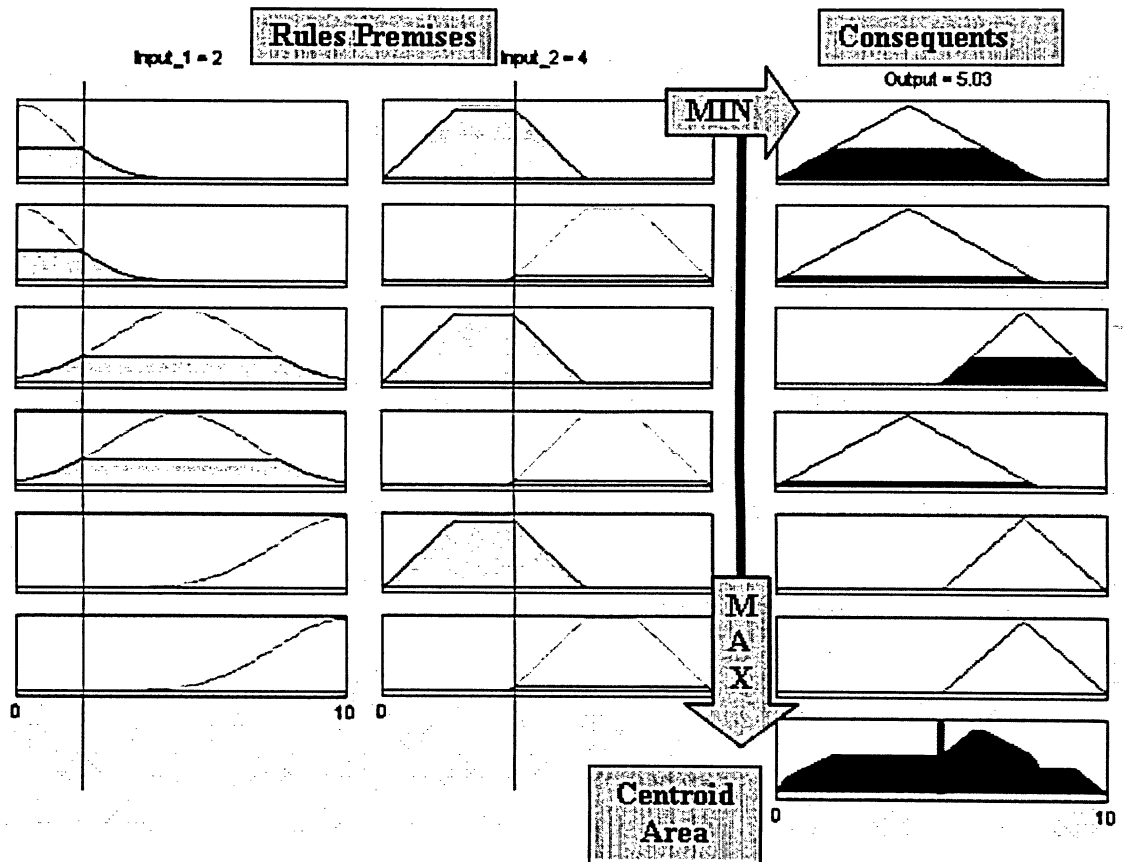


Figure 4: Fuzzy inference process with two inputs and one output.

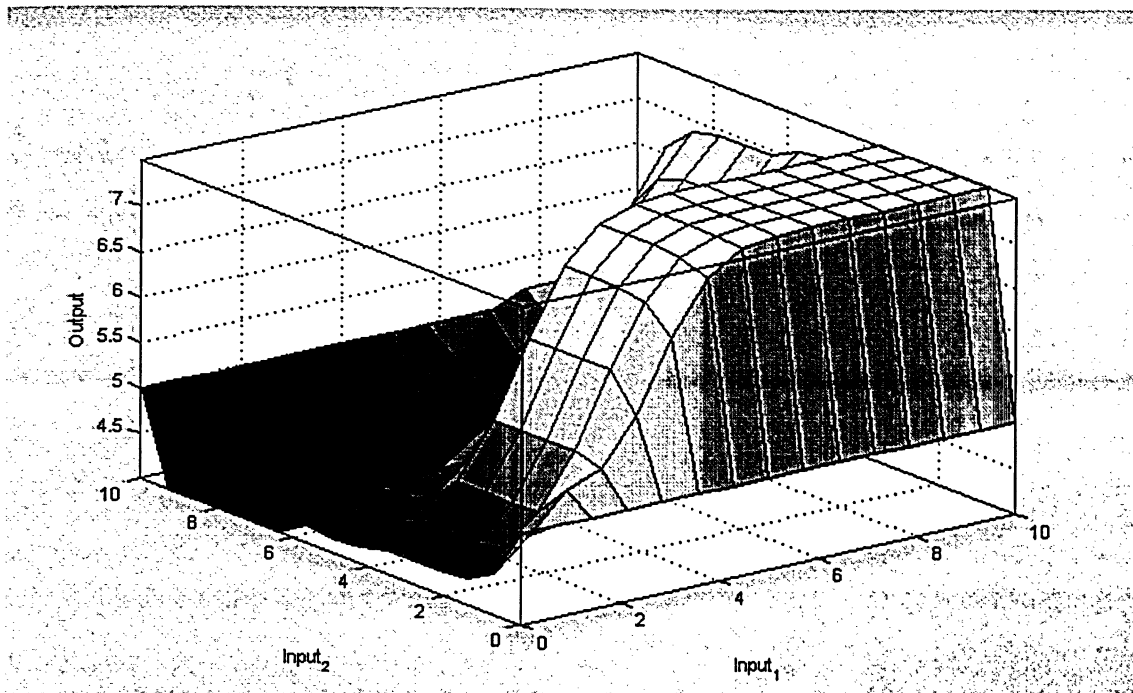


Figure 5: The nonlinear mapping of input and output space.

Table 2: An example of fuzzy rules

Input1	Operation	Input2	Output
In1_MF1	AND	In2_MF1	Out_MF1
In1_MF1	AND	In2_MF2	Out_MF1
In1_MF2	AND	In2_MF1	Out_MF2
In1_MF2	AND	In2_MF2	Out_MF1
In1_MF3	AND	In2_MF1	Out_MF2
In1_MF3	AND	In2_Mf2	Out_MF2

In Mamdani fuzzy models, the methods of fuzzy operations; implication, aggression and defuzzification, are not limited to the use of methods in this example. Even MATLAB® allows the programmers to design custom-made functions for every one of the fuzzy inference steps, if they wish. Mamdani fuzzy model is the only one that has the defuzzification in its process. The most common method of defuzzification is *Centroid of*

Area (COA). For defuzzifying a fuzzy set A of a universe of discourse of Z in *COA* method we have:

$$COA = \frac{\int_Z \mu_A(z) z dz}{\int_Z \mu_A(z) dz} \quad (5)$$

There are also other popular methods of defuzzification such as: bisector of area, mean of maximum, smallest/biggest of maximums [36]-[37].

Mamdani fuzzy model works well, when the set of rules could be built based on the knowledge of experts or scientific observations. *Sugeno Fuzzy Model* is built to have more flexibility in terms of building the rules automatically based on the recorded inputs and outputs of the systems. In Sugeno fuzzy systems, the premise parts of rules are still fuzzy but the consequent parts are a function of the inputs:

$$\text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = f(x, y)$$

where x and y are the systems inputs, A and B are fuzzy sets and z is usually a polynomial function of the inputs. The order of polynomial indicates the order of Sugeno systems. If we choose the order of polynomial as zero, a singleton function in the consequent of the rules, Sugeno model then becomes a powerful modeling tool for nonlinear systems. A significant advantage of Sugeno models in comparison with Mamdani is the avoidance of the time-consuming process of defuzzification, since the consequent parts are already crisp. Usually, the straightforward weight-averaging method is used to conclude the overall crisp output of the system.

2.2 Fuzzy PID Controllers

Fuzzy systems have been applied in various domains. One of the most common applications of fuzzy systems is in control systems. These types of controllers are typically known as *Fuzzy Controllers*. The Proportional Integral Derivative (PID) controllers are the most prevalent types of the linear closed loop controls. There has been extensive research in the past two decades to employ fuzzy systems in the field of PID controllers [16]-[30]. The application of fuzzy systems in PID controllers is referred to as fuzzy PID controllers. Fuzzy controllers are used both in tuning the classical PID controller's coefficients and as a replacement for classical PID controllers. This section has summarized the fuzzy systems' method utilized to tune the coefficients of the classic PID controllers, known as *Gain Scheduling* [17]. Fuzzy PID Gain Scheduling is an example of the fuzzy controls. Due to its potential for the future use in intelligent control of the EAF, it is selected as the practical example of the fuzzy logic. Later on in Chapter 7, the results of implementing the fuzzy gain scheduling PID in regulator loop of EAF are illustrated.

In fuzzy gain scheduling, the conventional PID controller and the fuzzy system work interactively alongside each other. The conventional PID controller in continuous time domain can be written as:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (6)$$

where, K_p, K_i, K_d in sequence represent the coefficients for proportional, integral and derivative parts.

In the discrete domain with the sample time of T_s , the controller output of $u(k)$ can be calculated as:

$$u(k) = K_p e(k) + K_i T_s \sum e(i) + \frac{K_d}{T_s} \Delta e(k). \quad (7)$$

where, $e(k)$ denotes the error, the difference between the desired signal reference and the actual process output. Furthermore, $\Delta e(k) = e(k) - e(k-1)$ is the difference between the error signals of the current sample versus the previous sample.

If we simplify the system by assuming a linear relationship among the integral and derivative time constants with a coefficient of α , then we can compile:

$$\begin{aligned} T_i &= \alpha T_d \\ K_i &= \frac{K_p}{\alpha T_d} = \frac{K_p^2}{\alpha K_d} \end{aligned} \quad (8)$$

where, T_i and T_d are the time constants for the integral and derivative parts. The function of the fuzzy gain scheduler is to observe the variations in $e(k)$ and $\Delta e(k)$ signals and determine the adjustment required on K_p, K_d and α . The changes suggested by the fuzzy gain scheduler are based on the progressive response of the system. Using the PID gain scheduler alongside the conventional PID, renders the PID controller adaptive to the dynamic changes in process. The interaction between the conventional PID controller and fuzzy gain scheduler are illustrated in Fig. 6.

Zhao and Isaka [17], as illustrated in Fig. 7, used triangular membership functions to fuzzify $e(k)$ and $\Delta e(k)$. In this fuzzification, the linguistic values of NB, NM, NS, ZO, PS, PM and PB sequentially represent Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium and Positive Big. A set of fuzzy rules illustrate the modifications in $e(k)$ and $\Delta e(k)$ in the premise of the rules and suggest the appropriate changes on K_p, K_d and α based on the membership functions, as shown in Figs. 8 and 9. As we could observe in Fig. 9, the membership functions suggested for α are singletons. In Fig. 8, the S and B

represent the linguistic values of Small and Big and the membership functions are chosen sigmoid.

The fuzzy rules for this system are summarized in the Table 3. Each element in this table sequentially represents K_p , K_d and α in the consequent of the rules based on the current values of the premise values of $e(k)$ and $\Delta e(k)$. For example, if we refer to the element of the table located at second row and the seventh column, we could interpret the fuzzy rule as follows:

If $e(k)$ is Negative Medium and $\Delta e(k)$ is Positive Big

Then K_p is Small and K_d is Big and α is 3.

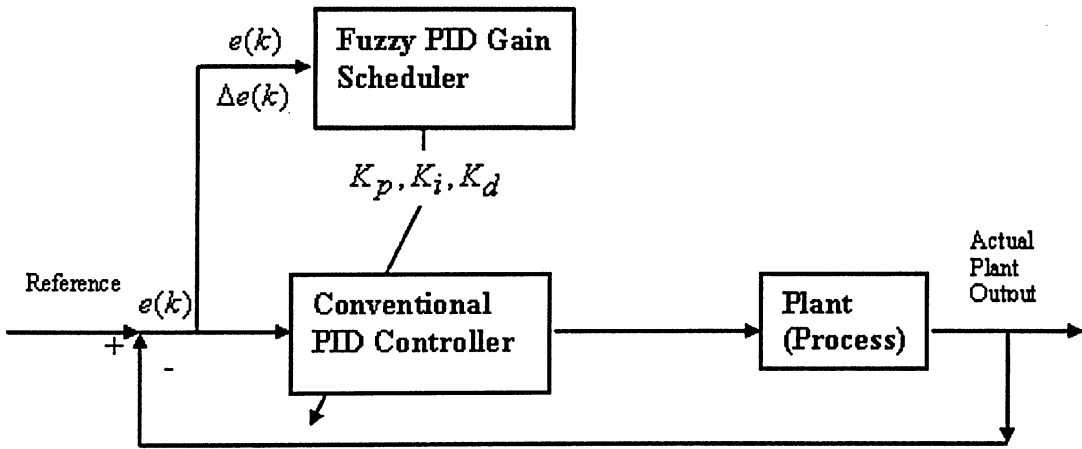


Figure 6: The Fuzzy Gain Scheduling PID Controller.

Zhao and Isaka demonstrated a better time response of fuzzy gain scheduling PID in comparison with a well tuned classic PID [17].

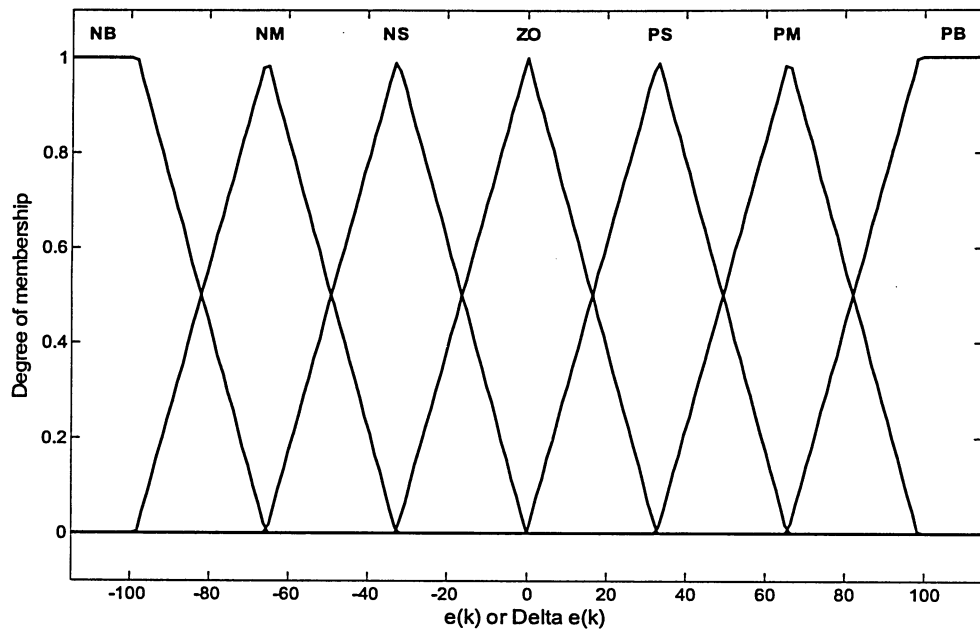


Figure 7: Membership functions for error and changes on error signals.

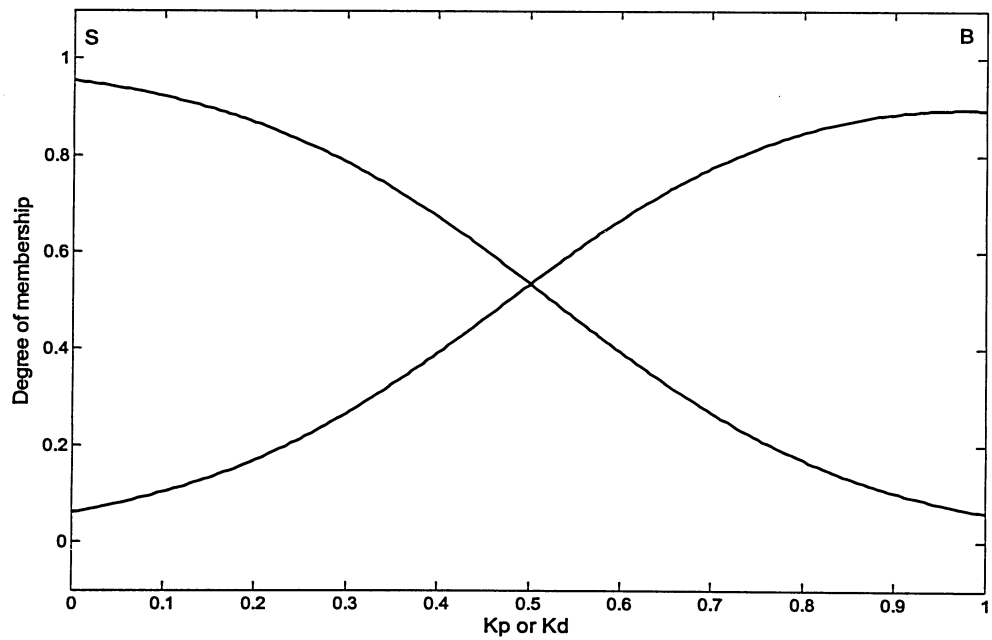


Figure 8: Membership functions for proportional and derivative coefficients.

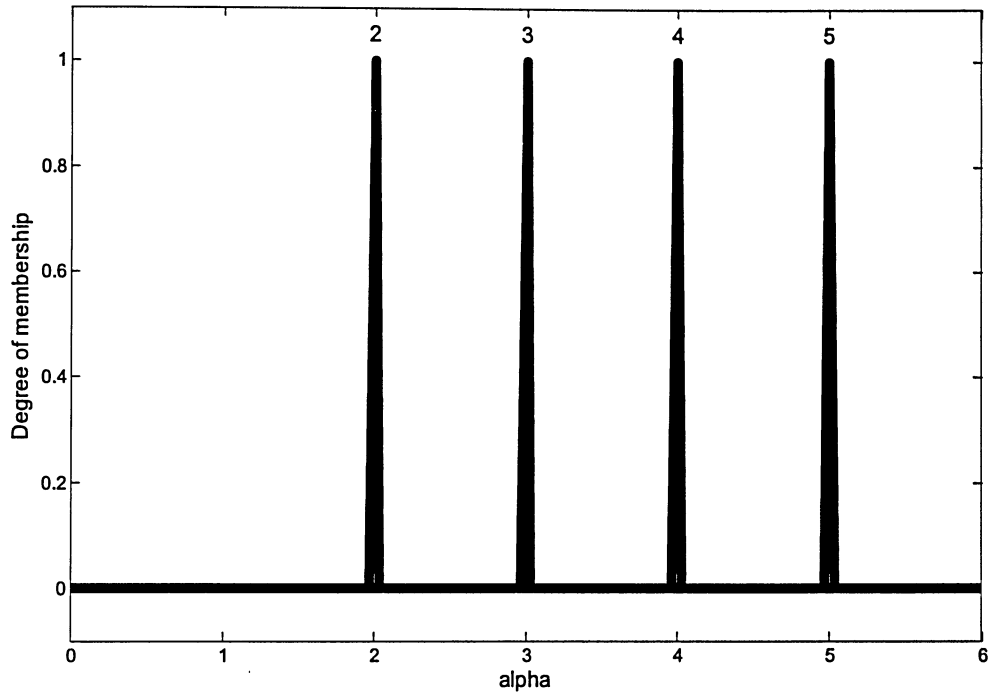




Figure 9: Singleton membership functions for linear coefficient to relate integral and derivative time constants.

Table 3: PID Gain Scheduling rules

	$\Delta e(k)$ 							
$e(k)$ 		NB	NM	NS	ZO	PS	PM	PB
NB		B,S,2	B,S,2	B,S,2	B,S,2	B,S,2	B,S,2	B,S,2
NM		S,B,3	B,B,3	B,S,2	B,S,2	B,S,2	B,B,3	S,B,3
NS		S,B,4	S,B,3	B,B,3	B,S,2	B,B,3	S,B,3	S,B,4
ZO		S,B,5	S,B,4	S,B,3	B,B,3	S,B,3	S,B,4	S,B,5
PS		S,B,4	S,B,3	B,B,3	B,S,2	B,B,3	S,B,3	S,B,4
PM		S,B,3	B,B,3	B,S,2	B,S,2	B,S,2	B,B,3	S,B,3
PB		B,S,2	B,S,2	B,S,2	B,S,2	B,S,2	B,S,2	B,S,2

2.3 Optimization

3.3.1 System Identification

System identification is about observing inputs and outputs of the system to identify the relationship between the input and output in order to predict the output of the system at any other applied inputs. The result of this identification could be in the form of parameterized function such as:

$$y = f(\mathbf{u}; \boldsymbol{\theta}) \quad (9)$$

where, y is the model output when the input vector of \mathbf{u} applies to the system and $\boldsymbol{\theta}$ is the parameter vector. Tuning the model denotes adjusting the parameter vector of $\boldsymbol{\theta}$ to minimize the error e , which is the difference between the actual output of the system Y and the model output y .

$$e = Y - y \quad (10)$$

Usually, tuning the model requires many iterations, where in each iteration the error is calculated and compared with the acceptable threshold. In order to minimize the error various optimization techniques need to be utilized. Some of these techniques are discussed in this chapter.

In order to use computers in system identification, first the system identification problems have to be defined in matrix form. Later on in this chapter the derivative optimization techniques are covered. Before we precede to the details of these optimization techniques, let's refresh with a few matrix derivative definitions.

2.3.2 Least-Square Estimator

Later on in this chapter the Least-Square technique is used to estimate the parameters of the ANFIS models. Therefore it is beneficial to have quick review of this technique. Assuming that m data pairs have been collected to train the system, the pairs of inputs and outputs could be shown as: $\{(u_i, y_i) | i = 1, \dots, m\}$. The optimization goal is to minimize the model error. If we deal with a linear system, the training data could be presented as the set of m linear equations, if the system assumed to be linear on its parameters as:

$$\begin{cases} f_1(u_1)\theta_1 + f_2(u_1)\theta_2 + \dots + f_n(u_1)\theta_n = y_1, \\ f_1(u_2)\theta_1 + f_2(u_2)\theta_2 + \dots + f_n(u_2)\theta_n = y_2 \\ \vdots \\ f_1(u_m)\theta_1 + f_2(u_m)\theta_2 + \dots + f_n(u_m)\theta_n = y_m. \end{cases} \quad (11)$$

We could show the above set of linear equations as the matrix form of:

$$A\theta = y \quad (12)$$

where $A = \begin{bmatrix} f_1(u_1) & \dots & f_n(u_1) \\ \vdots & & \vdots \\ f_1(u_m) & \dots & f_n(u_m) \end{bmatrix}$

The i^{th} row could be show as $a_i^T = [f_1(u_i), \dots, f_n(u_i)]$

It is necessary for the number of training data to be greater than or equal to the number of parameters ($m \geq n$). If A is square ($m = n$) and nonsingular, we could solve the equation:

$$\theta = A^{-1}y \quad (13)$$

However, the equation has to be modified to accommodate the error vector of e to account for noise and model errors:

$$A\theta + e = y \quad (14)$$

Where $e = y - A\theta$

Rather than finding the exact solution to the set of equations, we want to find $\theta = \hat{\theta}$ to minimize the sum square error defined by:

$$E(\theta) = \sum_{i=1}^m (y_i - a_i^T \theta)^2 \quad (15)$$

The square error is minimized when $\frac{\partial E(\theta)}{\partial \theta} = 0$ at $\theta = \hat{\theta}$

Least Square Estimator (LSE) produces the minimum point of $\theta = \hat{\theta}$ at:

$$A^T A \hat{\theta} = A^T y \quad (16)$$

(for proof please see [36]).

If $A^T A$ is nonsingular, $\hat{\theta}$ is unique and is given by:

$$\hat{\theta} = (A^T A)^{-1} A^T y \quad (17)$$

If $A^T A$ is singular then the concept of generalized inverse can be utilized to find the minimum.

2.3.3 Gradient Based Optimization Method

The derivative based optimization methods are mostly based on the concept of gradient decent techniques. The subject is briefly explained here.

Suppose we have an objective (cost) function of E defined on an n -dimensional input space of θ as:

$$\theta = [\theta_1, \dots, \theta_n]^T \quad (18)$$

In finding the minimum of θ , we choose an iterative descent method and in every iteration step the θ_{next} is determined by a step down from the current point θ_{now}

In the direction vector d :

$$\theta_{next} = \theta_{now} + \eta d \quad (19)$$

In this equation, η is called step size and indicates to what extent to proceed in the direction of d . In order to achieve the local or global minimum, d and η has to satisfy the following inequality:

$$E(\theta_{next}) = E(\theta_{now} + \eta d) < E(\theta_{now}) \quad (20)$$

If the cost function E is a differentiable function, the first derivative of E is called gradient denoted by g , as of:

$$g(\theta) (= \nabla E(\theta)) \stackrel{def}{=} \left[\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}, \dots, \frac{\partial E(\theta)}{\partial \theta_n} \right]^T \quad (21)$$

We wish to find a value of θ_{next} that satisfies the following:

$$g(\theta_{next}) = \left. \frac{\partial E(\theta)}{\partial \theta} \right|_{\theta=\theta_{next}} = 0 \quad (22)$$

In the gradient iterative techniques, we continue the algorithm until the objective function is smaller than the predefined threshold or the computation times out. In order to move to the direction of minimum, if $d = -g(\theta)$ then d is the Steepest Decent direction at a local point θ_{next} . Therefore, using the steepest decent method, we could simply calculate the new points based on:

$$\theta_{next} = \theta_{now} - \eta g(\theta) \quad (23)$$

2.4 Adaptive Networks

Adaptive network is a connection of cells or nodes. In this type of network each node performs a parametric function on its inputs and passes the result for processing in other nodes. The nodes are organized in layers. The output of one node is usually used as input of another node in the next layer (feedforward network). The first layer connected to the inputs is called the input layer and the last layer connected to the outputs is called output layer. The layers in between are known as hidden layer. The nodes are usually identified with their layer number and their corresponding order (top to bottom) in the layer.

Fig. 10 shows a typical adaptive network with three layers, two inputs, and two outputs. Some of the functions have adjustable parameters and others are fixed. In this figure, the circular nodes represent the parametric nodes and the square nodes are fixed. By adjusting the parameters of the adjustable nodes in adaptive networks, we could achieve the desired nonlinear mapping of the input space into the output space. The process of adjusting the parameters of the adaptive network is called *learning*. There are two different methods of learning for adaptive networks. If the desired adaptive outputs of networks are known for the presented inputs, the learning method is called *supervised learning* (training). In contrast, the *unsupervised learning* methods are used when the desired outputs are not known. In this

research, we focus on one of the most popular methods of supervised learning known as *backpropagation*. The backpropagation method is explained more in the next section. As it is shown in Fig. 10, the output of each node is indicated with the index of layer number and node number. The adaptive network is called *Feedforward* when the direction of the connections between the nodes is forward, indicating that the output of each node in every layer proceeds to the next layer in the network. If there is any feedback link existing in the network (connections from right to left), the network is called *recurrent*. The Feedforward adaptive networks are used in this research.

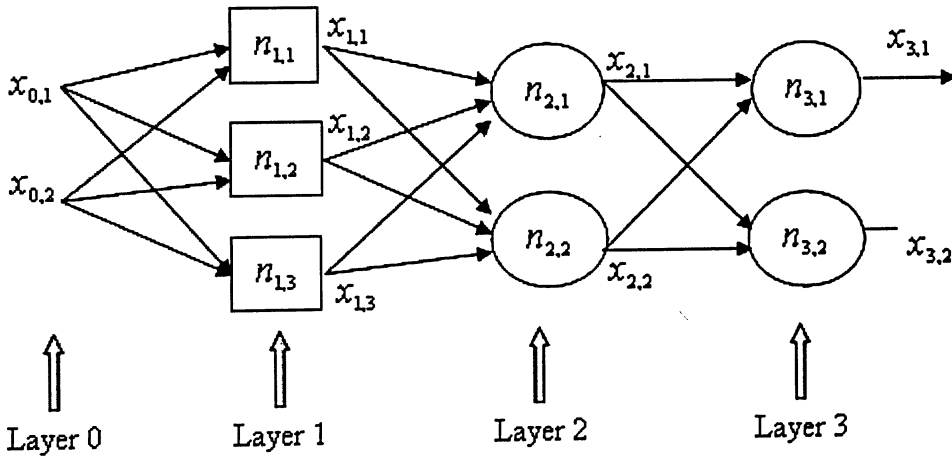


Figure 10: A typical feedforward adaptive network.

2.4.1 Backpropagation

Backpropagation is one of the most effective and popular methods of supervised learning for adaptive networks. The idea of back propagation is to introduce a training data set with P entries. For each one of these data entries, the desired output is indicated by the data set D . For example, for p^{th} input data ($1 \leq p \leq P$) there is a d_k representing the desired out put. If we consider a multilayer feedforward network with L layers and use the index l for layer number, where ($l = 0, 1, \dots, L$), $l = 0$ denotes the input layer and $l = L$ the output layer. If

there are N nodes in the final layer, the error when we introduce the p^{th} member of the input vector can be calculated by the use of sum of squared error formula as:

$$E_p = \sum_{k=1}^{N(L)} (d_k - x_{L,k})^2. \quad (24)$$

In this formula $x_{L,k}$ represents the k^{th} nodes output in the output layer of L when the p^{th} input has been fed to the network. In order to get the best performance from the Neural Network, we have to minimize the error. Backpropagation utilizes the steepest descent optimization method to minimize the error. The output of each node in the multilayer network at the location of layer l and node i is a function $f()$ of its inputs (the previous layer outputs) and the parameters of its own function ($\alpha, \beta, \gamma, \dots$):

$$x_{l,i} = f_{l,i}(x_{l-1,1}, \dots, x_{l-1,N(l-1)}, \alpha, \beta, \gamma, \dots) \quad (25)$$

Therefore, when the parameters of one of these node functions (anywhere in the network) are changed, it not only affects its own output but also the final output of the network in the last layer. As a result, the measured error will also be changed.

In order to use the steepest decent method in minimizing the error, we must first obtain the gradient vector. The method is referred to as backpropagation since the gradient vector passes from the derivative of the output layer towards the input layer. In taking the derivative, rather than ordinary partial derivative, we use ordered derivative with the symbol of $\frac{\partial^+}{\partial}$. The ordered derivative of measured error E_p with respect to the output layer of node i in the layer l is defined by:

$$\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}}. \quad (26)$$

The reason for the use of ordered derivative is the fact that the ordinary partial derivative of $\frac{\partial E_P}{\partial x_{l,i}}$ for all the middle layers ($l \neq L$) is zero. However, we know that the output of every

node in the middle layers affects the output of measured error indirectly. In calculating the $\varepsilon_{l,i}$ when ($l = L$), it simply means for the output layer $\varepsilon_{l,i} = -2(d_i - x_{L,i})$. For the middle (hidden) layers, whereby using the chain rule from basic calculus, we have:

$$\varepsilon_{l,i} = \underbrace{\frac{\partial^+ E_P}{\partial x_{l,i}}}_{\text{error signal at layer } l} = \sum_{m=1}^{N(l+1)} \left(\underbrace{\frac{\partial^+ E_P}{\partial x_{l+1,m}}}_{\text{error signal at layer } l+1} \underbrace{\frac{\partial f_{l+1,m}}{\partial x_{l,i}}}_{\text{P. derivative of node function layer } l+1} \right) = \sum_{m=1}^{N(l+1)} \varepsilon_{l+1,m} \frac{\partial f_{l+1,m}}{\partial x_{l,i}} \quad (27)$$

The above equation is the basic formula for backpropagation. It illustrates that we could always start from the last layer, and propagate the error signals backward iteratively to the desired layer. This equation in algorithmic form could also be shown as:

$$\varepsilon_i = \frac{\partial E_P}{\partial x_i} = \underbrace{\frac{\partial f_{n+1}}{\partial x_i}}_{\text{Direct effect of } x_i \text{ on } E_P} + \underbrace{\sum_{i < j \leq n} \varepsilon_j \frac{\partial f_j}{\partial x_i}}_{\text{Indirect effect}} \quad (28)$$

The gradient vector is defined as the derivative of the error measured with respect to each node function parameter. By applying the chain rule for the gradient vector, we have:

$$\frac{\partial^+ E_P}{\partial \alpha} = \frac{\partial^+ E_P}{\partial x_{l,i}} \frac{\partial f_{l,i}}{\partial \alpha} = \varepsilon_{l,i} \frac{\partial f_{l,i}}{\partial \alpha} \quad (29)$$

In this equation, α is the parameter of the i^{th} node at layer l . This method is used for *online-training*; where the parameters of the adaptive network would be adjusted subsequent to applying each member of the input vector.

For the training set of P, we can calculate gradient vector as:

$$\frac{\partial^+ E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial^+ E_p}{\partial \alpha} \quad (30)$$

This equation is used for *offline (batch) learning* when the parameters of the adaptive network are corrected once all the training data has been applied. With applying the steepest descent, the steps of updating the parameter α is defined as:

$$\Delta \alpha = -\eta \frac{\partial^+ E}{\partial \alpha} \quad (31)$$

In this equation, η is the learning rate and can be calculated from:

$$\eta = \frac{k}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha}\right)^2}} \quad (32)$$

k is called the step size and illustrates how far we have moved along the gradient in each iteration. In order to use backpropagation learning method, the adaptive network node functions have to be piecewise differentiable, and requires to have feedforward structure.

2.5 ANFIS

Due to the adaptive network flexibilities, they are suitable as solutions to many linear and non-linear problems. A specific case of adaptive network, which behaves similar to the fuzzy systems, is the ANFIS structure. ANFIS employs the strength of both fuzzy systems in knowledge presentation and adaptive networks in learning and adaptation.

The following example captured from [36] illustrates the ANFIS structure. Take into consideration a simple Sugeno fuzzy system with two inputs and a single output with the following two rules:

Rule1: If x is A_1 and y is B_1 , then $f_1 = p_1x + q_1y + r_1$

Rule2: If x is A_2 and y is B_2 , then $f_2 = p_2x + q_2y + r_2$

Fig. 11 illustrates the equivalent five-layer ANFIS structure for this fuzzy system. In the first layer, every node is adaptive and the output applies the A or B membership functions into the inputs. Furthermore, the crisp inputs of x and y are fuzzified in this layer. In this example the output of node 1 in layer 1 is:

$$O_{1,1} = \mu_{A_1}(x) \quad (33)$$

And similarly the output of the 4th node in the first layer is:

$$O_{1,4} = \mu_{B_2}(y) \quad (34)$$

Next, the second layer with the fixed nodes applies the product of all incoming signals to calculate the firing strength of each rule. The operation in the nodes of this layer depends on the T-norm method that is chosen. The output of the nodes in this layer can be shown as:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), i = 1, 2 \quad (35)$$

The fixed nodes in the third layer perform the normalization to calculate the ratio of each rule's strength to the sum of all rules' strength. The outputs of this layer could be shown as:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2 \quad (36)$$

Subsequently, the adaptive nodes in the forth layer applies the consequent polynomials with the coefficients of the normalized weight for the rule:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \quad (37)$$

Finally, in the fifth layer, the summation of all the outputs from the forth layer is calculated as the single output of the network:

$$O = O_{5,1} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (38)$$

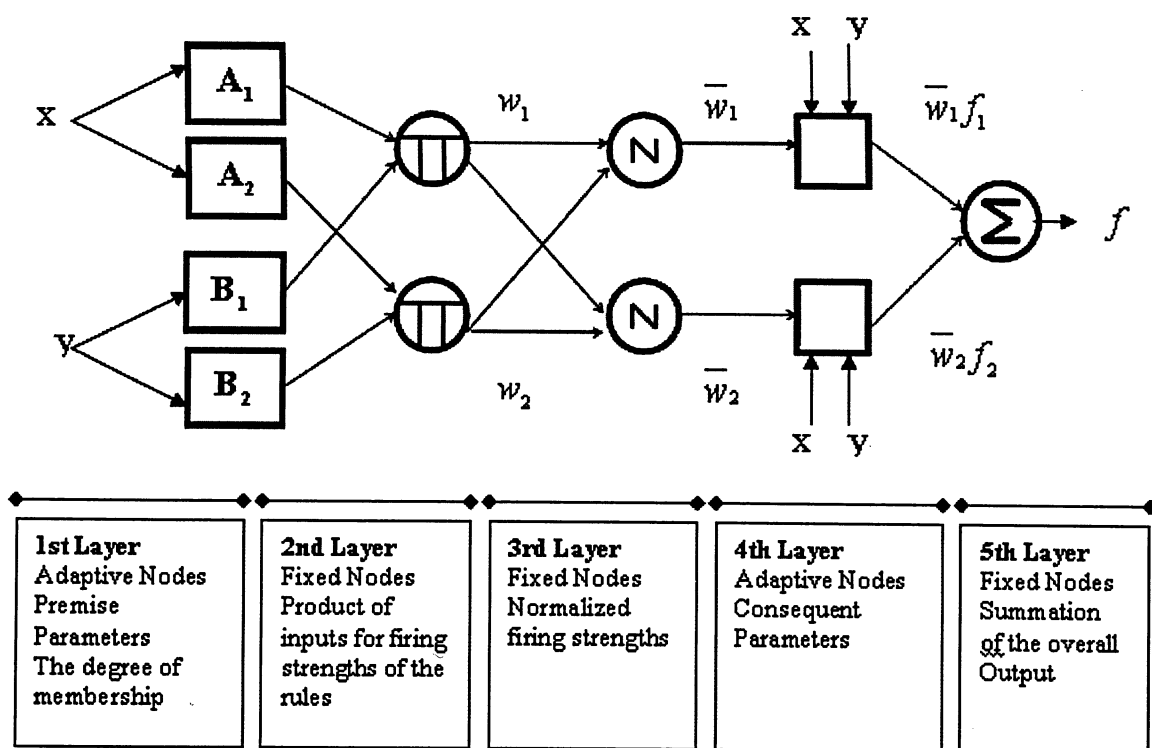


Figure 11: An example of ANFIS general structure (Figure with slight modification has been borrowed from [36]).

It is imperative to mention, there is a proved theorem that claims Radial Basis Function Network (RBFN) under certain conditions has an equivalent function of fuzzy structure. As a result, any leaning method applicable in Neural Network could be practiced in ANFIS structure.

Moreover, ANFIS is a zero-ordered Sugeno model with unlimited approximation power for modeling the nonlinear systems. This property makes ANFIS a favorable tool for nonparametric (black-box) modeling of nonlinear systems.

2.5.1 The Hybrid Learning Method

Although backpropagation is used widely as a learning method for adaptive networks, it is slow in converging. The hybrid methods of backpropagation and LSE converges rapidly and could be applied for nonlinear optimization. If the adaptive network has just one output, the output has a linear relationship with some of the network parameters. In this method, these parameters are separated and the LSE general formula is used to estimate them, as training set are introduced. The combined method could also be used as an either off-line or on-line learning method.

In the ANFIS structure, we have two types of adjustable parameters for premise (membership functions) and consequent (the first degree polynomial) of the fuzzy rules. Considering the premise parameters to be constant, the final single output of the ANFIS structure is a linear function of the consequent parameters. In the previous Sugeno ANFIS structure with two inputs, a single output, and two rules, a linear relationship can be illustrated as:

$$\begin{aligned}
 f &= \bar{w}_1(p_1x + q_1y + r_1) + \bar{w}_2(p_2x + q_2y + r_2) \\
 &= (\bar{w}_1x)p_1 + (\bar{w}_1y)q_1 + (\bar{w}_1)r_1 + (\bar{w}_2x)p_2 + (\bar{w}_2y)q_2 + (\bar{w}_2)r_2
 \end{aligned}
 \tag{39}$$

Consequently, we can establish iteratively a learning system that utilizes the LSE in the forward passes to estimate the consequent linear parameters. Subsequently, Gradient descent (backpropagation) could be used in the backward pass to adjust the premise parameters. The advantage of combining the two methods is that the algorithm will converge more rapidly by the use of LSE, in comparison with the exclusive use of backpropagation.

As Jang has discussed [36], the ANFIS learning algorithm is dependent on the volume of data and computation restriction. In cases where we have significant amount of data, tuning the premise parameters is recommended. On the other hand, when we have a rather trivial size of data with only few fuzzy rules from human expert, tuning the membership functions is not such a good idea.

2.5.2 C-Means Clustering

When we deal with relatively large input data set, clustering helps to save computation time and provide a better chance for algorithms to converge. In general, clustering creates several groups in the input space and appoints a member of each group as a representative or the center. One of the most popular clustering methods is called C-Means (or K-means). It is most often used as an off-line clustering method or rather, as a means of pre-processing a batch of data before the final processing in speech recognition, image processing, and black-box modeling.

In C-Mean algorithm, the collection of n vectors $X_j, j=1, \dots, n$ partitions into C groups $G_i, i=1, \dots, C$. The representative of each partition or group has the minimum distance from all of the other members of the group. If the C_i represents a cluster center, the cost function could be shown as:

$$J = \sum_{i=1}^C J_i = \sum_{i=1}^C \left(\sum_{k, x_k \in G_i} d(X_k - C_i) \right) = \sum_{i=1}^C \left(\sum_{k, x_k \in G_i} \|X_k - C_i\|^2 \right) \quad (40)$$

In this equation, $d(\dots)$ is not an actual function, rather only represents the general measured distance. While, in the latter part of the equation, Euclidean distance is specifically used. Initially, the centers of the clusters $C_i, i=1, \dots, C$ are selected randomly. Then, we form a

membership matrix of U , where the members in the conventional C-Mean algorithm are either zeros or ones.

$$\begin{cases} u_{ij} = 1 & \text{if } \|X_j - C_i\|^2 \leq \|X_j - C_K\|^2, \text{ for each } k \neq i, \\ u_{ij} = 0 & \text{otherwise.} \end{cases} \quad (41)$$

When $u_{i,j}$ is one, it signifies that the j th element of vector X_j belongs to group i . In other words, when $u_{i,j}$ is one, C_i is the closest center among all of the centers to the data point of X_j . Since in the conventional C-Mean algorithm each member of U matrix is either one or zero and each data element belongs to one and only one group, the following properties are obvious.

$$\sum_{i=1}^C u_{ij} = 1, \forall j = 1, \dots, n \quad (42)$$

$$\sum_{i=1}^C \sum_{j=1}^n u_{ij} = n \quad (43)$$

Therefore, the optimum center of C_i in each of the iteration can be calculated from:

$$C_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} X_k \quad (44)$$

where $|G_i|$ (is the size of G_i) = $\sum_{j=1}^n u_{ij}$.

The algorithm could continue until the cost function is less than the acceptable threshold. Because the centers are selected randomly at the beginning, the algorithm may need to be repeated several times for better results.

Moreover, C-Mean clustering has been implemented as Fuzzy C-Means Clustering (FCM) as opposed to conventional C-Means. In this method, each member belongs to a cluster (group)

with a degree of membership between 0-1. Therefore, the components of the U matrix are numbers between 0-1. In this approach, each data member hence belongs to multiple clusters with different membership degree.

2.6 MATLAB® ANFIS Toolbox

MATLAB® has a built in GUI for ANFIS data modeling. This toolbox is easy to use but has certain restrictions.

- It only supports single output models.
- The fuzzy system has to be first or zero order Sugeno type.
- All of the output functions must be the same type (either linear or constant).
- The defuzzification method only accepts weight averaging.
- All of the rules' weights are one and hence cannot discriminate amongst the rules.
- It is restricted to predefined membership functions and user defined functions cannot be accepted.
-

Fig. 12 illustrates MATLAB® ANFIS GUI. The steps to build an ANFIS model are grouped in GUI at the bottom menus from left to the right. These steps are:

- Importing the data from the workspace or disk;
- Generating the fuzzy inference system for the imported data;
- Training the ANFIS model; and
- Finally, validating the model to plot its output versus the trained or another set of data.

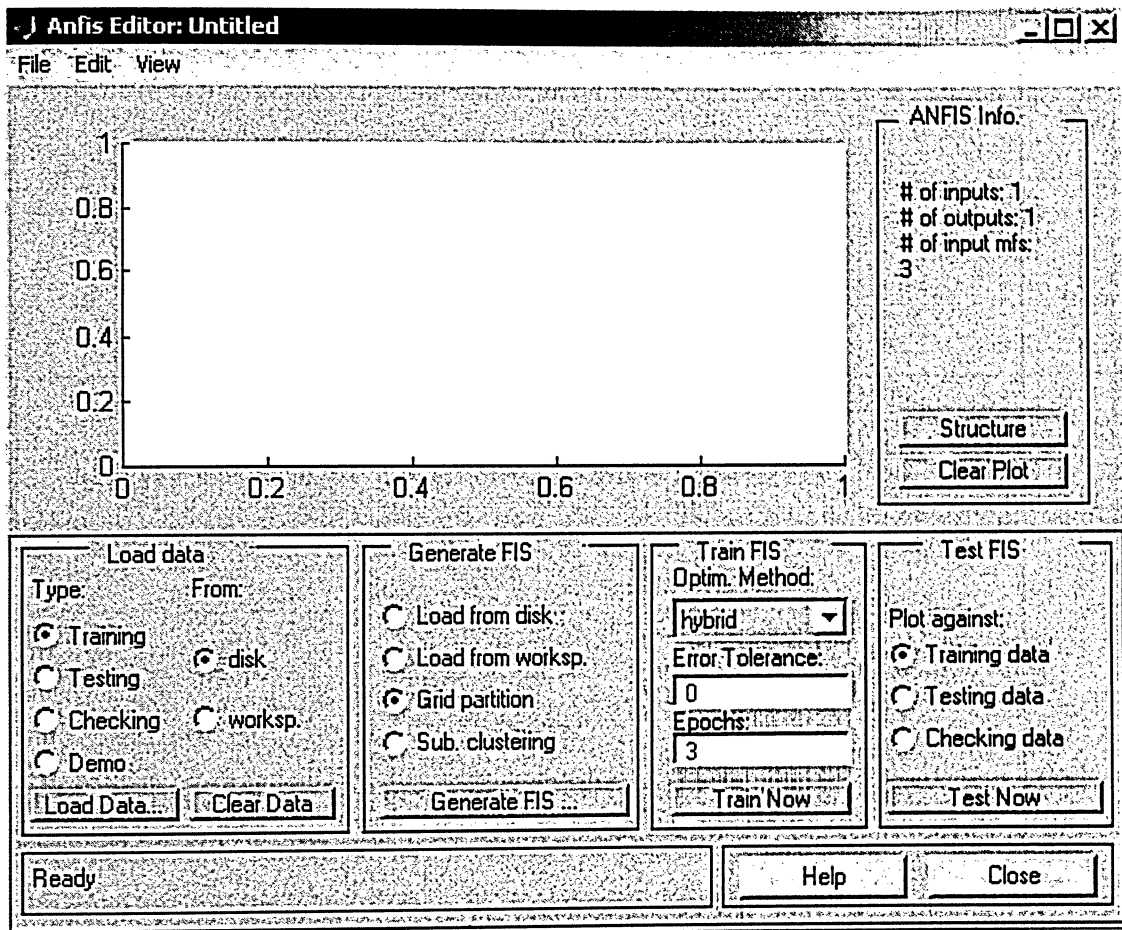


Figure 12: MATLAB® ANFIS GUI.

The data loaded into the ANFIS GUI could be utilized either for training or the validation of the model. It may also be used for checking to avoid over fitting the model during the training. If there is a saved fuzzy structure, as long as it complies with the MATLAB® ANFIS toolbox, it could be called into this GUI for further training or test. As well, an ANFIS structure could be built automatically by the toolbox based on the loaded data.

Once data is loaded, we have two options for the toolbox to generate a new ANFIS structure; the Grade partitioning or sub-clustering. In Grade Partitioning, the default number of membership functions is three per input. Membership functions could be selected from any

of Triangular, trapezoidal, Gaussian, generalized bell, sigmoid, or polynomial curves types. The default is generally set at Triangular. In this step, the default number of the input membership functions and their types could be changed from the default values. Furthermore, the output membership functions are selectable between constant or linear type (zero or first degree Sugeno type). If we choose to use Sub-clustering, the Range of Influence, Squash Factor, and Accept and Reject Ratio could be either defined by the user or remain as the default values.

After generating the fuzzy structure, the trained data will be used to teach the model the indicated error tolerance and the number of training rounds (epochs). The optimization method is also selectable amongst the simple backpropagation and the hybrid of backpropagation and the least square. In the training step, the system adjusts the number, type, and parameters of the membership functions to optimize the output error. As well, the rules are automatically generated by the system. Ultimately, the output of the model could be plotted with the training or test data on the same graph and the average error will be shown at the bottom of the screen. Fig. 13 illustrates a typical Fuzzy Inference System (FIS) structure view in ANFIS GUI. The structure consists of the inputs, input membership functions, the rules, the output membership functions, and the output. The rule viewer could be employed to graphically observe the response of the ANFIS on particular input vectors. Fig. 14 shows an example of this response in ANFIS GUI

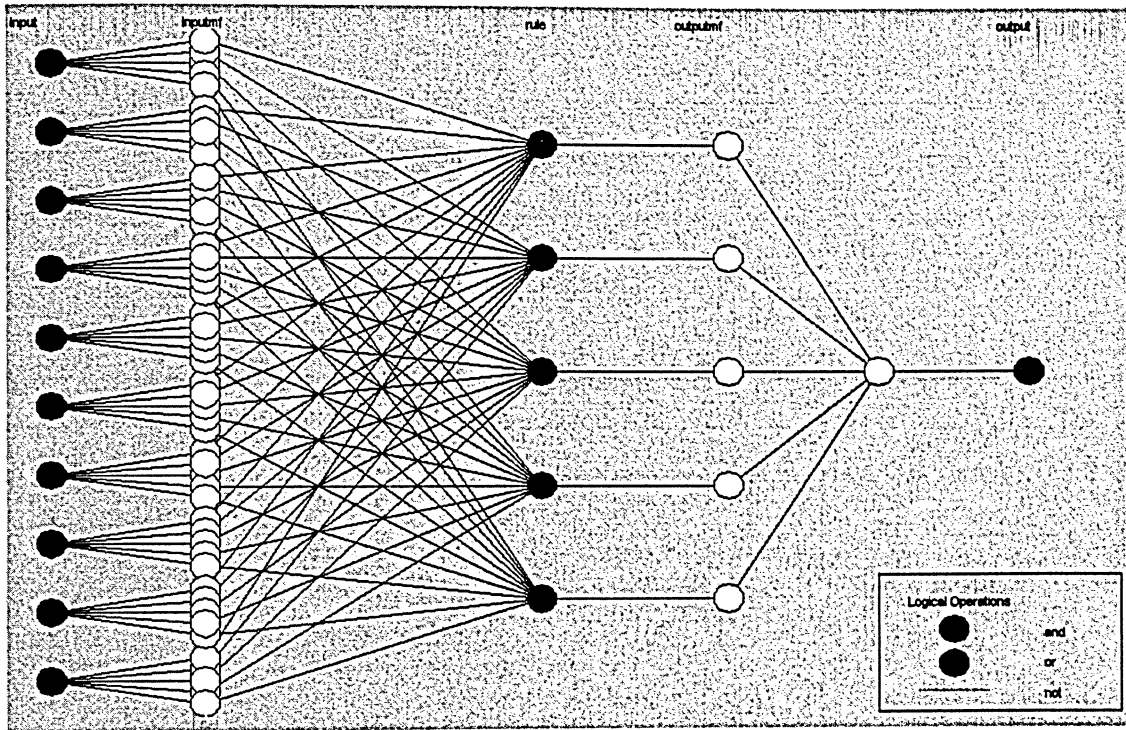


Figure 13: The typical FIS structure generated by MATLAB® ANFIS toolbox.

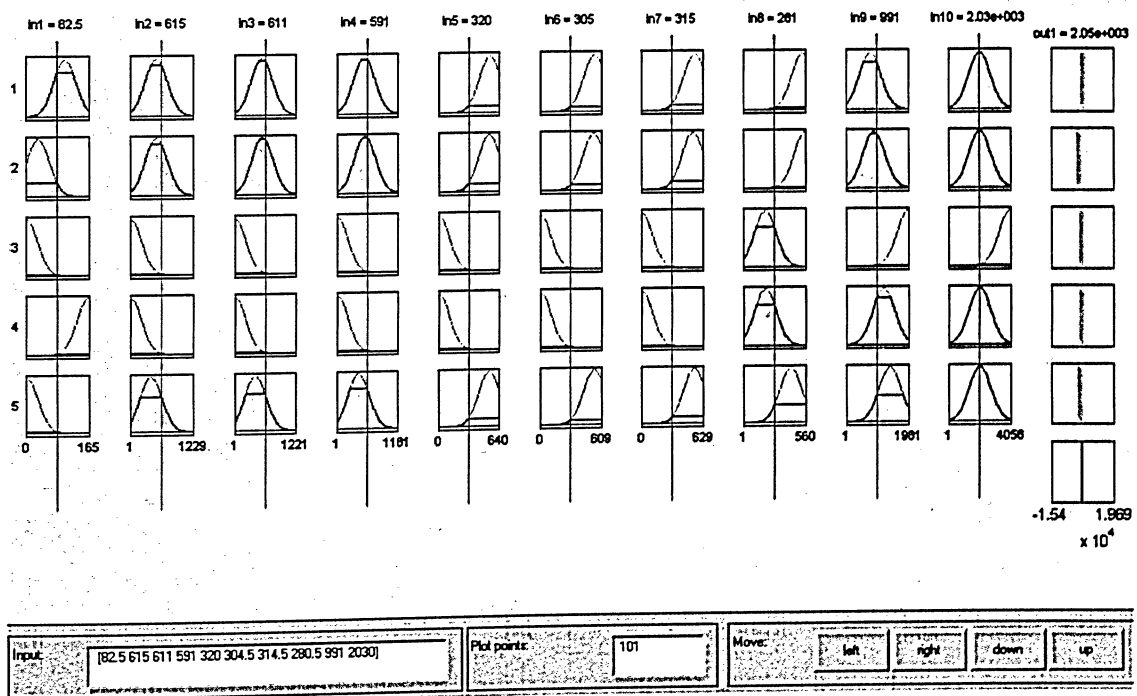


Figure 14: The FIS rule viewer generated by MATLAB® ANFIS toolbox.

Chapter 3

Data Collection

All of the data used in this study has been collected from the EAF in GAW. In this section, the structure of the data acquisition will first be discussed, followed by the list of the significant data used for this analysis. Initially in this research, to understand the process and correlation of the data, the analyses were done on the collected data. In the last section of this chapter, some of the analyses are illustrated with graphs.

3.1 Data Acquisition System

In EAF of GAW, multiple Modicon type Programmable Logic Controllers (PLCs) are responsible for controlling the EAF processes and operations. In the Fall of 2004, a PC-based data acquisition system (ibaTM) was installed in the network with the PLCs to collect the data for this project and other power analysis in EAF. This PC-based data acquisition system is capable of collecting the data through Ethernet TCP/IP. The sample time to collect the data on each group of signals could be adjusted from one ms to hrs. In the first phase of the project, since most of the power data was already filtered and averaged in the PLCs, the decision was made to start with one sec sample time. The sample time in this case was selected experimental. Because the earlier version of the Modicon PLCs cannot communicate through Ethernet TCP/IP, they are all networked together and to the newer version of Modicon PLC (Quantum series) through Modbus Plus protocol. The Quantum PLC then uses its Ethernet TCP/IP port to connect to the PC-based data acquisition PC. In fact, in this network architecture, the Quantum PLC is used just as a gateway to convert the data between Modbus Plus and TCP/IP protocols. Fig. 15 shows the general overview of the data collection network installed in GAW.

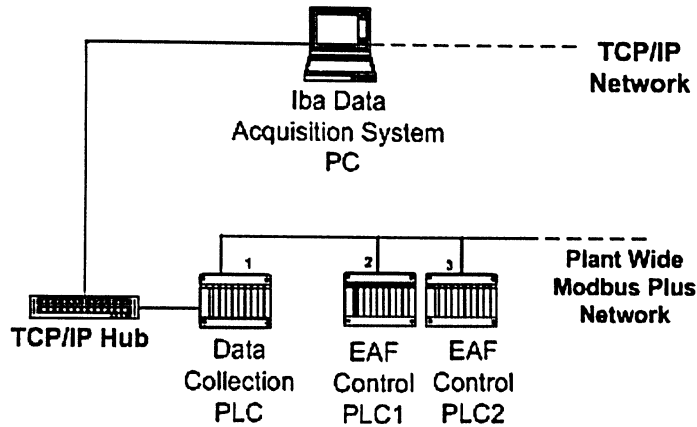


Figure 15: The data acquisition system network diagram.

3.2 Collected Data Items

In the data collection phase of this research, more than one hundred signals have been collected. Among all of these, Table 4 lists the most crucial signals and their corresponding units. The signals' names are the same as the variable names for the signals in the simulator program.

Naming convention: In order to identify each one of the phases in both side of the EAF transformer, the three phases are named 1B, 2A and 3C. This naming convention is used through out this report.

Table 4: The important collected plant signals.

NO	Signal Name	Unit	Description
1	Kwh	KWH	Electrical Energy Consumption
2	MD	Percentage %	Meltdown percentage in each charge of the EAF
3	AltEn	KWH	Alternative form of energy generate heat from combustion in the burners and material oxidization
4	Ph1B_Volt	Volt	Measured voltage of phase 1B in secondary side of the EAF transformer
5	Ph2A_Volt	Volt	Measured voltage of phase 2A in secondary side of the EAF transformer
6	Ph3C_Volt	Volt	Measured voltage of phase 3C in secondary side of the EAF transformer
7	Ph1B_KA	KAmp	Measured active current of phase 1B in the secondary side of the EAF transformer
8	Ph2A_KA	KAmp	Measured active current of phase 2A in the secondary side of the EAF transformer
9	Ph3C_KA	KAmp	Measured active current of phase 3C in the secondary side of the EAF transformer
10	Ph1B Ref	Volt	PLC voltage reference for phase 1B
11	Ph2A Ref	Volt	PLC voltage reference for phase 2A
12	Ph3C Ref	Volt	PLC voltage reference for phase 3C
13	Ph1B_Reg	Integer-Count (0 4095)	PLC regulator output goes to proportional valve of the electrode in phase 1B
14	Ph2A_Reg	Integer-Count (0 4095)	PLC regulator output goes to proportional valve of the electrode in phase 2A
15	Ph3C_Reg	Integer-Count (0 4095)	PLC regulator output goes to proportional valve of the electrode in phase 3C
16	Ph1B_Pos	Integer-Count (0 4095)	The actual position of the electrode in phase 1B read from position encoder
17	Ph2A_Pos	Integer-Count (0 4095)	The actual position of the electrode in phase 2A read from position encoder
18	Ph3C_Pos	Integer-Count (0 4095)	The actual position of the electrode in phase 3C read from position encoder
19	Ph1B Re KA	KAmp	The measured reactive current for phase 1B
20	Ph2A Re KA	KAmp	The measured reactive current for phase 2A
21	Ph3C Re KA	KAmp	The measured reactive current for phase 3C
22	Ph1B MW	MW	The active power measured for phase 1B
23	Ph2A MW	MW	The active power measured for phase 2A
24	Ph3C MW	MW	The active power measured for phase 3C
25	Ph1B On	No Unit Binary(0/1)	Current flowing into the phase 1B
26	Ph2A On	No Unit Binary(0/1)	Current flowing into the phase 2A
27	Ph3C On	No Unit Binary(0/1)	Current flowing into the phase 3C
28	Ph1B Cos	No unit, 0-1	The calculated $\cos \phi$ for phase 1B (Power Factor)
29	Ph2A Cos	No unit, 0-1	The calculated $\cos \phi$ for phase 2A (Power Factor)
30	Ph3C Cos	No unit, 0-1	The calculated $\cos \phi$ for phase 3C (Power Factor)

3.3 The Process Analysis Based on the Collected Data

In this section, we will review several snapshots of graphs captured from the data acquisition system to better understand the process and data correlations.

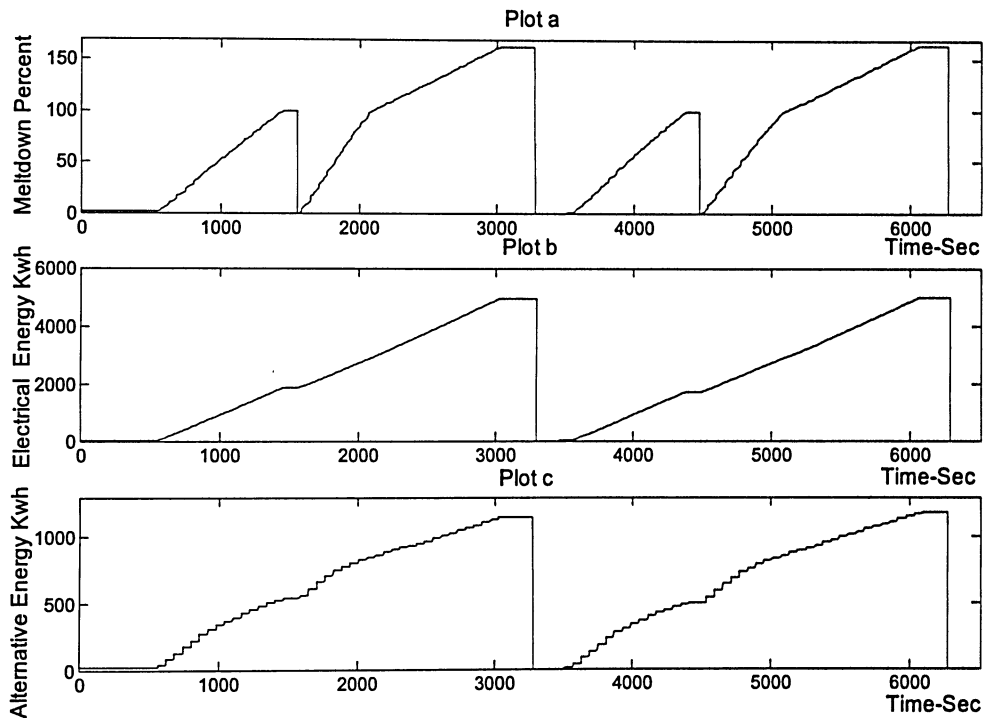


Figure 16: Energy consumption during the melting process.

In Fig. 16(a) illustrates meltdown percentage, Fig. 16(b) shows the electrical energy (Kwh) on and Fig. 16(c) represents the alternative energy (Kwh). The meltdown percentage includes both electrical and chemical energy. The total energy (Kwh) to melt down every ton of metal scrap is a known number. Based on the measured scrap weight dumped in the EAF and the measured consumed energy (Kwh), the meltdown percentage is calculated. Fig. 16 also shows the two Heats in a row with each of them having two Charges of the scrap dump. The meltdown percentage for the first Charge initiates at zero and progresses to 100. By reaching 100 percent meltdown, the first charge concludes. Following a short delay, the crane dumps

more scrap metal into the EAF and the melting process resumes. As we could see from this figure, the KWh, as a result of the meltdown percentage in the second charge, rapidly reaches 100 percent. The reason for this is that, in the second charge, the EAF is already carrying a high volume of molten steel. With a close inspection of Fig. 16, it is also obvious that the melting process will continue even after reaching meltdown level of 100 percent. This is due to the fact that subsequent to the 100 percent meltdown, the power is still consumed by electrodes and burners to refine and reach the desired metallurgical property of steel. This stage of the melting process is called the refining stage.

Furthermore, it can be concluded from Fig. 16 that the chemical (alternative) energy in the refining stage is not increasing as rapidly. This is because the steel is already molten and the burners' flame cannot be as effective in the process. Nevertheless, a substantial part of the chemical energy contribution comes from the oxidization, which is more active in the refining stage as a result of the extra oxygen injected into the molten batch.

In Fig. 17(a) demonstrates the meltdown percentage, Fig. 17(b) represents the voltage feedback, and Fig. 17(c) shows the voltage reference. The graphs cover two Charges of a Heat. By first observation of these graphs, it is apparent that the voltage reference profile changes in various stages of the melting process (based on the meltdown percentage). The voltage feedback attempts to follow the reference with lots of spikes, with its ability dependent on how good the regulator loop controls the electrodes. The function of the regulator loop was explained in Chapter 1. It can also be observed from this graph that the voltage reference profile is slightly different between the two charges. This profile is from a lookup table and was again explained in Chapter 1.

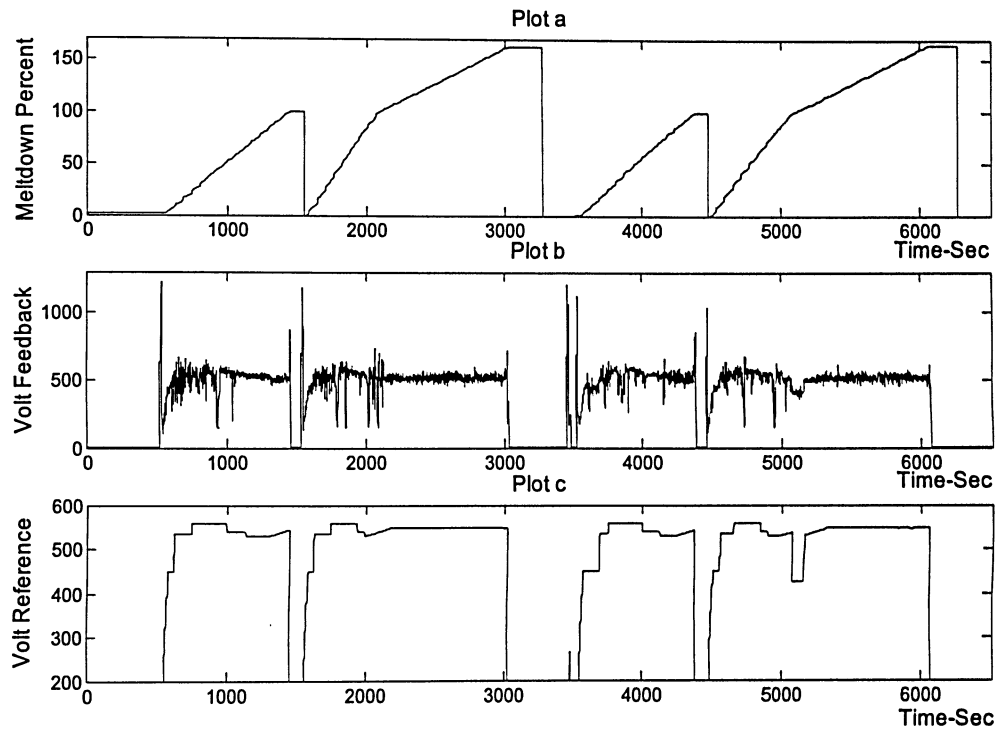


Figure 17: Voltage signals' behaviors during the melting process.

In Fig. 18(a) shows the power factor, Fig. 18(b) represents the active current (KA), Fig.18(c) illustrates the reactive current (KA). The graph is for the period of two Charges of a Heat. The first observation from this graph indicates that at the beginning of both Charges, the reactive power is at its maximum level and as the melting progresses it decreases significantly. This is because, the nature of the scrap metals at the beginning, it is more inductive but as the scrap is melted, it becomes more resistant load. It is also illustrated in this graph that the power factor for the majority of the process remains above 0.9. There are other components in the power circuit, such as capacitor banks, to assure the high power factor and avoid forced penalties from hydro due to poor power factor.

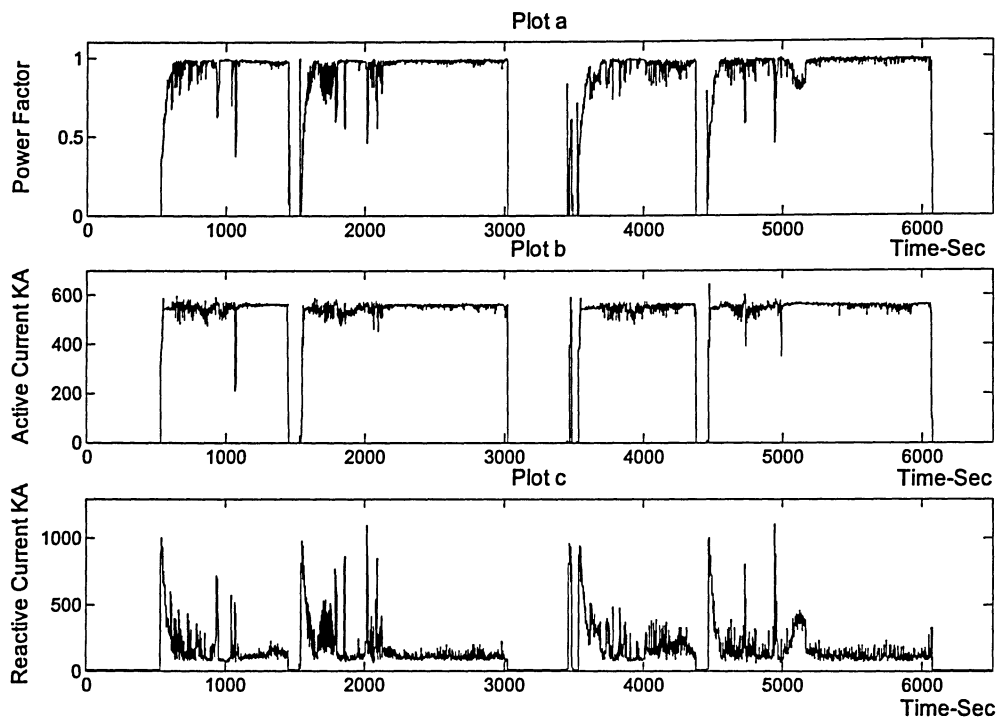


Figure 18: Current and power factor signals behaviors during the melting process.

Fig. 19(a) illustrates the meltdown percentage, Fig. 19(b) shows the regulator output and Fig. 19(c) illustrates the electrode's actual position. Once again, the graph covers the two Charges of one Heat. The unit for regulator output is 12 bits integer number (0-4095), equivalent to ± 10 Volt reference for the proportional valves. The proportional valves control the flow of the hydraulic and, as a result, the movement of the electrodes. As well, attached to each electrode mass, there is a *position encoder* to track the movement of the electrodes. As observed in this graph, the electrodes are moving free at the beginning of the movement. The feedback quickly follows the reference and subsequently slows down once the electrodes come in contact with scrap. Further, the movement depends on the difference between the voltage reference and voltage feedback (the phase voltage error). The regulator controls the up or down movement of electrodes according to the phase voltage error. The arc length will follow as the distance between the electrodes and scrap will change as well.

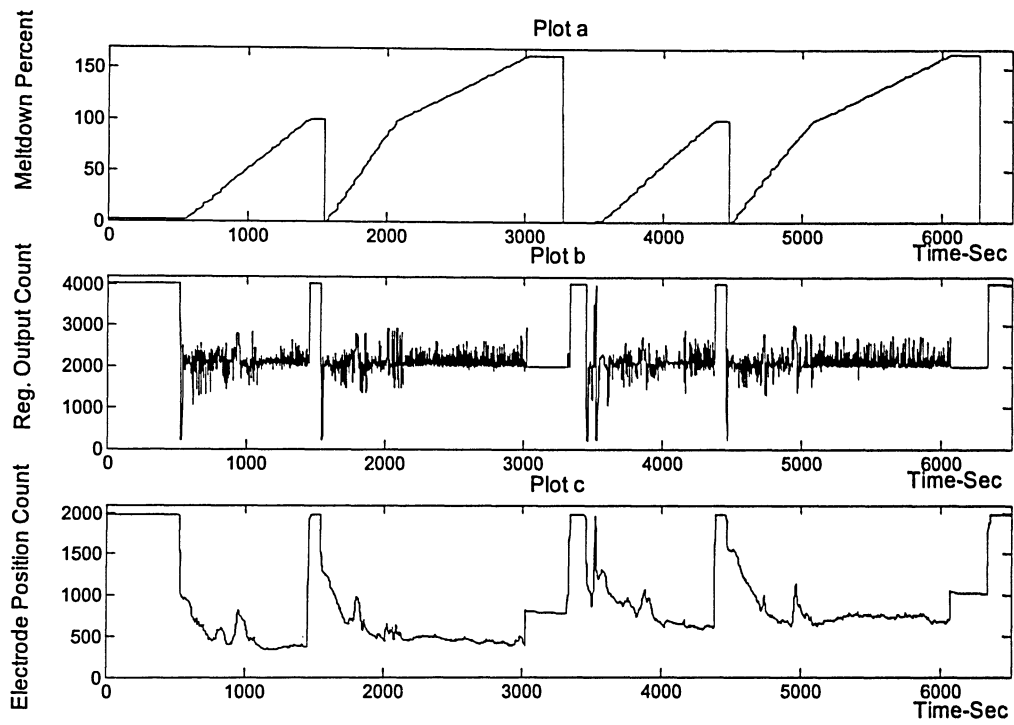


Figure 19: The graph of the electrode movement during the melting process.

Chapter 4

The Simulator Structure and Implementation

This chapter discusses the implementation of the simulator components in MATLAB® Simulink® environment. In the first section, the structure and the roles of each component are described. The following sections explain more details in regards to implementing each component. The snapshots from some of the Simulink® blocks, which are developed in this project, are presented to explain each of the components. Please note that the snapshots illustrate part of the logic and were not meant to be used as program printout.

4.1 The Overview of the Structure and Components

Fig. 20 is an overview of the components and their interconnection in the simulator for one of the phases. The same structure repeats for the remaining two phases. The most vital part of the simulator is the section that the measured signals from the plant. This section utilizes three individual models for measured active current, reactive current, and voltage. These models have been built and trained outside of Simulink® by the use of MATLAB® fuzzy toolbox. The outputs of these components are used to calculate the active power consumption in the phase.

The summation of power consumed in this phase with the power consumption of the other two phases, results in the overall power consumption. The consumed electrical power is an indicator for progress of melting. Therefore, it has been applied as a pointer in the lookup table for capturing the voltage reference and PI coefficient in every stage of the melting. The

regulator loop, which is a classic adaptive PI controller, uses the voltage error as its input and transmits a reference to position the electrode. The two components for the regulator PI control loop and the total power consumption are simply classic control and calculation routines. The classic control components of the simulator were programmed to be quite similar to their implementation in the PLC program. The regulator output and voltage reference of each phase are used as inputs in all ANFIS models.

The EAF sequential event simulator uses the total three-phase power consumption and the simulator clock to generate the main events involved in the melting process. In the following sections, some more details of each component in the simulator are discussed.

4.2 Measured Values ANFIS Models

The ANFIS blocks were generated and trained outside of Simulink® by the use of MATLAB® ANFIS toolbox. For the training of these models eight batches of data (each batch representing one complete Heat, about one hour) were used. ANFIS models for active current, reactive current, and voltage feedback all utilize the voltage reference and regulator output for the same phase as their inputs. Fig. 21 illustrates an overview of these blocks in Simulink®. As apparent in this figure, the reactive current ANFIS model, in addition to the voltage reference and regulator output, also uses the active current of the same phase as an input. The complete details of these models as well as the block diagrams showing their inputs and outputs are shown in Appendix A.

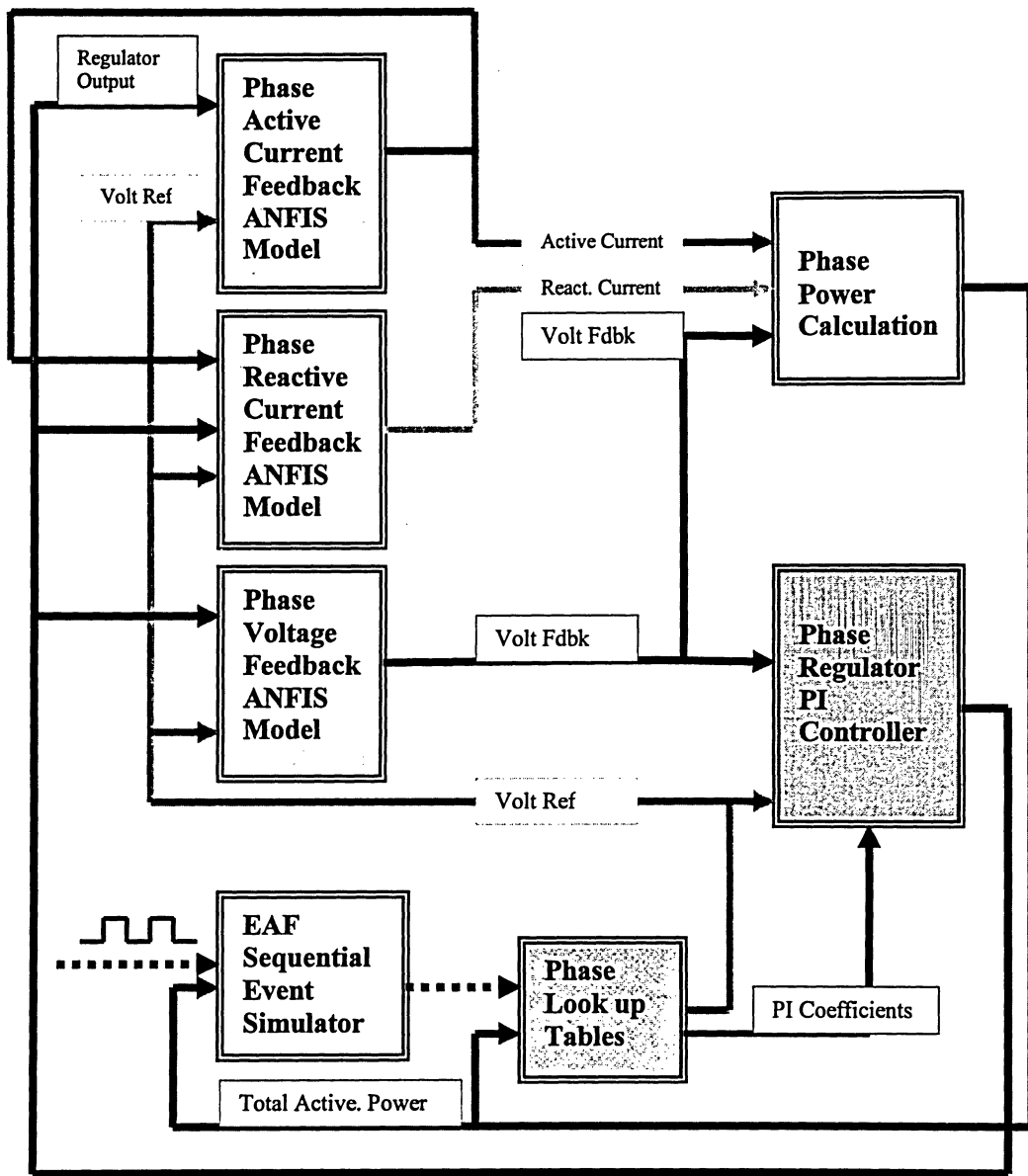


Figure 20: Components of the simulator for one of the phases and their interconnections.

Figs. 22, 23 and 24 are snapshots from the inside of these blocks. As could be observed from each of these blocks, the FIS block is the core of each intelligent simulator component. As well, in all of the ANFIS models, the pervious state of the output is used as the input of the current state. Another common factor among all of these models is the high and low limit check. If the output of the model is progressing above or below the set limits, it would be limited to the permitted maximum value. The values of the high and low limit have been captured from the practical plant data.

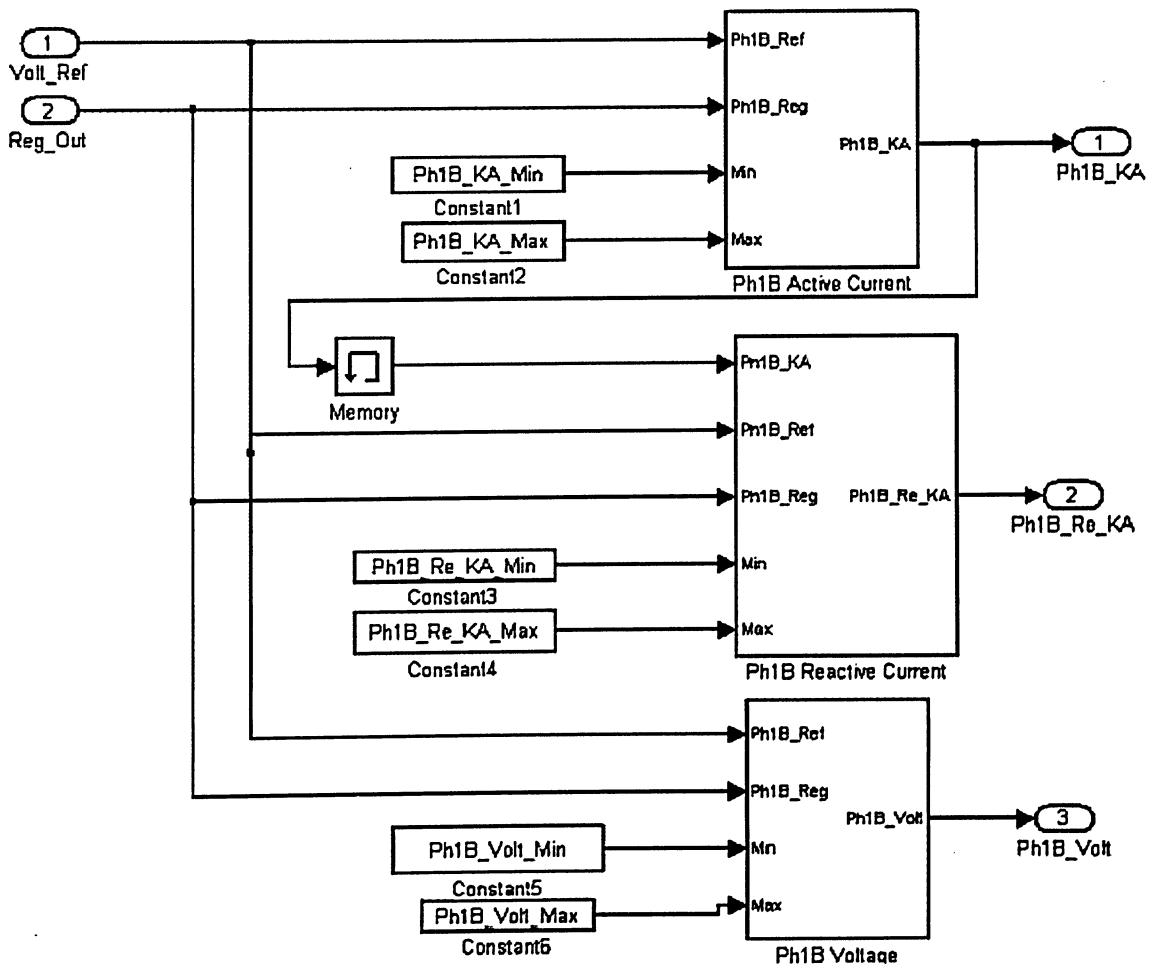


Figure 21: ANFIS models for EAF measured signals.

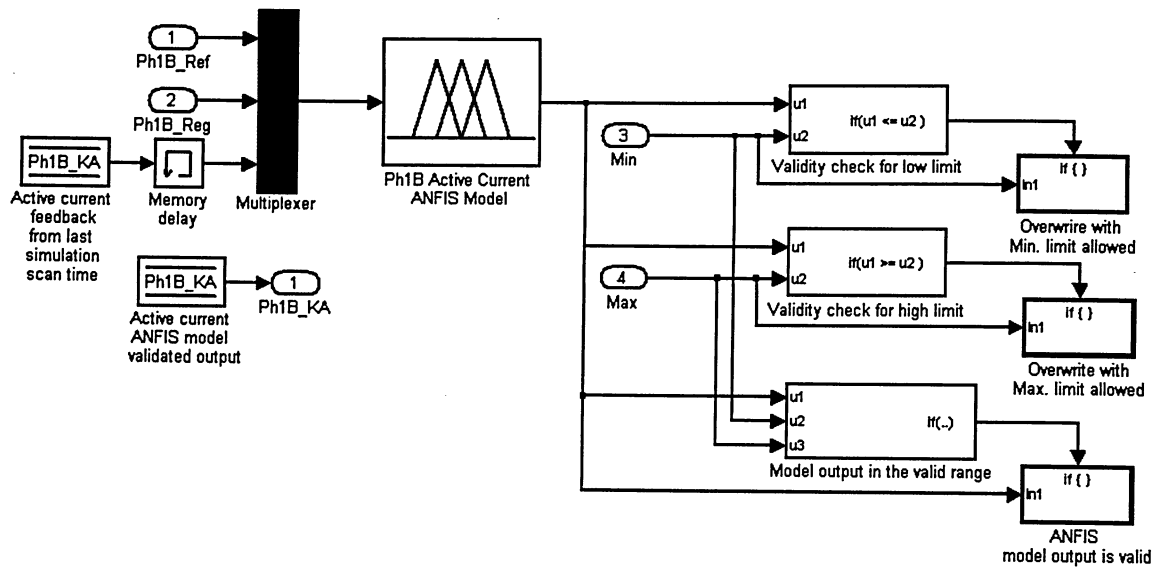


Figure 22: Phase 1B measured active current ANFIS model.

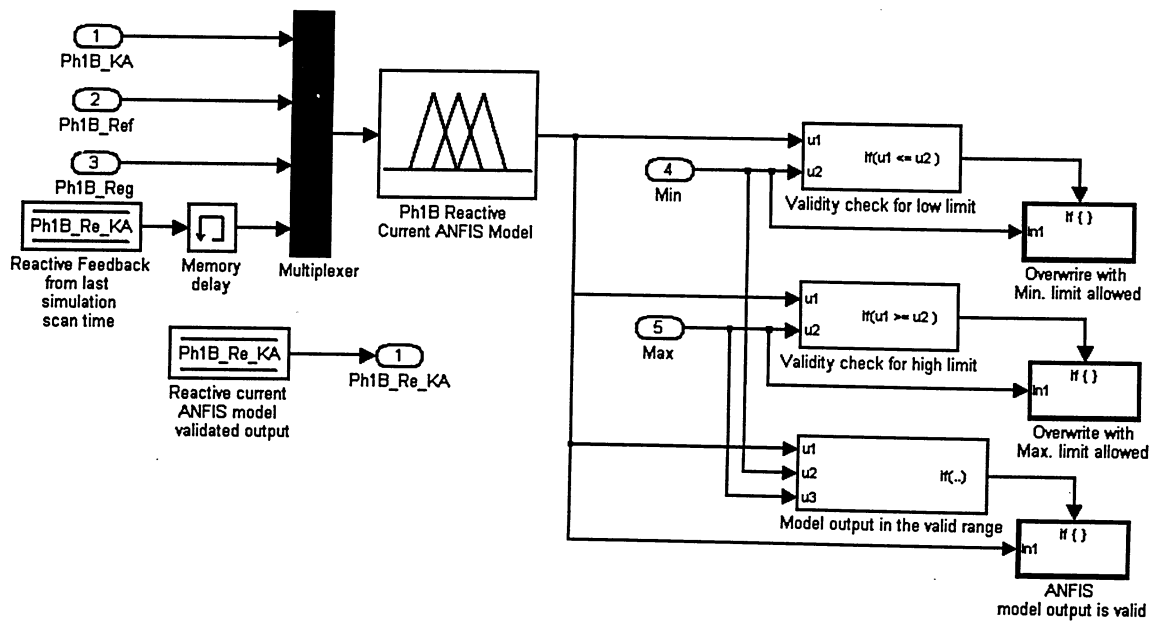


Figure 23: Phase 1B measured reactive current ANFIS model.

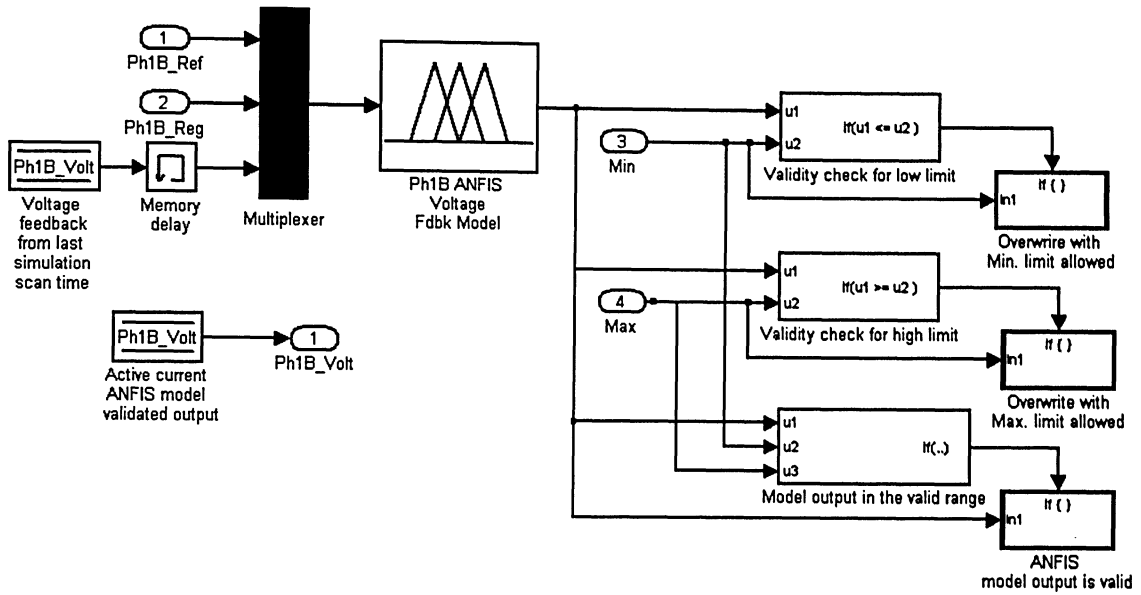


Figure 24: Phase1B measured voltage ANFIS model.

4.3 The Simulated Regulator Control

The present regulator controller in the EAF of GAW uses a PI controller. The coefficients of the PI vary in different stages of the meltdown. These coefficients are captured from the lookup table. In this fashion, we could call the control classic adaptive PI control. Fig. 25 illustrates an overview of the regulator loop block. It uses the voltage reference and voltage feedback of the phase to generate the voltage error. The voltage error, then, is applied to the PI controller. The total power consumption and current feedback are used to detect the different stages of meltdown and switch the PI coefficients and voltage reference from one set to the other.

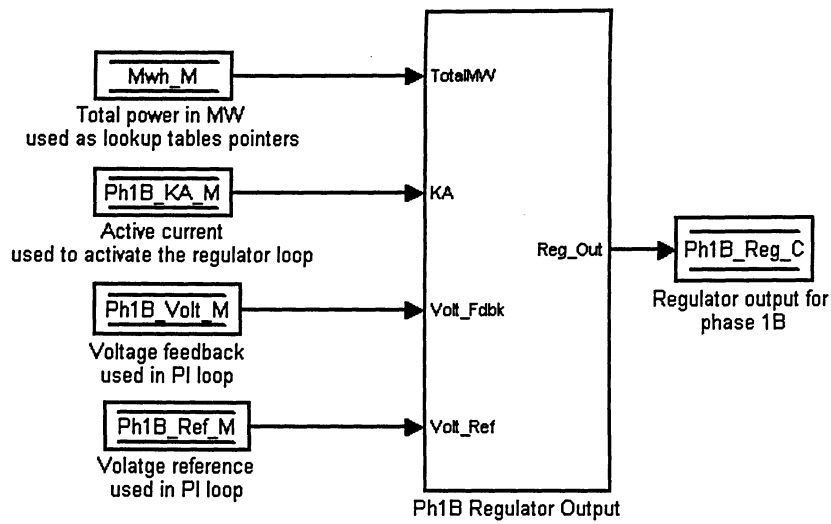


Figure 25: Phase 1B regulator Simulink® block overview.

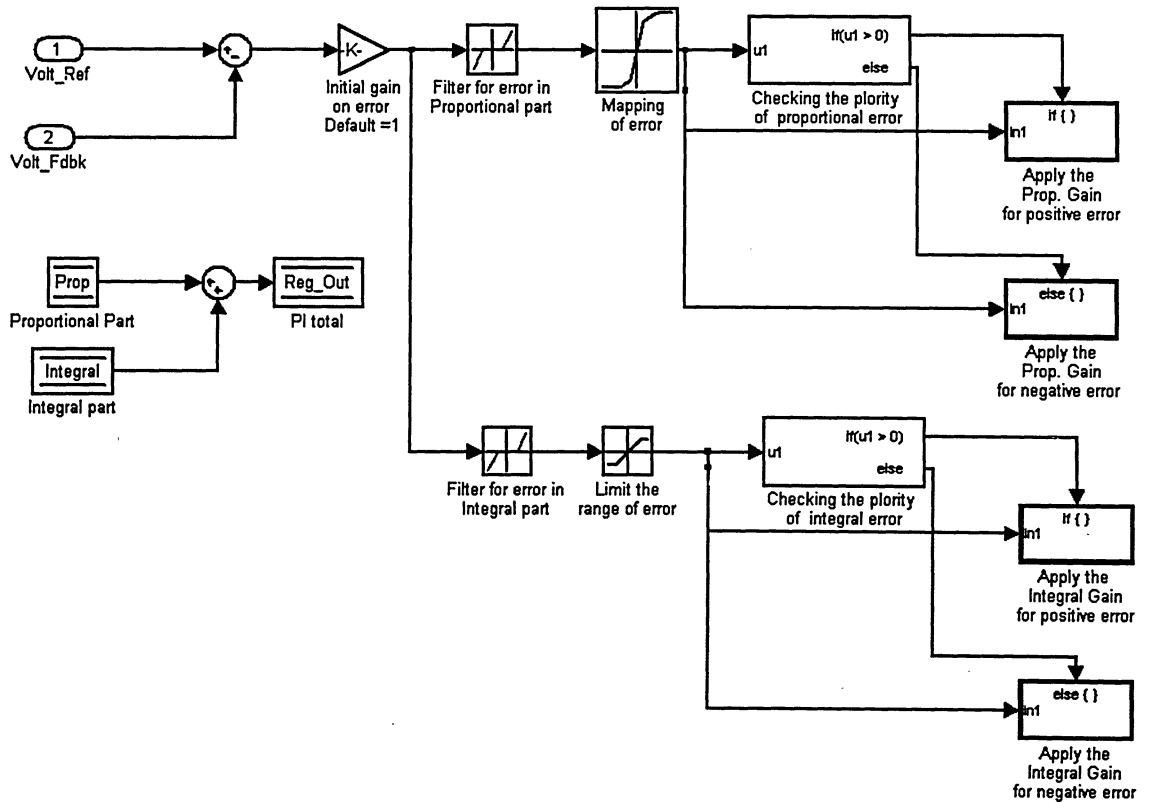


Figure 26: Phase 1B regulator PI controller.

Fig. 26 is an illustrations of the inside view of the PI controller. As apparent, the error before applying the PI controller is normalized and limited in the acceptable range. As well, the output of the PI controller is clamped and scaled. The values to normalize and clamp the voltage error are retrieved from the lookup table and the parameters slightly vary in every stage of the melting. The output of the PI controller is scaled in 12 bit number (0-4095). In real control systems in plant, these values are forwarded to the analog cards to generate a control voltage (-/+10V) to the proportional valves. The values control the flow of the hydraulic fluid into/from the cylinders, which the electrodes are resting on, and cause the electrodes to move vertically.

4.4 Power Consumption Calculation

In Fig. 27, the Simulink® block to calculate the power factors for each one the phases is illustrated. The calculation within the block is based on the basic power equation of:

$$PF_{phase} = \cos \varphi_{phase} = \frac{I_{act_{phase}}}{\sqrt{I_{act_{phase}}^2 + I_{react_{phase}}^2}} \quad (45)$$

In Eq. (45), PF represents the power factor for each of the phases or $\cos \varphi$, I_{act} and I_{react} represent the active and reactive current.

Fig. 28 shows another power calculation block which simply uses the power factor and calculates each one the phase's real power (active power) based on:

$$P_{real_{phase}} = V_{LN_{phase}} I_{act_{phase}} PF_{Phase} \quad (46)$$

In Eq. (46), P represents the power and V_{LN} represents the voltage of the line to ground for the phase. Ultimately, the total power is calculated by adding the power of all the three phases.

$$P_{real_{total}} = P_{real_{phase1B}} + P_{real_{phase2A}} + P_{real_{phase3C}} \quad (47)$$

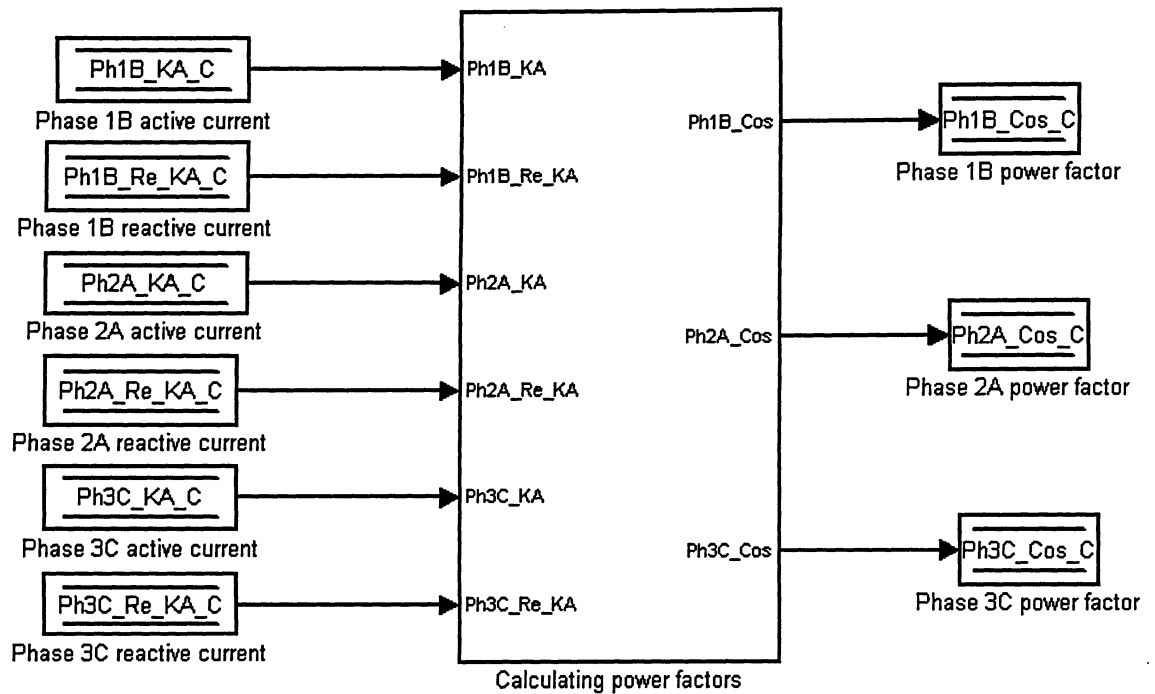


Figure 27: Power factors calculation block.

In order to apply the total real power as the indicator of the meltdown progress, the value is reset at the end of each charge. As well, to obtain the total real power closer to the behavior of the meltdown percentage, a residual gain is defined to replace the effect of the alternative (chemical) energy. The residual gain modifies following the 100 percent meltdown in the second charge. This is because, in the actual process, the generated heat from burners is less effective in the meltdown process after achieving 100 percent meltdown.

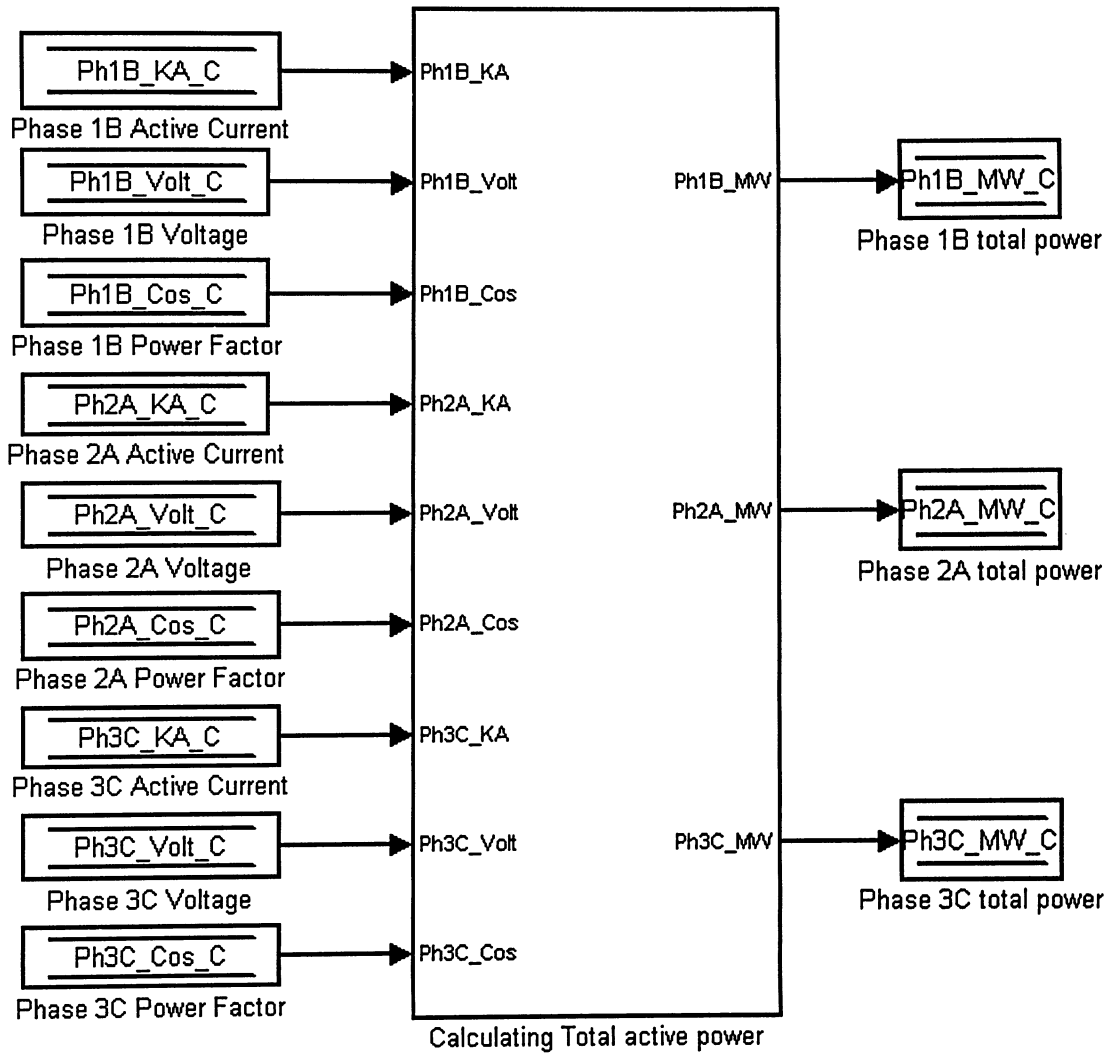


Figure 28: Active power calculation.

4.5 The Discrete Event Generator

The purpose of the discrete event simulator is to simulate the sequence of the events that initiate and control the progression of melting. In the event sequencer, an internal variable known as state has been used to store different stages of the meltdown. The remaining parts of the simulator use the state value to synchronize their activities. In Fig. 29, an overview of the decision making process based on the simulator state variable is shown.

The state variable's initial value is zero, which translates waiting time between the Heats. In state zero, all three electrodes are retracted at the top. Also, in this state, the voltage and current are zero and the regulator loop is inactive. Subsequent to a defined time delay (approximately 8 Minutes), the state automatically converts to one. In state one, the electrodes are forced to retreat down, and the voltage and current for all three phases are reported with initial non-zero values. In this state, the melting process is initiated for a short period of time, then allowing the ANFIS models to take over the system. In the actual process, this state implies that the operator would request to close the vacuumed breaker switch. By closing the switch, the secondary of the EAF transformer connects to the electrodes and, consequently, lowers the electrodes. State one is rather brief and lasts for approximately 10 seconds. Following this time delay, the state variable will be set to two. In state two, the ANFIS models used to simulate the feedback values for three phases as well as the regulator PI controllers are active. The regulator PI controller and voltage reference use the total real power consumption as the pointer in their lookup table. The lookup tables are slightly dissimilar between first and second charges. The voltage reference lookup table presentation in Simulink® for the first and second charges is shown in Figs. 30 and 31. When the total real power consumption in the Charge passes a predefined threshold, this state concludes. State two ends in the total MW that represents 100 percent meltdown for the first Charge. For the second Charge, this state will extend longer to provide time for mixing the alloys and purification. If it is the first Charge, and according to the power consumption the meltdown is completed, the state changes to three. In the third state, the electrodes will retract for the new scrap dump and begin the second Charge by changing the state variable set back to two. If the Charge number is at two when the second state concludes, the state variable resets to zero and Charge number to one. It indicates that one simulation cycle has concluded and the simulator initiates the next Heat.

The timing as well as the initial values for this event sequencer have been captured from actual data and intend for the simulator to perform comparable to the actual process. This process is repeated periodically for the duration of time that the simulation is in operation. In

this stage, random factors are not introduced into the system. This signifies that everything is assumed to follow a normal automatic batch melting process, which could be true for more than 50 percent of the batches. Of course, in the real process, there are occasions when operators interrupt the automatic process for various reasons, which have been excluded from the interest of this study.

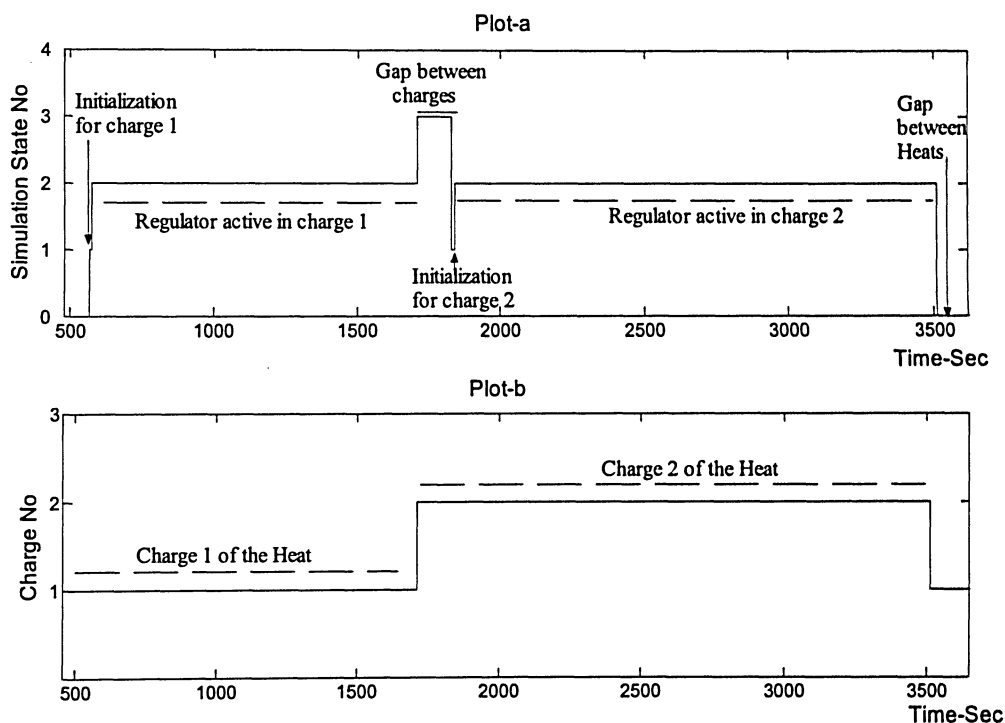


Figure 29: Sequential event simulator.

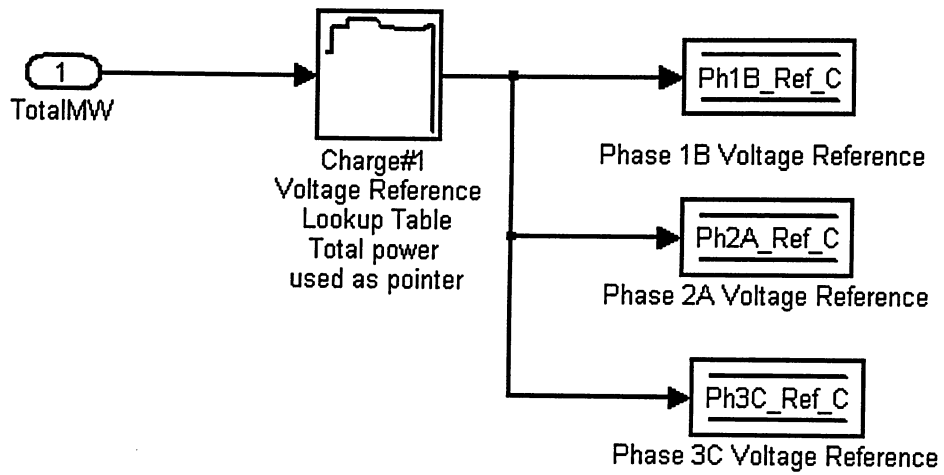


Figure 30: Voltage reference lookup table for the first charge.

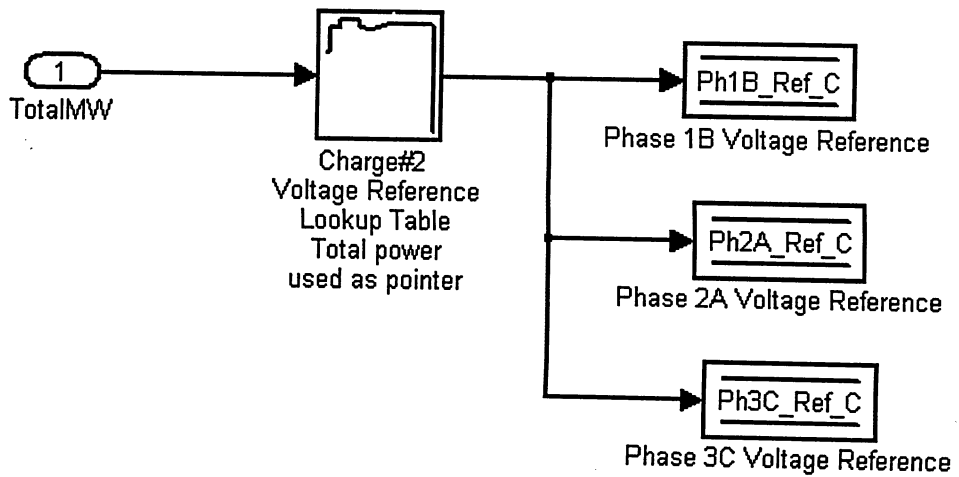


Figure 31: Voltage reference lookup table for the second charge.

Chapter 5

Model Verification, Results and Discussions

5.1 Model Verification Procedure

The simulator has been verified in two different steps. The first step was focused to verify each individual component of the simulator (open loop verification). In this step the plant recorded data was used, and the event sequencer component of the simulator was disabled to synchronize the simulator with the plant data. In the second step, the simulator was disconnected from the plant data and the event sequencer was enabled to initiate and control the process (closed loop verification). In this step, also the behavior of the system was observed and recorded. The results and errors are illustrated in the next sections of this chapter.

5.2 Verification Process for the Simulator Components

One batch of data (the complete Heat, about one hour) among the eight batches of data that ANFIS model had been trained on, was fed to the simulator. The plant data was used as input to the ANFIS models and regulator PI controller. Then, the simulator output for active current, reactive current, voltage and regulator output were compared with the associated values from the plant. The errors were calculated in both average and Root Mean Square (RMS) format. The average error was calculated with adding the errors of each input samples, divided by the number of samples.

In the second step, a set of data (the complete Heat, about one hour) that ANFIS models had never seen during the training (verification data) was selected. Then the model verification process was repeated on this set and the errors again were measured.

The summary of errors for each of the ANFIS models, as well as regulator output for all three phases are tabulated. Tables 5 and 6 show, these errors for the set of training and verification (unseen) data, respectively.

Table 5: The error summary for the set of the training data

Model Output	Average Error %	RMS Error %
Active Current Feedback for Phase 1B	2.6	6.8
Active Current Feedback for Phase 2A	2.1	4.7
Active Current Feedback for Phase 3C	2.4	7.6
Reactive Current Feedback for Phase 1B	4.4	9
Reactive Current Feedback for Phase 2A	4	8.2
Reactive Current Feedback for Phase 3C	4.8	11.3
Voltage Feedback for Phase 1B	2.9	7.4
Voltage Feedback for Phase 2A	4.1	11.8
Voltage Feedback for Phase 3C	3.2	8.2
Regulator Output for Phase 1B	2.8	6.5
Regulator Output for Phase 2A	3.2	7.3
Regulator Output for Phase 3C	3	5.9

Table 6: The error summary for the set of the verification data

Model Output	Average Error %	RMS Error %
Active Current Feedback for Phase 1B	2.6	6.7
Active Current Feedback for Phase 2A	2.5	6.4
Active Current Feedback for Phase 3C	2.2	7.4
Reactive Current Feedback for Phase 1B	4.4	9.8
Reactive Current Feedback for Phase 2A	5.2	10.2
Reactive Current Feedback for Phase 3C	4.6	11.1
Voltage Feedback for Phase 1B	2.9	7.4
Voltage Feedback for Phase 2A	4.8	14.1
Voltage Feedback for Phase 3C	3.1	9.6
Regulator Output for Phase 1B	2.8	7.6
Regulator Output for Phase 2A	3.2	8.3
Regulator Output for Phase 3C	2.9	7.6

Figs. 32 and 33 illustrate bar graphs to visually compare the calculated errors tabulated in Tables 5 and 6. These figures show an acceptable level of error among both training and verification data for ANFIS models. We could also observe that results of the verification data are comparable with the training data. Therefore we could conclude training. In the next section, the model output and plant data have been plotted together for the set of verification data.

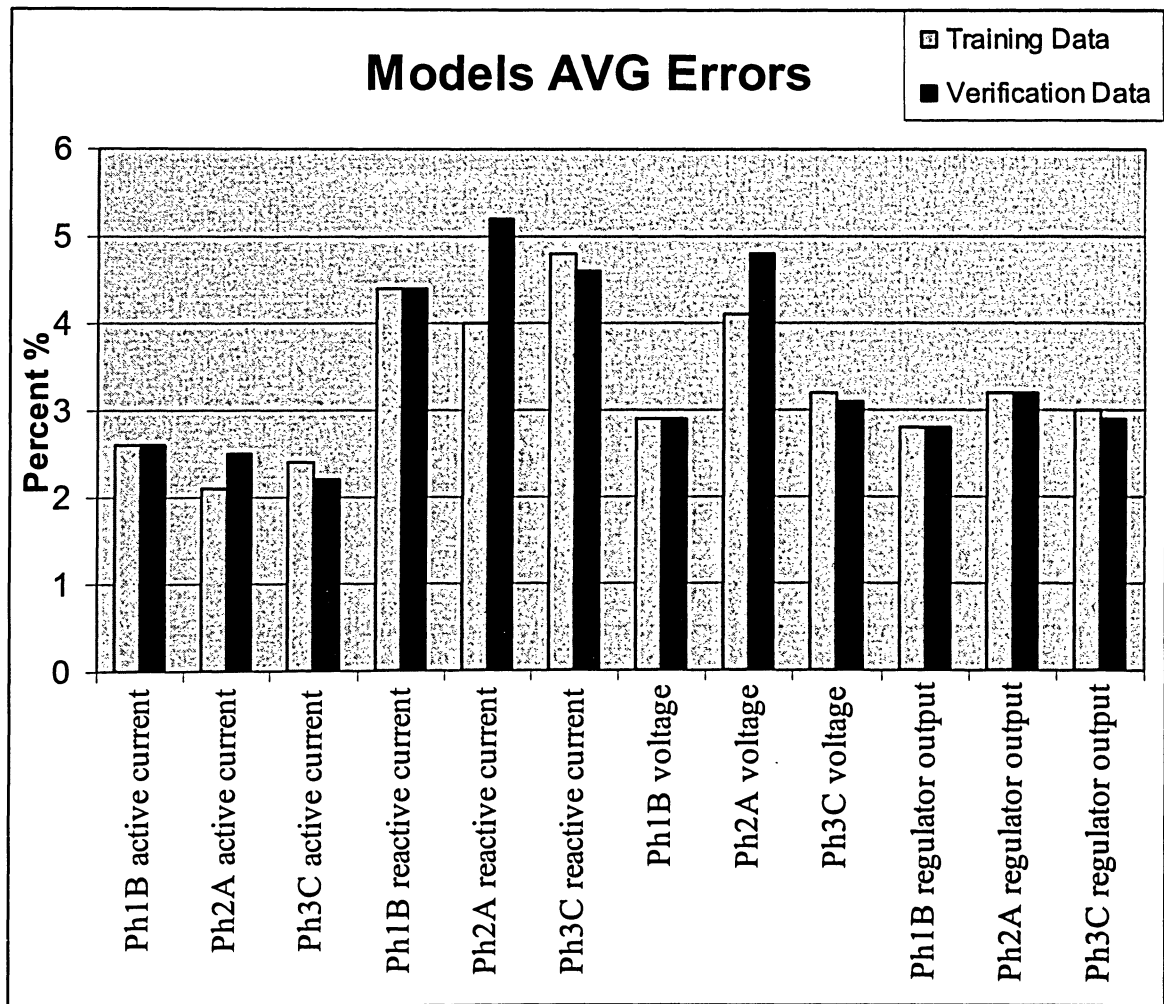


Figure 32: Model verification average error bar graphs.

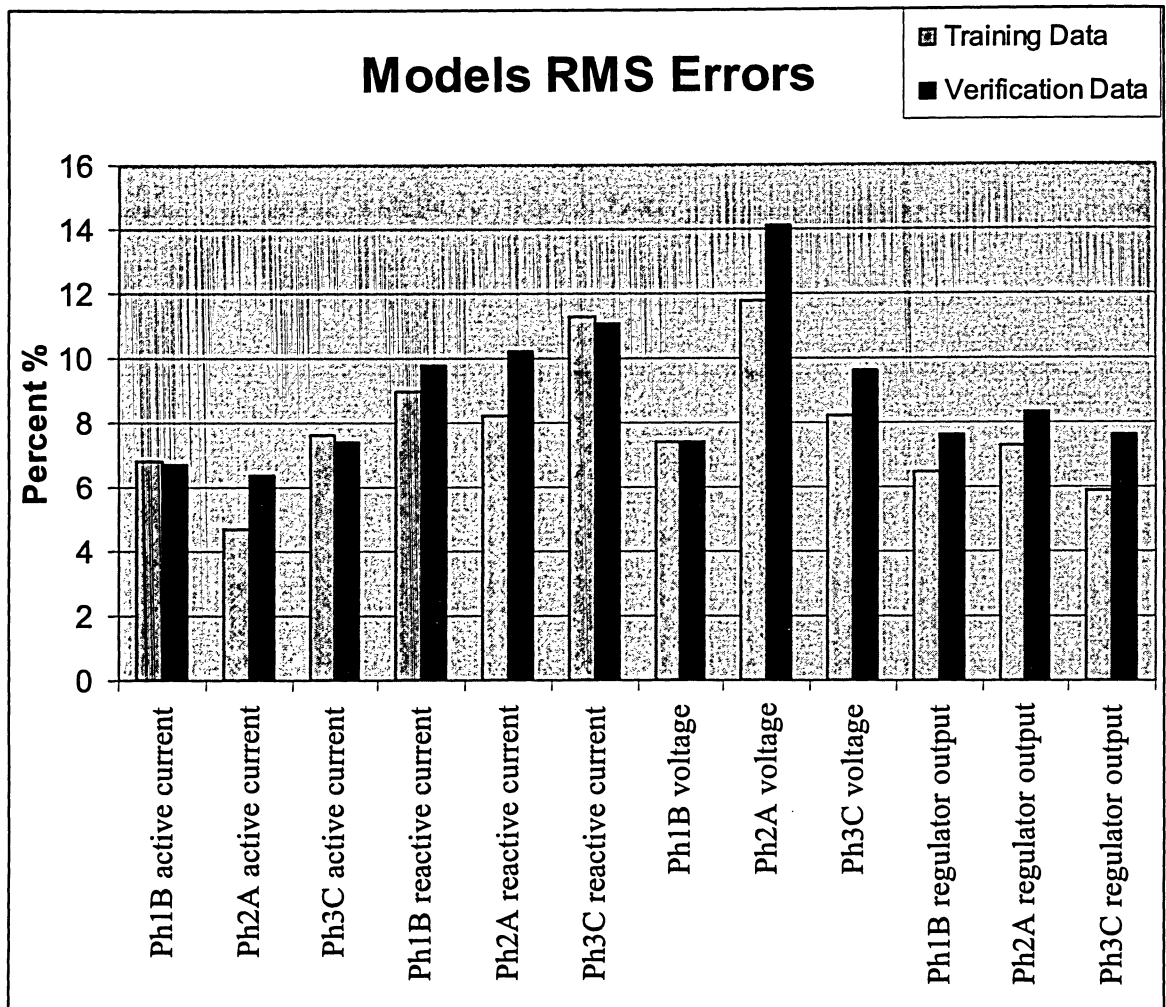


Figure 33: Model verification, RMS error bar graphs.

5.3 Model Outputs Graphs

Figures 34, 35, 36, and 37 illustrate the result of running the simulation on a set of the verification data together with the actual plant recorded data for a period of one Heat (one complete melting batch about an hour). All of these graphs indicate a very close behavior from the simulator. Generally, all of the ANFIS models perform in the same level of the plant

signals, only smoother. Smoothness of the ANFIS models could be explained with the fact that data clustering were used in generating their ANFIS structure.

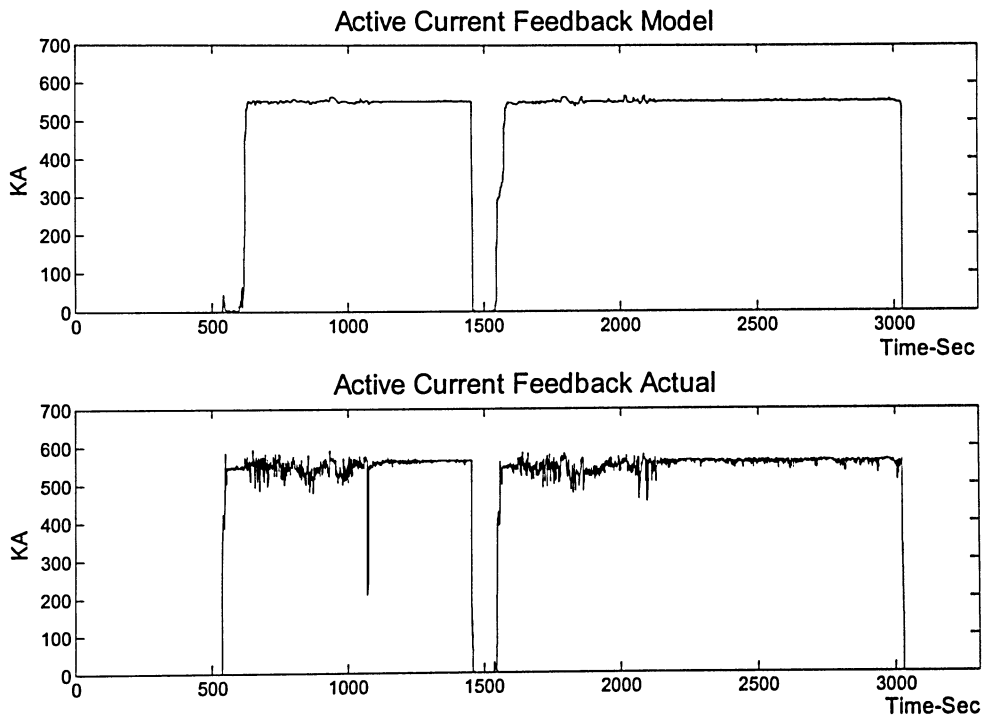


Figure 34: Active current, model output and plant measured value graphs.

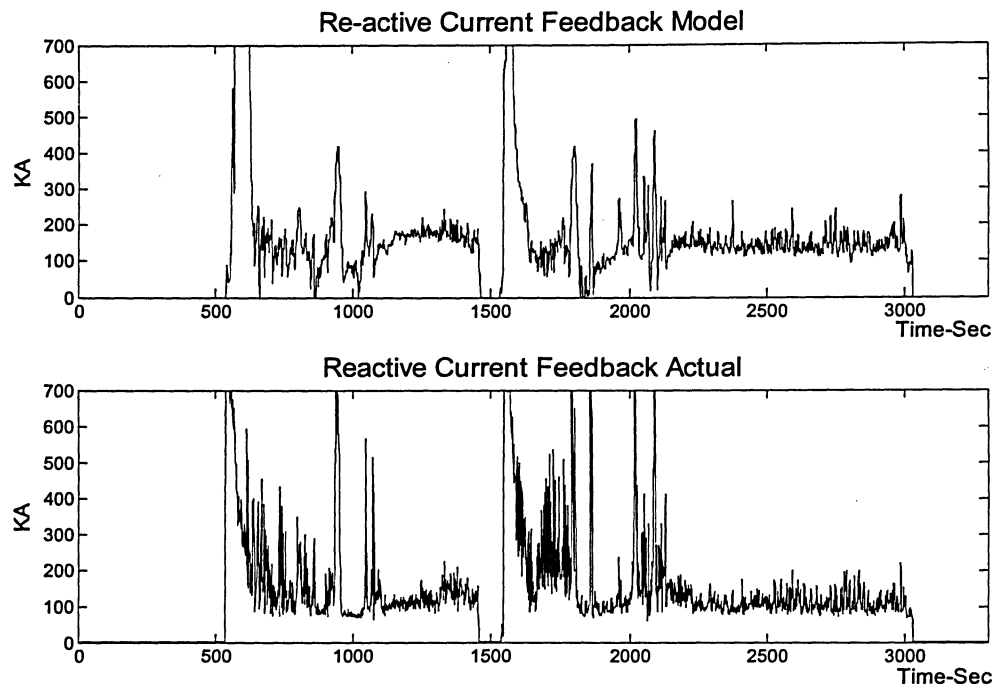


Figure 35: Reactive current, model output and plant measured value graphs.

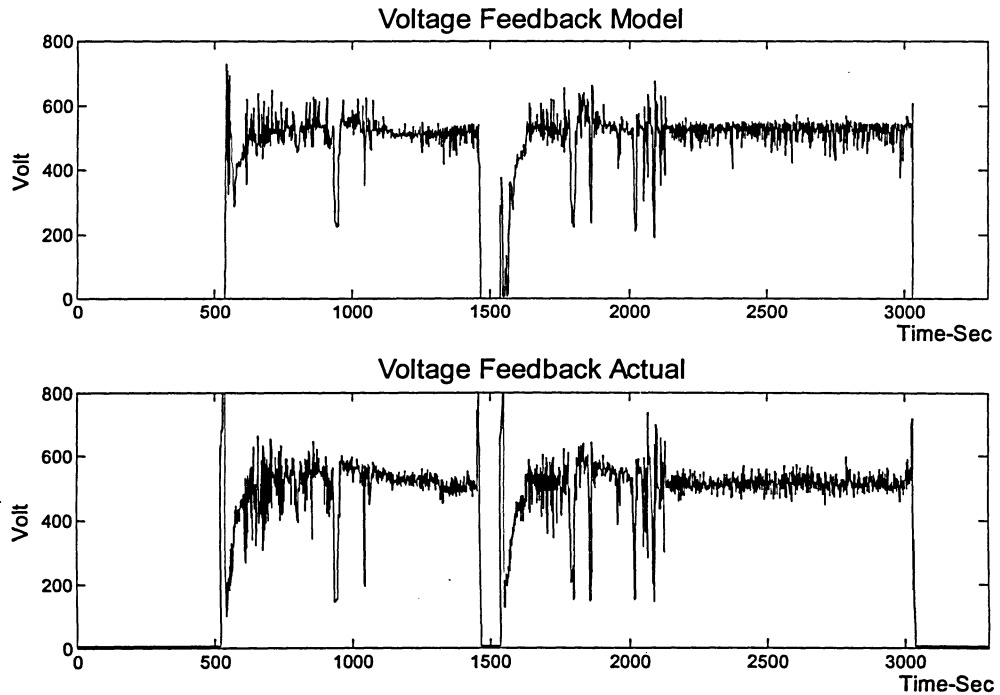


Figure 36: Phase voltage, model output and plant measured value graphs.

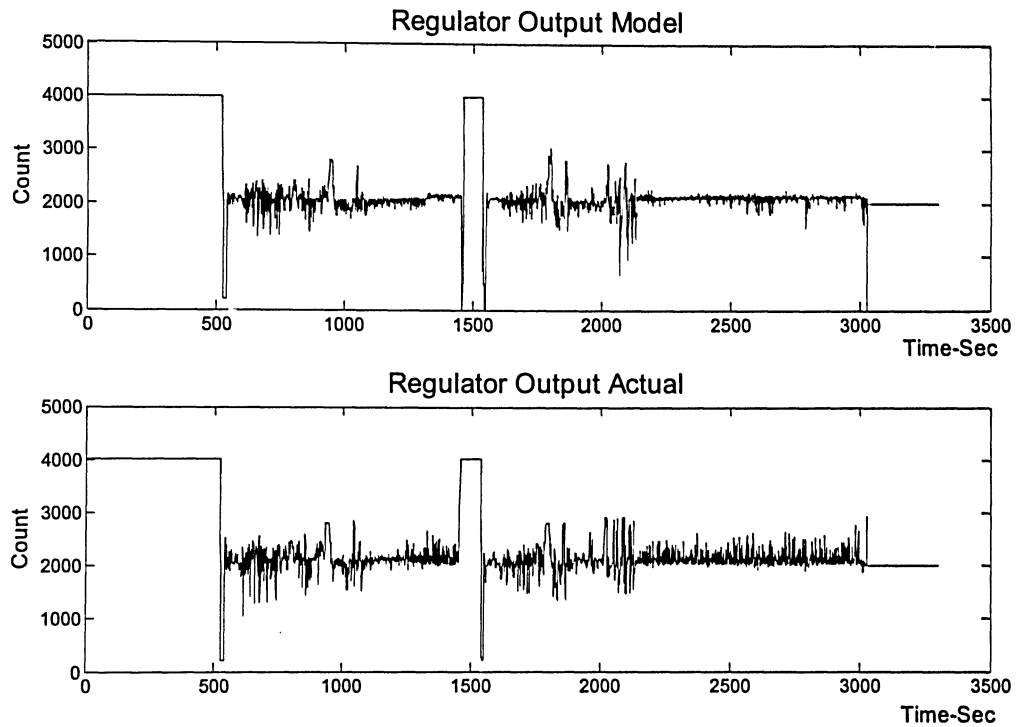


Figure 37: Regulator output, model output and plant measured value graphs.

5.4 The Closed Loop Simulator Graphs

In previous sections, the results of verification for each simulator component with the use of the plant's recorded data were illustrated. In this section, verification results for the entire simulator are presented. In this procedure, the plant's data was disconnected and the event sequencer module was used to control the batch process. The meltdown percentage signal from the plant was utilized to adjust all of the timings and a number of the gains. Subsequently, the simulator has been completely disconnected from the plant data and run for the desired time. The result illustrates the components of the simulator functioning almost perfectly under control of the sequential event simulator. The final result has been repeated as the simulator clock runs to its stop.

Fig. 38 illustrates the results of the simulation for a period of two hours. In this figure, Fig. 38(a) shows the internal state number for the simulator. As we could observe, the state number transforms from zero, time between the Heats, to one where Electrodes are forced down for a very short time (10 sec). It then proceeds to state two when the regulator is enabled for the first Charge. Then state three represents the time between Charges. Finally, it arrives back at state two once again to melt the second Charge. In Fig. 38(b) demonstrates the Charge number altering between 1 and 2 in every Heat. Fig. 38(c) in this figure shows the voltage reference generated from the lookup table and its profile as the melting process completes various stages. Fig. 38(d) illustrates the total real electrical power consumption based on the simulator calculation. This value is used as the pointer for all of the simulator lookup tables. Again, it must be emphasized that all of the lookup table's values, initial values, and timings in the simulator have been adjusted based on the real plant's data.

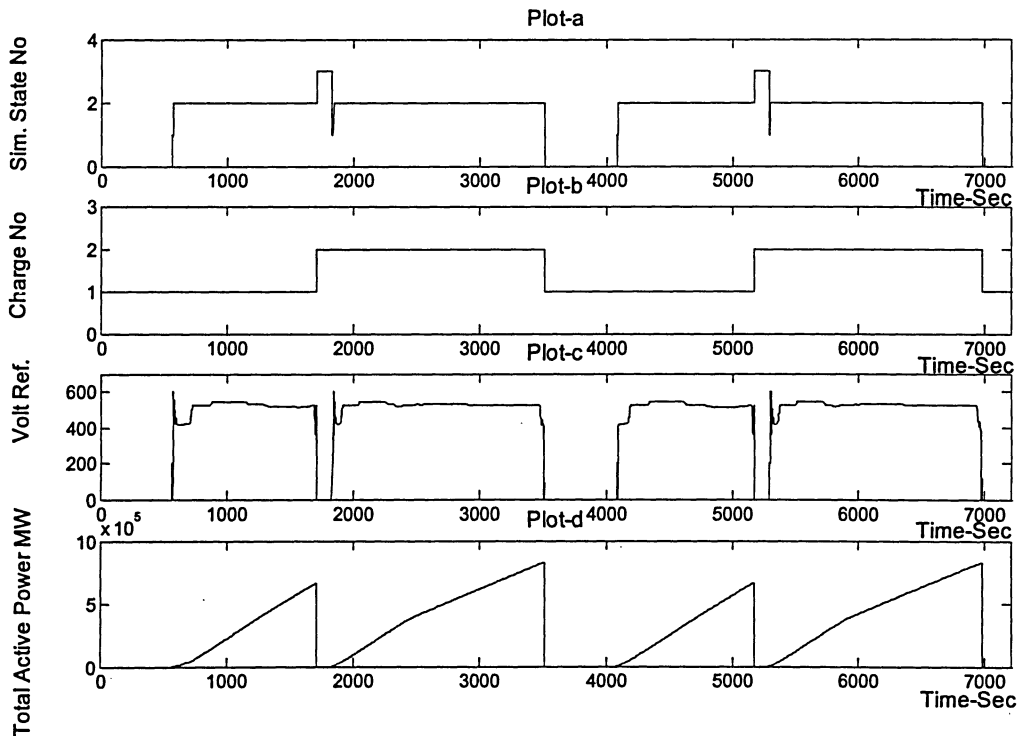


Figure 38: The closed loop simulator results without using the plant data.

5.5 Replacing the Classic Adaptive PI Controller with a Fuzzy Gain Scheduling PID

After achieving the ANFIS models for individual components of EAF regulator and also the full simulator of the regulator, logically the next step is to try the simulator platform in control improvements of the EAF. The process of control improvement and energy saving in EAF of GAW needs to be done in joint effort between the research team and the process engineers of the plant. Here to just prove the effectiveness of the simulator in control development effort, an attempt was made to replace classic adaptive PI controller of the regulator loop (explained in section 1.2) with fuzzy Gain Scheduling (GS) PID controller (explained in section 2.2).

In order to adapt the fuzzy GS PID explained in sec 2.2 with regulator simulator, first the range of the signals of the error and error changes were evaluated. In the case of the PID controller in the regulator loop, the error signal ($e(k)$) is the difference between the voltage reference and the voltage feedback for the phase. Also the changes in error signal ($\Delta e(k)$) is the difference of the error in the current sample and pervious sample ($e(k) - e(k-1)$). After finding the range of changes in signals $e(k)$ and $\Delta e(k)$, the triangle membership functions of those signals was scaled accordingly (Fig. 7). In addition the coefficients of proportional, integral and derivatives were normalized between zero to one. With keeping the rules as they are listed in Table 3, the regulator outputs from classic adaptive PI and fuzzy GS PID controllers were compared. Fig. 39 shows this comparison for phase 1B. Fig. 39(a) shows the regulator output using classic adaptive PI and Fig. 39(b) shows the regulator output utilizing the fuzzy GS PID. Also Fig. 39(c) is showing the voltage reference profile to appreciate the smoothness of the fuzzy GS PID in step changes of reference. As it was expected, the fuzzy GS PID is achieving the steady state level with much less overshoot and undershoot levels and with faster raising time. The smoothness of the electrode movement signals will also reduce the electrode consumption and lower the possibilities of the electrode break downs.

In another attempt to investigate the use of fuzzy GS PID in conserving energy, the regulator with fuzzy GS PID controllers were adapted into the full simulator for all three phases. With keeping the same thresholds to start and finish the Heats, the time of melting compared between the regulator with classic adaptive PI and fuzzy GS PID. Fig. 40 shows the total of 38 sec (or approximately 2% saving) in the melting time. This saving is achieved even without extra effort of fine tuning the fuzzy GS PID. This phase of the research was done to prove the usefulness of the simulator in designing and implementing modifications in supervisory control systems of the EAF.

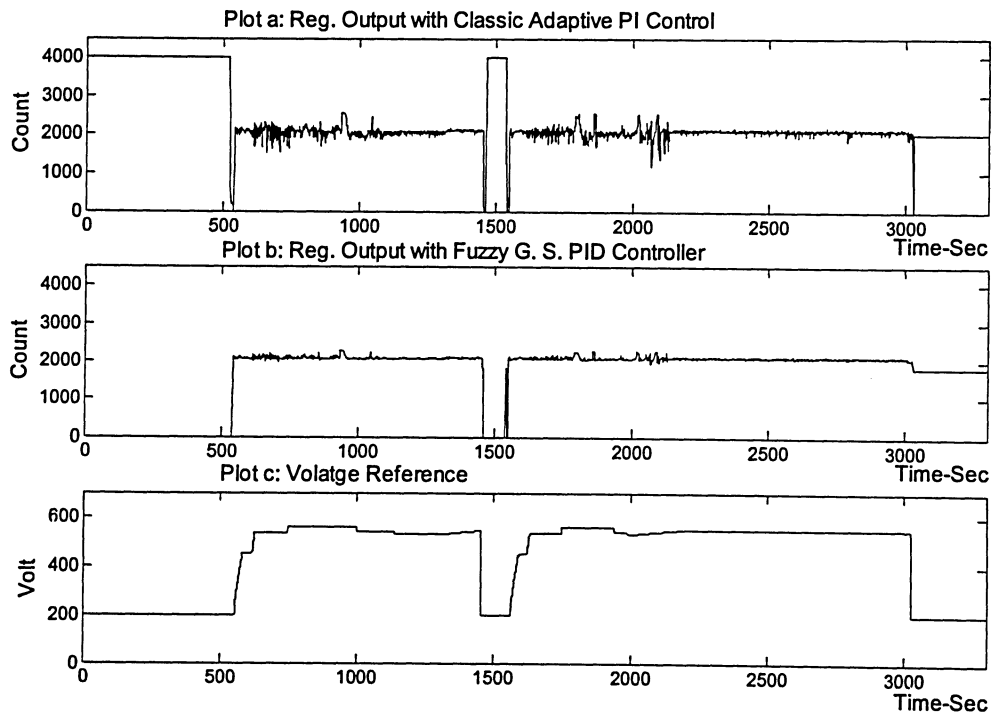


Figure 39: Comparing the regulator output with classic PI and GS Fuzzy PID controller.

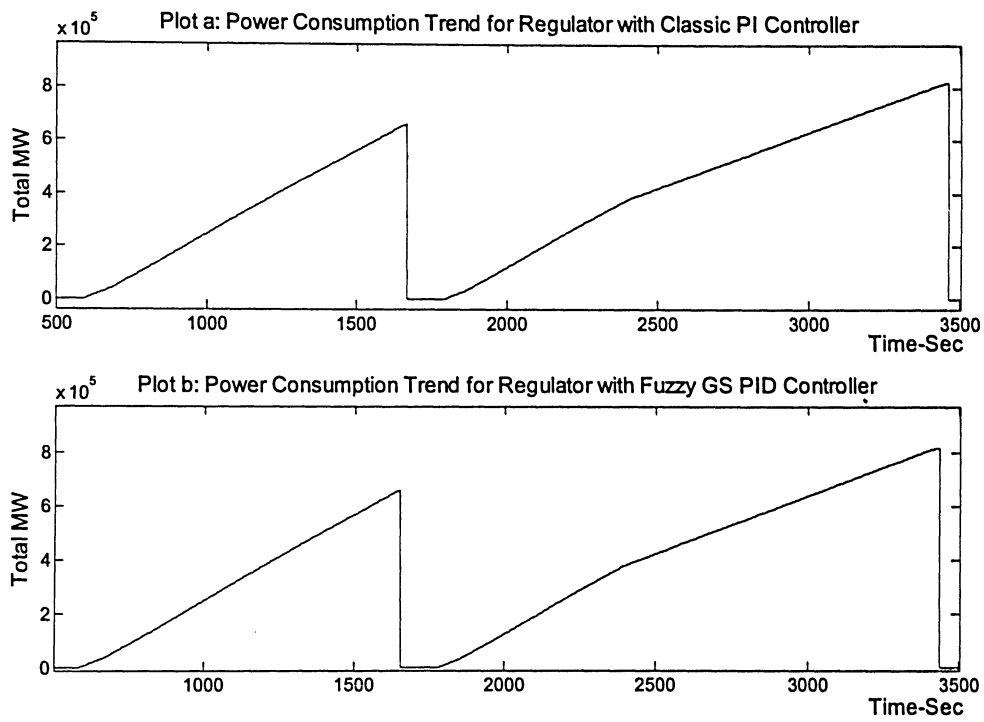


Figure 40: Comparing total melting time between regulator uses a classic PI and GS Fuzzy PID controller.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This research demonstrates the capability of ANFIS as a black box modeling method for the electrical measured signal in the regulator loop of EAF with a new set of inputs and outputs. Considering the nonlinearity and complexity of the process, the black box modeling, by use of ANFIS, assists in modeling a system which is mathematically ill defined. The integrated simulator, along with the developed sequential event generator and conventional control modules, indicate a performance, close to the actual melting batch process.

By concluding this research, a full closed loop computer simulator of the EAF regulator was developed in MATLAB® Simulink® environment. The simulator for the EAF regulator control is a closed loop system, which functions independently of the plant data and does not require manual operation. Also, it was demonstrated in the simulator that fuzzy gain scheduling PID control could perform better than conventional gain scheduling PI controller in the regulator loop. Although the data for this study was collected from GAW steel plant, the majority of the research concepts and techniques can be generalized or adapted to other electrical arc furnaces or similar systems.

6.2 Future Work

The objective of this study was to provide a platform in attempting various control systems improvement in the off-line simulator. At this time, by successfully building this platform, the immediate potential improvement would be to undertake intelligent control in generating the adaptive voltage reference and PI coefficients for the regulator control. The intelligent controller in format of Fuzzy, Neural or Nero-Fuzzy could essentially replace the lookup table and result in a more robust and adaptive system. Furthermore, since the source of the lookup tables in the existing system is the expertise of the process engineer, intelligent methods such as fuzzy controllers should be a rather suitable solution in expanding this expertise in an automated decision making system.

Moving ahead, the addition of the simulation of alternative energy generation from the burners and oxidization is a further potential direction for this research to consider. The alternative energy in the EAF is highly nonlinear as well and ANFIS could be undertaken to model their behaviors. By including these components, we could obtain a full energy consumption model or simulator of the EAF. The simulator may then be used to optimize energy consumption with electrical and chemical forms. Since the electrical and chemical energy interact with each other, building this full simulator will be quite constructive in capturing the optimum reference points for electrical and combustion systems in the duration of the melting process.

Bibliography

- [1] S. Varadan, E. B. Makram, and A.A. Girgis, "A new time domain voltage source model for an arc furnace using EMTP," *Proceedings IEEE Transactions on Power Delivery*, Vol. 11, no.3, pp. 1685-1691, July 1996.
- [2] R. C.Bellido and T. Gomez, "Identification and modeling of a three phase arc furnace for voltage disturbance simulation," *IEEE Transactions on Power Delivery*, Vol.12, no.4, pp. 1812-1817, October 1997.
- [3] T. Zheng, E.B. Makram, and A. A. Girgis, "Effect of different arc furnace models on voltage distortion," *Proceedings IEEE/PES and NTUA 8th Int. Conference Harmonics and Quality of Power*, Vol.2, pp.1079-1085, Athens, October 1998.
- [4] A.T. Tzanev, A.W. Ordys, and M.V. Tzaneva, "Modeling and simulation of EAF meltshop," *Proceedings American Control Conference*, Chicago, Illinois, pp. 2048-2052, June 2000.
- [5] T. Zheng and E.B. Makram, "An adaptive arc furnace model," *IEEE Transactions on Power Delivery*, Vol.15, no.3 pp. 931-939, July 2000.
- [6] P.E. King and M.D. Nyman, "Modeling and control of an electric arc furnace using feedforward artificial network," *Journal of Applied. Physics*, pp. 1872-1877, Aug. 1996.
- [7] A.R. Sadeghian and J.D. Lavers, "Application of radial basis function networks to model electric arc furnace," *Proceedings of IJCNN International Joint Conference Neural Networks*, Vol.6, pp3996-4001, Washington DC, 1999.
- [8] A.R. Sadeghian and J.D. Lavers, "Application of adaptive fuzzy logic systems to model electric arc furnace," *Proceedings of IEEE Fuzzy Information Processing Society NAFIPS, 18th International Conference of North America*, pp. 854-858, New York, NY. July 1999.
- [9] A.R. Sadeghian and J.D. Lavers, "Nonlinear black-box modeling of electrical arc furnace: an application of fuzzy logic systems," *Proceedings of IEEE International Fuzzy Systems Conference*, pp.234-239, Korea, Aug. 1999.
- [10] A.R. Sadeghian and J.D. Lavers, "Recurrent neuro-fuzzy predictors for multi-step prediction of v-I characteristics of electric arc furnace," *Proceedings of IEEE International Conference on Fuzzy Systems*, Vol.1, pp.110-115, San Antonio, TX. 2000.

- [11] D.Raisz, M. Sakulin, H. Renner, and Y. Tehlivents, "Recognition of the operational states in Electric Arc Furnace," *IEEE Proceedings International Conference on Harmonics and Quality of Power*, Vol. 2, pp. 475-480, Orlando, FL. 2000
- [12] A.R. Sadeghian and J.D. Lavers, "On the use of recurrent neuro-fuzzy networks for predictive control," *Proceedings of IEEE IFSA World Congress and 20th NAFIPS International Conference*, Vol.1 pp. 233-238, Vancouver, BC. 2001.
- [13] W.E. Staib and R.B. Staib, "The intelligent arc furnace controller: a neural network electrode position optimization system for the electric arc furnace," *Proceedings International Joint Conference Neural Networks: IJCNN*, Vol. 3, pp. 1-9, Baltimore, MD. June 1992.
- [14] Q.P. Wang, D.L. Tarn, and Y.C. Wang, "Event-based intelligent control systems of carbide electric arc furnace (CEAF)," *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, pp. 471-476, P.R. China, 2000.
- [15] E.L. Wilson, C.L. Karr, and J.P. Bennett, "An adaptive, intelligent control system for slag foaming," *Applied Intelligent*, pp. 165-177, 2004.
- [16] P.Wang and D.P. Kwok, "Optimal fuzzy PID control based on genetic algorithm," *IEEE International Conference Power Electronics and Motion Control.*, Vol. 2 pp. 977-981, San Diego, CA. 1992.
- [17] Z.Y. Zhao, M. Tomizuka, and S.Isaka, "Fuzzy gain scheduling of PID controllers," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, pp. 1392-1398, 1993.
- [18] S.J. Qin, "Auto-tuned fuzzy logic control," *Proceedings of the American Control Conference*, Vol. 3, pp. 2485-2489, Baltimore, MD. June 1994.
- [19] G.K.I. Mann, B.G. Hu, and R.G. Gosine, "Fuzzy PID controller structures," *IEEE Electrical and Computer Engineering Canadian Conference*, Vol. 2, pp. 788-791, St. Johns, Nfld., 1997.
- [20] G.K. Mann, B.G. Hu, and R.G. Gosine, "Analysis of direct action fuzzy PID controller structures," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 29, no 3, June 1999
- [21] J. Lu, R.S. Ranganathan, G. Chen, and H.A. Maiki, "Predictive fuzzy PID control for complex processes," *IEEE fuzzy systems Interational. Conference*, Vol. 2, pp. 544-548, San Antonio, TX. May 2000.

- [22] Y. Huang and S. Yasunobu, "A general practical design method for fuzzy PID control from conventional PID control," *IEEE International Fuzzy Systems Conference*, Vol. 2, pp. 969-972, San Antonio, TX. May 2000.
- [23] B.G. Hu, G.K.I Mann, and R.G. Gosine, "A systematic study of fuzzy PID controllers – function based evaluation approach," *IEEE Transactions on Fuzzy Systems*, Vol. 9, pp.699-712, Oct. 2001.
- [24] K.S. Tang, K.F. Man, G. Chen, and S. Kwong, "An optimal fuzzy PID controller," *IEEE Transactions on Industrial Electronics*, Vol. 48, pp. 757-765, Aug. 2001.
- [25] H.B. Kazemian, "Development of an intelligent fuzzy controller," *IEEE International Fuzzy systems Conference* Vol. 1, pp. 517-520, Melbourne, Vic. 2001.
- [26] Y. Yang, W.G. Wang, D.J. Yu, and G. Ding, "A fuzzy parameters adaptive PID controller design of digital positional servo system," *Proceedings of International Conference on Machine Learning and Cybernetics*, Vol. 1, pp. 310-314, Beijing, China Nov. 2002.
- [27] M. Petrov, I. Ganchev, and A. Taneva, "Fuzzy PID control of nonlinear plants," *IEEE Symposium on Intelligent Systems*, pp. 30-35, Vancouver, B.C., Sep. 2002.
- [28] K.L. Lo and M.O. Sadegh, "Systematic method for the design of a full-scale fuzzy PID controller for SVC to control power system stability," *IEE Proceedings Generation, Transmission and Distribution* Vol. 150, no 3, PP. 297-304, May 2003.
- [29] J. Xu and X. Feng, "Design of adaptive fuzzy PID tuner using optimization method," *IEEE Proceedings Intelligent Control and Automation*, pp. 2454-2458, Hangzhou, China, June 2004.
- [30] Y. Yongquan, H. Ying, and Z. Bi, "The dynamic fuzzy method to tune the weight factors of neural fuzzy PID control," *IEEE International Joint Conference on Neural Networks*, Vol 3. pp. 2397-2402, Budapest, Hungary, July 2004.
- [31] L.A. Zadeh, *Fuzzy Sets Information and Control*, pp. 338-353, 1965.
- [32] E.H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with fuzzy logic controller," *International Journal Man-Machine Studies*, Vol. 7, pp. 1-13, 1975.
- [33] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Transactions on Systems Man and Cybernetics*, Vol.15, pp. 116-132, 1985.

- [34] M. Sugeno and G.T. Kang, "Structure identification of fuzzy model," *Fuzzy sets and systems*, Vol. 28, pp. 15-33, 1988.
- [35] J. Shing and R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665-685, 1993.
- [36] J.S.R. Jang , C.T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, NJ, 1997.
- [37] Roger Jang, "Ph.D. dissertation", Department of electrical engineering and computer science, University of California at Berkeley, July 1992.

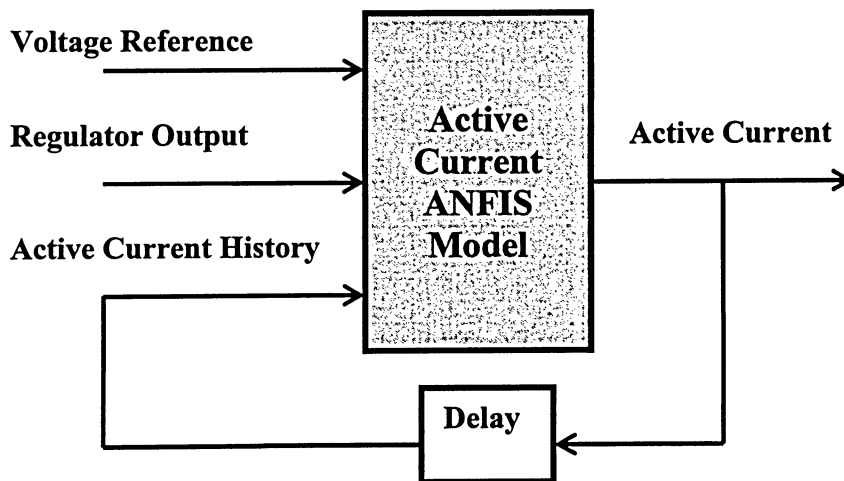
Publications

F.Janabi-Sharifi, G. Jorjani, I. Hassanzadeh, "Using adaptive neuro fuzzy inference systems in developing an electrical arc furnace simulator," *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics, 2005*, pp. 1210-1215, CA, 2005.

Appendix A:

ANFIS models inputs, outputs and parameters

a- Active current ANFIS model



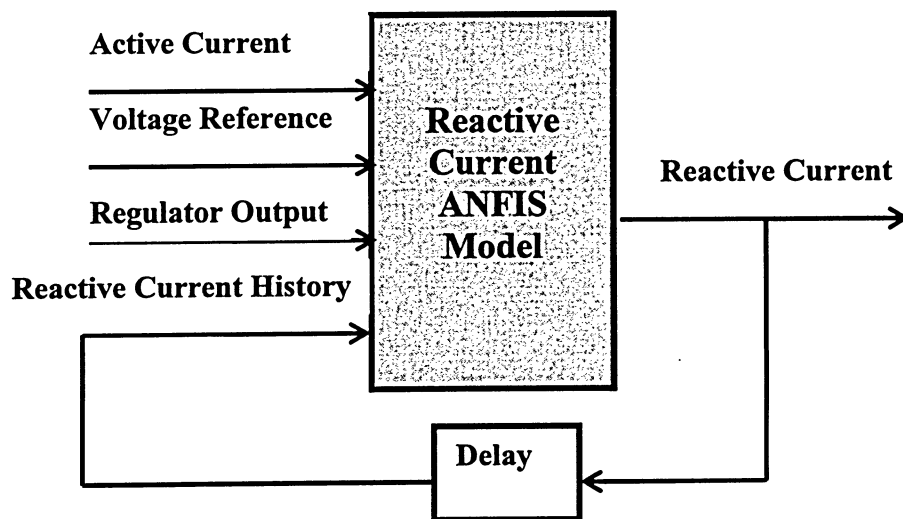
The format of the listed FIS parameters are the output of MATLAB ® ‘showfis ()’ function:

Name	Ph1BcurrentFdbk
Type	sugeno
Inputs/Outputs	[3 1]
NumInputMFs	[3 3 3]
NumOutputMFs	3
NumRules	3
AndMethod	prod
OrMethod	probor
ImpMethod	min
AggMethod	max
DefuzzMethod	wtaver

InLabels	in1 in2 in3
OutLabels	out1
InRange	[200 560] [854 4054] [0 609]
OutRange	[0 609]
InMFLabels	in1mf1 in1mf2 in1mf3 in2mf1 in2mf2 in2mf3 in3mf1 in3mf2 in3mf3
OutMFLabels	out1mf1-outmf3
InMFTypes	gaussmf
OutMFTypes	linear
InMFParams	[63.63 540 0 0] [63.64 200 0 0] [63.64 200 0 0] [565.7 2048 0 0] [565.7 4048 0 0] [565.7 2039 0 0] [107.7 539 0 0] [107.7 0.00952 0 0] [107.6 -0.008692 0 0]

OutMFParams	[0.01088 0.006218 0.6664 164.5]
	[-10.75 0.02656 0.161 2043]
	[0.02368 -0.0183 1.009 32]
Rule Antecedent	[1 1 1]
	[2 2 2]
	[3 3 3]
Rule Consequent	1
	2
	3
Rule Weigth	1
	1
	1
Rule Connection	1
	1
	1

b- Reactive current ANFIS model

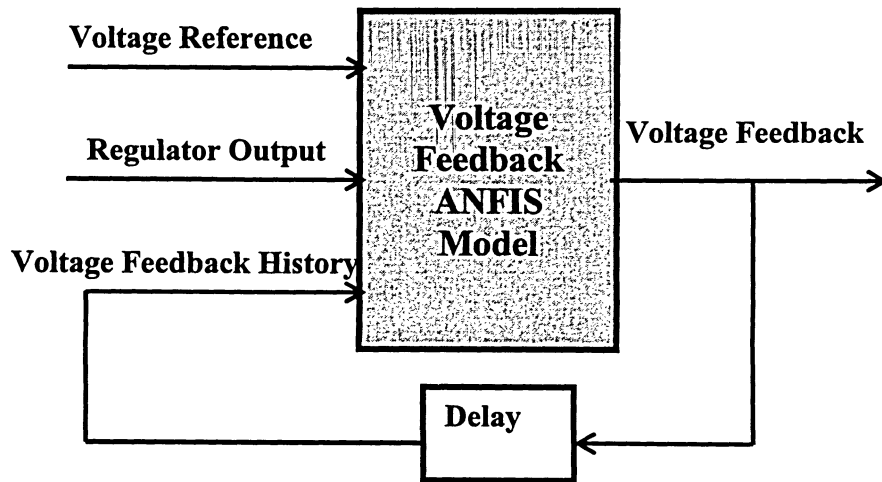


Name	Ph1B_Reactive_KA
Type	sugeno
Inputs/Outputs	[4 1]
NumInputMFs	[3 3 3 3]

NumOutputMFs	3
NumRules	3
AndMethod	prod
OrMethod	probor
ImpMethod	min
AggMethod	max
DefuzzMethod	wtaver
InLabels	in1
	in2
	in3
	in4
OutLabels	out1
InRange	[0 640]
	[200 560]
	[207 4007]
	[1 1104]
OutRang	[1 1104]
InMFLabels	in1mf1
	in1mf2
	in1mf3
	in2mf1
	in2mf2
	in2mf3
	in3mf1
	in3mf2
	in3mf3
	in4mf1
	in4mf2
	in4mf3
OutMFLabels	out1mf1-outmf3
InMFTypes	gaussmf
OutMFTypes	linear
InMFParams	[113.1 554 0 0]
	[113.1 3.898e-006 0 0]
	[113.1 -0.004694 0 0]
	[63.69 546 0 0]
	[63.64 200 0 0]
	[63.64 200 0 0]
	[671.8 2131 0 0]
	[671.8 3978 0 0]
	[671.8 1965 0 0]

	[195 135 0 0]
	[195 1.999 0 0]
	[195 1.998 0 0]
OutMFParams	[-0.3809 -0.401 0.1021 0.7954 240.4]
	[-0.01175 -63.94 -0.02555 0.8065 1.289e+004]
	[-0.05789 2.212 -0.03692 1.041 -368.9]
Rule Antecedent	[1 1 1 1]
	[2 2 2 2]
	[3 3 3 3]
Rule Consequent	1
	2
	3
Rule Weigth	1
	1
	1
Rule Connection	1
	1
	1

c- Voltage feedback ANFIS model



Name	Ph1BvoltFdbk
Type	sugeno
Inputs/Outputs	[3 1]
NumInputMFs	[3 3 3]
NumOutputMFs	27
NumRules	27
AndMethod	prod
OrMethod	probor
ImpMethod	min
AggMethod	max
DefuzzMethod	wtaver
InLabels	input1 input2 input3
OutLabels	output
InRange	[200 560] [207 4007]
OutRange	[4 1229] [4 1229]

InMFLabels	in1mf1 in1mf2 in1mf3 in2mf1 in2mf2 in2mf3 in3mf1 in3mf2 in3mf3
OutMFLabels	out1mf1 –out1mf27
InMFTypes	gaussmf
OutMFTypes	linear
InMFParams	[76.44 200 0 0] [76.44 380 0 0] [76.44 560 0 0] [806.9 207 0 0] [806.9 2107 0 0] [806.9 4007 0 0] [260.1 4 0 0] [260.1 616.5 0 0] [260.1 1229 0 0]

OutMFParams	[71.8 0.3743 -6.489 -1.913e+004] [32.19 1.421 -21.01 -4443] [-230.5 -38.01 284.6 38.27] [-2.621 -3.824 6.808 8483] [-15.94 0.6997 19.1 -5825] [-554.7 -21.99 219.2 1878] [64.79 1.704 -10.34 -3848] [94.03 -3.345 -16.25 -5589] [1361 -125.3 -287.3 125] [412.9 -103.1 461.9 -1.046e+004] [-671.9 1.247 474.7 1.014e+004] [4008 377.4 -4599 242.2] [-57.9 10.02 -36.26 3411] [65.72 0.6859 -70.28 5705] [1776 195.2 -1645 -846.9] [512.1 -91.89 25.48 1.002e+004] [-471 48.29 41.62 1.856e+004] [-1.446e+004 1991 3171 204.4] [-728.7 213.7 199.9 -1.141e+004] [52.82 -13.76 -7.474 1.81e+004] [-2419 -40.75 1777 3799] [49.08 3.64 7.175 -4.059e+004] [12.09 3.027 7.292 -1.809e+004] [246.4 -45.36 -28.71 -1.384e+004] [-61.93 -7.866 12.3 5.672e+004] [-202.8 -0.7476 18.63 1.063e+005] [1041 6.821 -1067 386.6]
Rule Antecedent	[1 1 1] [1 1 2] [1 1 3] [1 2 1] [1 2 2] [1 2 3] [1 3 1] [1 3 2] [1 3 3] [2 1 1] [2 1 2] [2 1 3] [2 2 1] [2 2 2] [2 2 3] [2 3 1]

[2 3 2]
 [2 3 3]
 [3 1 1]
 [3 1 2]
 [3 1 3]
 [3 2 1]
 [3 2 2]
 [3 2 3]
 [3 3 1]
 [3 3 2]
 [3 3 3]

Rule Consequent 1 ...27

Rule Weigth 1

Rule Connection 1