THE UNIVERSITY OF CALGARY

AN ADAPTIVE FUZZY LOGIC POWER SYSTEM STABILIZER

by

ALI HARIRI

A DISSERTATION
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING

CALGARY, ALBERTA
APRIL, 1997

0-612-20741-2

Abstract

In this dissertation, an adaptive fuzzy logic control algorithm has been developed for a Power System Stabilizer (PSS) to improve dynamic performance of the system. The proposed PSS deals with automating the parameter tuning and structure optimization in order to achieve the desired performance.

This approach combines the advantages of both Fuzzy Logic Control (FLC) and Artificial Neural Network (ANN) and avoids their drawbacks. The parameters of the controller, membership functions and inference rules are adjusted according to gradient decent learning algorithm.

Moreover, the mechanism of how the FLC can be trained in a closed-loop control system is investigated. In the first step, a desired controller is employed to generate the input-output data required for training. The FLC learns to copy the desired controller. This approach needs the existence of the desired controller. To overcome this problem, in the next step, a self-learning approach is utilized to train the FLC directly from the plant output. A genetic algorithm is also used to optimize the structure of FLC, preventing the learning algorithm from the overfitting problem.

Simulation studies and comparison between the proposed adaptive fuzzy PSS and the conventional PSS using a single-machine connected to an infinite bus are conducted. For verification, it has been applied to a multi-machine model of the power system.

A TMS320C30 Digital Signal Processor (DSP) and an ABB PHSC2 Programmable Logic Controller (PLC) were employed to develop a prototype real-time digital control environment and to implement adaptive fuzzy logic PSS.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. O. P. Malik, for his constant guidance, encouragement and support throughout the whole program. From him, I have learned so much not only in the area of my specialty but also in the style of teaching and doing research.

I wish to thank Dr. G. S. Hope and Dr. T. Chen for their useful discussions and productive suggestions during my program. Also, I would like to thank all professors and support staff, especially Mr. G. C. Hancock in the Department of Electrical and Computer Engineering, The University of Calgary, for their help during my study here.

I also wish to acknowledge the financial support from the Ministry of Culture and Higher Education, Tehran, I. R. Iran.

Special appreciation goes to Dr. K. A. El-Metwally, Dr. G. P. Chen., Dr. S. Chen, Mr. M. Sanaye Pasand, all other fellow students and my friends whose names are impossible to be listed here, for their valuable help from time to time.

I am indebted to my wife, who is always patient and encouraging. Without her constant spiritual support, I could not have finished this dissertation. My special thanks go to our families for their help and encouragement to me and my wife when we encountered difficulties.

Dedication

To

My Family

Table of Contents

List of Tables

List of Figures

List of Symbols

Control Theory

ADALINE	ADAptive LINear Element
A_i, B_i	Linguistic Variables
ART	Adaptive Resonance Theory
BP	Back Propagation
C	Output of Fuzzy Set
E_{chki}	network error obtained from checking data
E_p	error measure for the pth entry of training data
E_{tri}	network error obtained from training data
FAM	Fuzzy Associative Memory
F_i	fitness function
FLC	Fuzzy Logic Control
$J(w)$	cost function
LMS	Least Mean Square
MF	Membership Function
MIMO	Multi Input Multi Output
MLP	Multi-Layer Perceptron
M_x	Semantic Function
N_i	number of adaptive nodes
NL	Negative Large
NM	Negative Medium
NS	Negative Small
PA	Pole Assignment
PID	Proportional Integral Derivative controller
PL	Positive Large
PM	Positive Medium
PS	Positive Small
PS	Pole Shifting
RBF	Radial Basis Function
R_i	ith inference rule
S	a set of nodes
SGN	sign operator
T_s	sampling period
$T(x)$	set of inference rules
U	Universe of discourse
ZO	Zero
$e, \dot{e}$	system error, system derivation of error

exp	exponential function
$k_1,\ k_2,\ k_3$	weighting parameters in objective function
min	minimum function
μ_i	membership grade
u	control signal
u_d	desired control signal
u_i	linear output
$y_{i,p}^k$	node output in the ith position of kth layer
y_p	pth component of actual output vector
$\hat{y_p}$	pth component of desired output vector
w_{ij}	synaptic weight
x_i	input signal
y_i	output signal
z	plant output
z^{-1}	backward discrete time shift
z_d	desired plant output
Φ_i	objective function
Φ_{max}	maximum of objective function
α	parameter of adaptive network
α_i	firing strength of ith inference rule
β	momentum factor
η	learning rate
θ	threshold
λ	weighting parameter of control signal
τ	sampling time
ϕ	activation function

Power System

$A_1,\ A_2$	conventional power system stabilizer filter constants
AC	Alternating Current
AI	Artificial Intelligence
APSS	Adaptive PSS
ANF PSS	Adaptive-Network-based Fuzzy PSS
ANN	Artificial Neural Network
AVR	Automatic Voltage Regulator
B	transmission line suceptance
CPSS	Conventional Power System Stabilizer
DC	Direct Current
GA	Genetic Algorithm

H	generator inertia value
L	Load in admittance
P, Q	generator active and reactive power
PSS	Power System Stabilizer
PT	Potential Transformer
K_A, K_C, K_F	AVR gains
K_{LF}, I_{LF}	AVR gains
K_d	generator damping ratio coefficient
K_s, K_δ	conventional power system stabilizer gain
P_e	electric active power
P_{acc}	acceleration power
R_C, X_C	voltage transducer compensation constants
$T_1, \cdots, T_5$	conventional power system stabilizer time constants
T_A, T_R, T_F	AVR time constants
T_B, T_C	AVR time constants
T_{B1}, T_{C1}	AVR time constants
TCR	Time Constant Regulator
T'_{do}, T''_{do}	generator d-axis transient and sub-transient time constants
T''_{qo}	generator q-axis sub-transient time constant
T_Q	conventional PSS washout filter time constant
T_e	generator electrical torque output
T_g	governor time constant
T_m	generator mechanical torque input
U_{pss}, $Vpss$	power system stabilizer output
V_b	infinite-bus voltage reference
V_{OEL}, V_{UEL}	AVR over-excitation and under-excitation limit
V_t, V_{ref}	generator terminal voltage and AVR voltage reference setting
V_{XMAX}, V_{XMIN}	AVR V_X voltage variable upper and lower limit
a, b	governor gain constants
e_d, e_q, e_f	generator d-axis, q-axis and field winding voltage
e''_{do}	generator d-axis sub-transient voltage
e'_{qo}, e''_{qo}	generator q-axis transient and sub-transient voltages
$\{f'_i\}, \{g'_i\}$	digital conventional PSS coefficient
g	governor output
i_d, i_q, i_f	generator d-axis, q-axis and field winding current
i_{kd}, i_{kq}	generator d-axis, q-axis damper winding current
$p.f.$	power factor
$p.u.$	per-unit
r_a	generator armature resistance
r_f, r_{kd}, r_{kq}	generator field, d-axis and q-axis damper windings resistance

r_t	transmission line resistance
u, u_{pss}	power system stabilizer output
x_d, x'_d, x''_d	generator d-axis reactances, transient and sub-transient reactances
x_q, x''_q	generator q-axis reactances and sub-transient reactances
x_{kd}, x_{kq}	generator d-axis and q-axis damper winding reactances
x_{md}, x_{mq}	generator d-axis and q-axis mutual reactances
x_t	transmission line reactance
ΔP	generator power deviation
$\Delta\omega$	generator speed deviation
$\Delta\dot{\omega}$	generator acceleration deviation
δ	generator power angle
ω, ω_0	generator speed and its rated value
λ_d, λ_q, λ_f	generator d-axis, q-axis and field flux linkage
λ_{kd}, λ_{kq}	generator d-axis, q-axis damper winding flux linkage

Computer Technology

ABB	Asea Brown Boveri company
A/D	Analog to Digital conversion
D/A	Digital to Analog conversion
DAS	Data Acquisition System
DMA	Direct Memory Access
DSP	Digital Signal Processor
IC	Integrated Circuit
I/O	Input/Output
PC	Personal Computer
PHSC2	Programmable High Speed Controller
PLC	Programmable Logic Controller
TMS320C30	Texas Instruments DSP board
VLSI	Very Large Scale Integrated

Chapter 1

Introduction

1.1 Power System Control

An electric power system contains thousands of interconnected electric elements. Many elements are highly nonlinear and some of them are combinations of electrical and mechanical parts. Power systems have thus developed into complex operating and control systems with various kinds of unstable characteristic [1][2]. Since these systems are spread over vast geographical areas, some of which span over the entire continents, they are subject to many different types of disturbances. Also, the tendency of operating the generators with small stability margins has made these systems even more fragile [3][4].

With the advent of interconnection of large electric power systems, many new problems have emerged [1]. Some of these problems are the oscillations of the sub-systems of a large interconnected power system against each other, the sub-synchronous tortional oscillations of turbines in a steam power plant with capacitor-compensated transmission lines, and others [5].

The definition of stability, as applied to power systems may be stated as [3]:

> If the oscillatory response of a power system during the transient period following a disturbance is damped and the system settles in a finite time to a new steady state operating condition, the system is stable. Otherwise, it is considered unstable.

In order to simplify the analysis, power system stability is considered in its three aspects, namely [6][7][8]:

Steady State Stability – This refers to the stability of a power system subjected to small and gradual changes in load. If the synchronous machine maintains synchronism after such a small disturbance, it is said to be steady-state stable.

Dynamic Stability – This refers to the stability of a power system subjected to a relatively small and sudden disturbance. For this category, it is assumed that the system is steady-state stable and small variations around a certain steady-state point are studied.

Transient Stability – If a synchronous machine maintains equilibrium when subjected to a sudden impact, then it has transient stability. Large changes in load, line switching and system faults can be considered to be impact disturbances which may lead to transient instability.

A small signal perturbation model around an equilibrium point can be considered for dynamic stability studies and the system can be described by linear differential equations. However, for transient stability analysis and control design, the power system must be described by nonlinear differential equations.

Although there are several sources of positive damping in a power system, there are also sources of negative damping, notably voltage-regulating and speed- governing systems. Furthermore, although ordinarily the inherent positive damping predominates, in some circumstances the net damping can become negative. With net negative damping, angular swing of the machine, instead of declining, increases either until equilibrium amplitude is reached or synchronism is lost.

Over the years, considerable efforts have been devoted to improve power system stability in various ways [9][10][11][12]. These attempts can be divided into three broad groups as below:

- generator excitation control,
- generator input power control, and
- system operating condition and configuration control.

For a particular problem, any one or more of the above methods can be employed. Among these methods, excitation control is preferred due to the following reasons:

- generally electrical systems have much smaller time constants than mechanical systems,
- electrical control systems are more economical and easy to implement than mechanical control systems,
- additional equipment required operates at low power level, whereas other methods (such as resistor braking and capacitor switching) need a much higher power level.

Effectiveness of damping produced by excitation control has been demonstrated both by computation and by field tests [13][14]. To date, many of the major electric power plants in large interconnected systems are equipped with this supplementary excitation control, commonly referred to as Power System Stabilizer (PSS). Several kinds of supplementary signals (speed deviation, frequency deviation and accelerating power) have been used as input signals to the PSSs.

1.2 Supplementary Excitation control

Excitation controllers have been used widely in power systems for decades. The main object is to achieve an acceptable voltage profile at the consumer terminal and to control the reactive power flow in the network. High gain, short time constant and high ceiling voltage excitation control are among the characteristics of this control loop. These result in increasing both the steady state and transient stability limits of the system [15].

As the size of the interconnected power system grew, the possibility of withstanding unexpected disturbances without loss of system stability increased. It became apparent that the voltage control loop had a detrimental impact upon the dynamic stability of the power system. Oscillations of small magnitude and low frequency often persisted for long periods of time and in some cases presented limitations on power transfer capability. Similar types of oscillations might also be observed when remote generating units are connected to a relatively large power system through long radial transmission lines.

Various methods have been proposed to enhance the dynamic performance of the power system. They can be divided into two broad groups:

- Design new excitation controller based on modern control theory,
- Improve the performance of the presently used excitation controllers by introducing a supplementary control signal [16].

A typical method in the second group is to utilize a PSS [17][18]. The basic function of a PSS is to extend stability margin via modulation of the generator excitation to damp the oscillations of synchronous machine. The oscillations of concern occur in the frequency range of approximately 0.2 to 2.5 Hz. To provide damping, the stabilizer produces a component of electrical torque on the rotor which is in phase with speed variations. Independent of the type of input signal, the stabilizer must compensate for the gain and phase characteristics of the excitation system, the generator, and the power system, which collectively determine the open loop transfer function. This transfer function is strongly influenced by voltage regulator gain, generator power level, and AC system strength.

1.3 Different Types of Stabilizers

1.3.1 Conventional Power System Stabilizer

Today, PSSs are widely used on synchronous generators. The most commonly used PSS, referred to as the Conventional PSS (CPSS), is a fixed parameter analog -type device. The CPSS, first proposed in 1950's, is based on the use of a transfer function designed using the classical control theory [19]. It contains a phase compensation network for the phase difference from the excitation controller input to the damping torque output. By appropriately tuning the phase and gain characteristics of the compensation network, it is possible to set the desired damping ratio. CPSSs are widely used in the power systems these days and have improved power system dynamic stability.

The CPSS, however, has its inherent drawbacks. It is designed for a particular operating condition around which a linearized transfer function model is obtained. The high non-linearity, very wide operating conditions and unpredictability of perturbations of the power system exhibit the following problems to the CPSS:

- the accuracy of linear model for the power system,
- the accuracy of the parameters for that model,
- the effective tuning of the CPSS parameters,
- the interaction between the various machines,
- the tracking of the system non-linearity.

Extensive research has been carried out to solve these problems [20]. Numerous tuning techniques have been introduced to effectively tune the CPSS parameters [21]. Mutual interaction between CPSSs in multi-machine systems has also been studied [22]. To solve the parameter tracking problem, variable structure control theory was introduced to design the CPSS [23]. However, the CPSS is a linear controller which generally cannot maintain the same quality of performance at other operating conditions.

1.3.2 Adaptive Power System Stabilizer

The adaptive control theory provides a possible way to solve the above mentioned problems relating to the CPSS [24]. At each sampling instance, input and output of the generating unit are sampled, and a mathematical model is obtained by some on-line identification method to represent the dynamic behavior of the generating unit at that instant of time. It is expected that the mathematical model obtained at each sampling period can track changes in the system.

Following the identification of the model, the required control signal for the generating unit is produced based on the identified model. There are various control strategies, among them are Pole Assignment (PA) and Pole Shifting (PS) techniques [25]. These control strategies are generally developed by assuming that the identified model is the true mathematical description of the generating unit [26][27][28][29]. However, since the power system is a high order nonlinear continuous system, it is hard for the low order discrete identified model to precisely describe the dynamic behavior of the power system. Consequently, a high order discrete model is used to represent the power system, which consumes a significant amount of computing time. The computing time for an adaptive PSS is roughly proportional to the square of the order of the discrete model used in the identification. The longer computing time limits the control effect. This is more significant if the oscillation frequency is relatively high. There must be a compromise between the order of the discrete model and the computing time for parameter identification and optimization.

1.3.3 Neural Network Based PSS

Artificial neural networks (ANNs) attempt to achieve good performance via dense interconnection of simple computational elements [30]. Their structure is based on the present understanding of biological nervous systems.

ANNs have a number of advantages [31]:

- Capability of synthesizing complex and transparent mappings.
- Speed due to the parallel mechanism.
- Robustness and fault tolerance.

- Adaptively adjustable to the new environment.

Research on ANN application in power system stability has been reported in [32][33][34]. The success of ANNs to control unknown systems under significant uncertainties makes ANNs very attractive. However, there are some drawbacks to the using of conventional ANNs as follows:

- Black-box characteristics; it is difficult for an outside observer to understand or modify the network decision making process; the reason that initial values for the parameters are chosen randomly.

- Long training time; ANNs may require a long training time to get the desired performance. The larger the size of ANN and the more complicated the mapping to be performed, the longer the training time required.

1.3.4 Fuzzy Logic Based PSS

One of the new methods which has recently been used in many controller designs is Fuzzy Logic Control (FLC) [35]. Fuzzy control systems are rule-based systems in which a set of fuzzy rules represents a control decision mechanism to adjust the effects of certain causes coming from the system [36][37].

The followings are some of the major features of FLC [38][39]:

- Model free based; unlike other classical control techniques, this method doesn't require the exact mathematical model of the system.

- Robust nonlinear controller; FLC offers ways to implement simple but robust solutions that cover a wide range of system parameters and that cope with major disturbances.

- Development time reduction; FLC works at two levels of abstraction: the symbol level and compiled level. The symbol level is appropriate for describing the application engineers' strategies, while the compiled level is well understood by the electronics engineers. Since there is a well-defined translation between those levels, an FLC can help in reducing the communication problems.
- Knowledge based; fuzzy control simulates the strategy of the person controlling a process. Thus, the control strategy mimics the human's way of thinking. In this way, the experience of a human operator can be implemented through an automatic control method, not through the slow response of a human controller.

Designing stabilizers based on FLC is a very active area and satisfactory results have been obtained [40][41]. Although FLC introduces a good tool to deal with complicated, nonlinear and ill-defined systems, it suffers from a drawback - the "parameter tuning" for the controller. At present, there is no systematic procedure for the design of the FLC. The most straight forward approach is to define Membership Functions (MFs) and decision rules subjectively by studying an operating system or an existing controller. Therefore, there is a need for an effective method for tuning the MFs and decision rules so as to minimize the output error or maximize the performance index.

1.4 Thesis Objective

The objective of the thesis is to solve the problems encountered with the design of fuzzy logic and neural network based power system stabilizer. This work makes original contribution to the development and application of the power system stabilizer.

To be more specific, the objective of this thesis includes the following aspects:

1. Both the FLC and ANN have been employed together to design a new PSS, Adaptive-Network-Based Fuzzy Logic PSS (ANF PSS). In this approach, a fuzzy logic PSS with learning ability has been constructed and is trained directly from the input and output data of the generating unit. Because the ANF PSS has the property of learning, MFs and fuzzy decision rules can be tuned automatically by the learning algorithm. Learning is based on the error that is evaluated by comparing the output of the controller with the desired controller which in this case has been chosen a self-optimizing pole-shifting adaptive PSS.

2. In a typical situation, the desired controller may not be available. Therefore, a self-learning approach is utilized to train the ANF PSS from the performance of the generating unit output. In other word, without resorting to another existing controller, it is proposed to construct an FLC that performs a prescribed task. To train the controller, the error between the actual and the desired plant output is back-propagated through the plant model to produce the error in control signal.

3. Besides the problem of parameter tuning, the selection of the number of MFs and inference rules is not a trivial task. Finding the optimum number of rules for a specific application is, to a large extent, a process of trial and error, relying mostly on past experience with similar application. Also, by increasing the number of MFs the size of adaptive network grows exponentially, requiring more training time. This problem becomes more crucial when the number of input variables increases.

In order to solve this problem, Genetic Algorithm (GA), as a global optimization technique, is employed to construct an ANF PSS with optimum structure. Since the number of rules depends, in a direct manner, on the number of MFs, the number and shape of MFs are determined first by applying GA. Then the parameters in the consequent part of the rule table are specified by the learning algorithm which is a special form of the gradient descent.

4. Behavior of an ANF PSS under single machine power system environment as well as multi-machine power system environment is observed. The coordination with other PSSs is also investigated.

5. In addition to the theoretical and simulation studies, the behavior of the proposed PSS in a physical model of the actual power system is examined. The ANF PSS has been implemented on a Digital Signal Processor (DSP) mounted on a PC. Consistency of the theoretical and simulation results with the experimental results exhibits the effectiveness of the ANF PSS to improve the dynamic performance of the system over a wide range of operating conditions.

1.5 Thesis Organization

This thesis is composed of 9 chapters divided into 3 parts:

- **Part I – Control Algoritms**

  Three most popular branches of Artificial Intelligence (AI), Fuzzy Logic, Adaptive Neural Network and Genetic Algorithm, are briefly reviewed in Chapter 2.

Details of the Adaptive-Network-Based Fuzzy Logic controller and its advantages compared with conventional fuzzy controllers are given in Chapter 3. An Adaptive-Network-Based Fuzzy Logic Controller is trained to tune the parameters of the Fuzzy Logic Controller. This approach combines the benefits of both Fuzzy Logic Control and Adaptive Neural Network. Furthermore, the self-learning technique is discussed in this Chapter. In this technique, the ANF PSS is trained from the performance of the generating unit output, rather than from the controller output.

Optimization of the ANF PSS structure is also discussed in Chapter 3. First, the necessity and advantages of optimized structure are described. Then, the application of Genetic Algorithm for the structure optimization of the ANF PSS is described.

- **Part II – Simulation Studies:**

This part consists of 4 chapters and focuses on the results obtained from simulation program. Application of the Adaptive-Network-Based Fuzzy Logic power system stabilizer to a single-machine power system is investigated in Chapter 4. The ability of the proposed stabilizer to provide enough damping over a wide operating range is discussed.

Chapter 5 gives the detailed simulation studies of the proposed controller trained using self-learning technique. Similar to Chapter 4, the learning method is basically a special form of gradient descent. However, instead of employing a desired controller, the plant output error signal is back-propagated to find the error of the controller output signal.

In Chapter 6, the simulation results, obtained by utilizing both Genetic Algorithm and Adaptive Neural Network to tune the parameters and optimize the structure of the fuzzy logic power system stabilizer, are presented.

The last Chapter of this part, Chapter 7, focuses on the simulation and detailed analysis of the proposed ANF PSS behavior in a multi-machine power system. Especially the behavior of the ANF PSS in response to different oscillation modes and the ability of the ANF PSS to work in cooperation with the conventional PSS and the other ANF PSSs are described.

- **Part III – Experimental Tests:**

  Laboratory implementation and experimental tests of the proposed ANF PSS on a physical model of a power system are described in Chapter 8. Real-time tests were performed on this model employing an ABB PHSC2 Programmable Logic Controller (PLC) as AVR and a Digital Signal Processor as a stabilizer. For comparison, a digital type conventional PSS (CPSS) was implemented in the same environment and tested under the same conditions. Behavior of the ANF PSS and the CPSS in an actual physical power system is observed and details of implementation along with the experimental results are described in this Chapter.

Finally, conclusions and comments on further research topics in the area of Adaptive-Network-Based Fuzzy Logic power system stabilizer are summarized in Chapter 9.

Part I

Control Algorithms

Chapter 2

Fuzzy Logic Control, Artificial Neural Networks and Genetic Algorithms - An Overview

2.1 Introduction

In recent years, Fuzzy Logic Control (FLC), Artificial Neural Network (ANN) and Genetic Algorithms (GAs), as three branches of Artificial Intelligence (AI), have attracted considerable attention as candidates for novel computational systems because of the variety of advantages that they offer over the conventional computational systems. This chapter covers the basics of these three areas, which are addressed in separate sections. Each section contains a brief historical perspective, functionality, characteristics and drawbacks of each branch.

Unlike classical design approach which requires a deep understanding of the system or exact mathematical models, fuzzy logic incorporates an alternative approach. Fuzzy logic control technique has been found to be a good replacement for conventional control techniques which require highly complicated nonlinear mathematical models. However, the design process of fuzzy controllers at some point becomes a trial-and-error approach. Such an approach requires a large number of repetitions, and it is therefore, time consuming and tedious.

Artificial Neural Networks are based on a simplified model of the brain, with the processing tasks distributed across many simple nodes. The power of an ANN comes from the collective behavior of the simple nodes. In addition to capability of learning and adaptation, this structure offers many other advantages including speed, robustness and fault tolerance. On the other hand, ANNs suffer from some drawbacks, among them is its "black-box" characteristic. It is difficult for an outside observer to understand or modify the network decision making process.

Genetic Algorithm is a probabilistic optimization approach inspired by biological evolution in nature. In general, genetic algorithms have proven to be more effective in solving a variety of complex multi-dimensional systems, which the other techniques have difficulty in solving. Particularly, GAs are successful at catching the optimum solution where the hyperspace is nonlinear, or highly convoluted with many local optima.

2.2 Fuzzy Logic Control

2.2.1 History of FLC

Fuzzy logic control is based on fuzzy set theory. In a symposium on system theory in Brooklyn 1965, L. A. Zadeh from the University of California, Berkeley, presented the fuzzy set theory. He believed that fuzzy logic would find home in psychology, philosophy, and in human science. He suggested it would play an important role in control [42]. Due to its name, fuzzy logic was not welcomed by many scholars in the beginning. Many people did not realize that fuzzy logic is not a logic that is fuzzy but a logic that describes fuzziness.

In 1973, Zadeh published his second most influential paper, which laid the framework for fuzzy logic control [35]. This paper, which he calls the key paper outlining a new approach to analysis of complex systems, showed how engineers and corporations could use fuzzy logic. In the same year, E. Mamdani and S. Assilian at the University of London succeeded in implementing the fuzzy if-then rules to control a steam engine. The results were superior to those using numerical methods and modeling [43].

In 1980, P. Halmblad and J. Ostergaard, Danish engineers, installed a fuzzy logic controller permanently in a cement kiln [44]. The pair had developed the first commercial application of fuzzy controller. Currently about 10% of the world cement kilns use this approach [42].

Fuzzy logic faded in the West, but Japanese picked up the idea and started applying it in early 1980s. In 1983 a fuzzy logic based water purification plant was put to work by M. Sugeno. In the same year he pioneered the application of fuzzy logic in robot control and a self-parking car [45][46]. In 1985, following the invention of the first fuzzy logic processing chip by Togai, S. Miyamoto and S. Yasunobu published a paper describing the automatic train operation by a predictive fuzzy control. The train started working in 1987 at Sandai subway system after thousands of computer simulation and actual runs on the track [47][48].

It was not until the late 1980s, that efforts were made to investigate fuzzy logic more intensively. B. Kosko formulated many concepts in fuzzy set theory and introduced the Fuzzy Associative Memory (FAM) which is a broader view of fuzzy rules [49].

Currently there is a substantial literature within the field of fuzzy set theory that deals with dynamic systems, control applications and system modeling [39][50][51].

2.2.2 Fuzzy Sets

A fuzzy set is a generalization of the concept of an ordinary bivalent set or *crisp* set. If C is a crisp set defined on the universe U, then for any element u of U, either $u \in C$ or $u \notin C$. For any crisp set C it is possible to define a characteristic function $\mu_C : U \rightarrow \{0, 1\}$. In fuzzy set theory, the characteristic function is generalized to Membership Function (MF) that assigns to every $u \in U$ a value from the unit interval [0,1] instead of from the two-element set $\{0, 1\}$. The set that is defined on the basis of such an extended membership is called a fuzzy set.

Let X be a fuzzy set and A and B be two fuzzy sets with the membership functions $\mu_A(x)$ and $\mu_B(x)$, respectively. Then the union, intersection and complement of fuzzy sets are respectively defined as:

$$\forall x \in X : \mu_{A \cap B}(x) = min(\mu_A(x), \mu_B(x)), \tag{2.1}$$

$$\forall x \in X : \mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x)), \tag{2.2}$$

$$\forall x \in X : \mu_{A'}(x) = 1 - \mu_A(x). \tag{2.3}$$

2.2.3 Linguistic Variables

A linguistic variable means a variable whose values are words in an artificial intelligence language. A linguistic variable is characterized by:

$$\langle x, T(x), U, M_x \rangle. \tag{2.4}$$

in which x denotes the symbol name of a linguistic variable, e.g. age, speed, temperature, etc. and $T(x)$ is the set of linguistic values that x can take. In the case of the linguistic variable temperature x, $T(x) = \{$cold, cool, comfortable, warm, hot$\}$. In the case of error or change-of-error it usually is the set {NB, NM, NS, ZO, PS, PM, PB}. U is the actual physical domain over which the linguistic variable x takes its quantitatives. In the case of temperature it can be the interval $[-10°C, 35°C]$ and in the case of error one often uses a normalized value [-1,1]. M_x is semantic function which gives an interpretation of a linguistic value in term of the quantitative elements of x. In other words, M_x is a function which takes a symbol as its argument, e.g. NB, and returns the meaning as "an error less than -0.8".

These terms can be characterized as fuzzy sets whose membership functions are shown in Fig. 2.1.

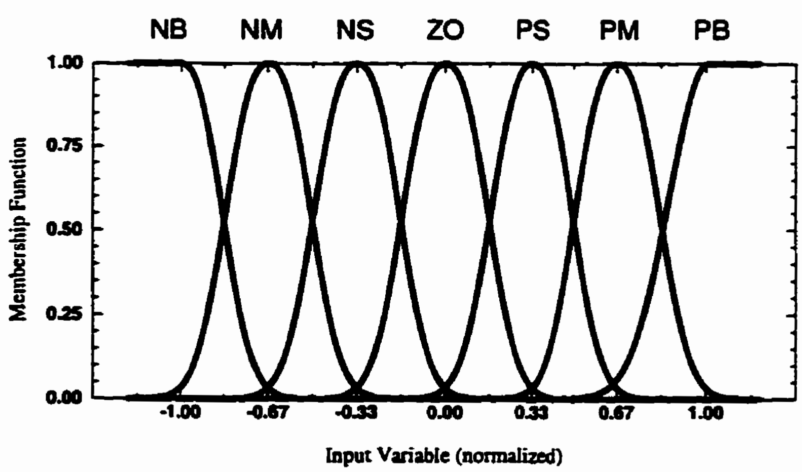

Figure 2.1: A typical set of gaussian membership functions

2.2.4 Fuzzy If-then Statements

A fuzzy if-then production rule is symbolically expressed as:

$$if\ \langle fuzzy\ propositon \rangle\ then\ \langle fuzzy\ propositon\ \rangle$$

where $\langle fuzzy\ propositon \rangle$ is a compound fuzzy proposition. For example if e and $\dot{e}$ are process state variables and u is the control output variable then:

$$if\ e\ is\ \text{NB}\ and\ \dot{e}\ is\ \text{PM}\ then\ u\ is\ \text{NS}.$$

Fig. 2.2 shows the domains of e and $\dot{e}$ and all the rules. In the case that e is PS and $\dot{e}$ is NS for example, the output field for u is ZO. Important properties for a set of rules are:

Error Derivative

Output Error	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

Figure 2.2: A typical set of fuzzy inference rules

- **Completeness** – any combination of input values results in an appropriate output value;
- **Consistency** – there are no two rules with the same rule-antecedent but different rule-consequent;
- **Continuity** – it does not have neighboring rules with output fuzzy sets that have empty intersection.

2.2.5 Basic Structure of Fuzzy Logic Controller

Fig. 2.3 shows the basic configuration of a Fuzzy Logic Controller (FLC), which comprises four principal components: fuzzification module, knowledge base, inference mechanism and a defuzzification module.

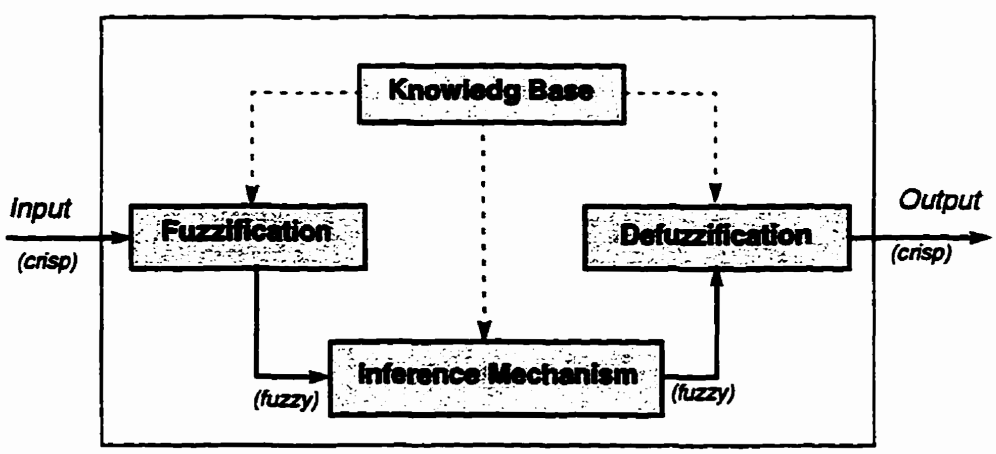

Figure 2.3: Basic structure of fuzzy logic controller

A. Fuzzification Module:

The fuzzification module performs the following functions:

- measures the values of input variables;

- performs a scale transformation (normalization) which maps the physical measured value into a normalized domain;
- using membership functions, converts the current value of a process state variable into a fuzzy set, in order to make it compatible with the fuzzy set representation of the process state variable in the rule-antecedent.

In fact, in fuzzification process, the input space is partitioned into sub-domains. Proper partitioning of this space requires some information about the system output state variables which is a part of knowledge base. Membership functions can be a variety of shapes, the most usual being triangular, trapezoidal or a bell shape (gaussian). The gaussian shape shown in Fig. 2.1 is used for the controller described in this thesis.

B. Inference Mechanism:

Inference mechanism plays an essential role in FLC. In this component, the membership values obtained in fuzzification step are combined through a specific T-norm, usually multiplication or minimization, to obtain the firing strength of each rule. Each rule characterizes the control goal and control policy of the domain experts by means of a set of linguistic control rules. Then, depending on the firing strength, the consequent part of each qualified rule is generated.

The most commonly used fuzzy inference mechanism can be classified into three groups:

1. Mamdani's Minimum Operation Rule

For simplicity, only two fuzzy control rules are assumed:

$$R_1\text{: if } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z \text{ is } C_1$$

R_2: if x is A_2 and y is B_2 then z is C_2

Then the firing strengths α_1 and α_2 of the rules can be expressed as:

$$\alpha_1 = \mu_{A_1}(x) \wedge \mu_{B_1}(y) \tag{2.5}$$

$$\alpha_2 = \mu_{A_2}(x) \wedge \mu_{B_2}(y) \tag{2.6}$$

where $\mu_{A_1}(x)$ and $\mu_{B_1}(y)$ are the degrees of membership for each input x and y. In this type [39], the ith rule leads to the control decision:

$$\mu_{C_i'} = \alpha_i \wedge \mu_{C_i}(w) \tag{2.7}$$

which implies that membership function μ_C is point wise given by

$$\mu_C = \mu_{C_1'} \vee \mu_{C_2'} = [\alpha_1 \wedge \mu_{C_1}(w)] \vee [\alpha_2 \wedge \mu_{C_2}(w)]. \tag{2.8}$$

To obtain a deterministic control action, a defuzzification strategy is required, as will be discussed later. This type of fuzzy reasoning process is shown in Fig. 2.4.

2. Tsukamato's Method with Linguistic Terms as Monotonic

As shown in Fig. 2.5, it is a simplified method of the first type in which the membership functions of fuzzy sets are monotonic [52]. The result inferred from each rule is α_i such that $\alpha_i = C_i(y_i)$ in which C_i is a monotonic fuzzy set. A crisp control action may be expressed as the weighted combination:

$$z = \frac{\alpha_1 z_1 + \cdots + \alpha_n z_n}{\alpha_1 + \cdots + \alpha_n}. \tag{2.9}$$

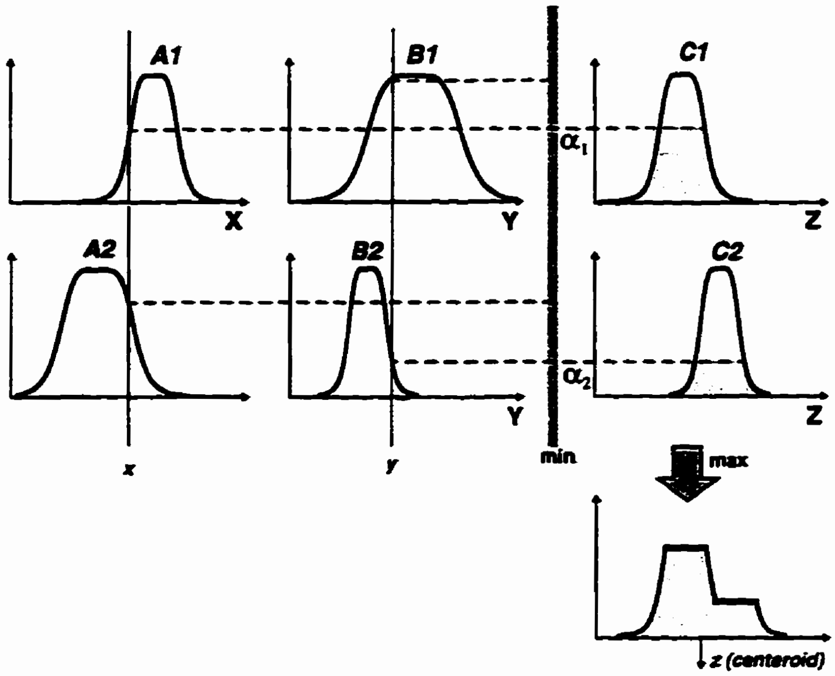

Figure 2.4: Mamdani fuzzy reasoning mechanism

3. Takagi and Sugeno's Method

In this method [36], shown in Fig. 2.6 the ith fuzzy control rule is of the form:

$$\text{if } x \text{ is } A_i \text{ and } \cdots \text{ and } y \text{ is } B_i \text{ then } z = f_i(x, \cdots, y)$$

where $x, \cdots, y$, and z are linguistic variables representing process state variables and the control variable, respectively; $A_i, \cdots, B_i$ are linguistic values of those variables. The final crisp control action is the weighted average of each rule:

$$z = \frac{\alpha_1 f_1(x, y) + \cdots + \alpha_n f_n(x, y)}{\alpha_1 + \cdots + \alpha_n}. \tag{2.10}$$

C. Knowledge Base:

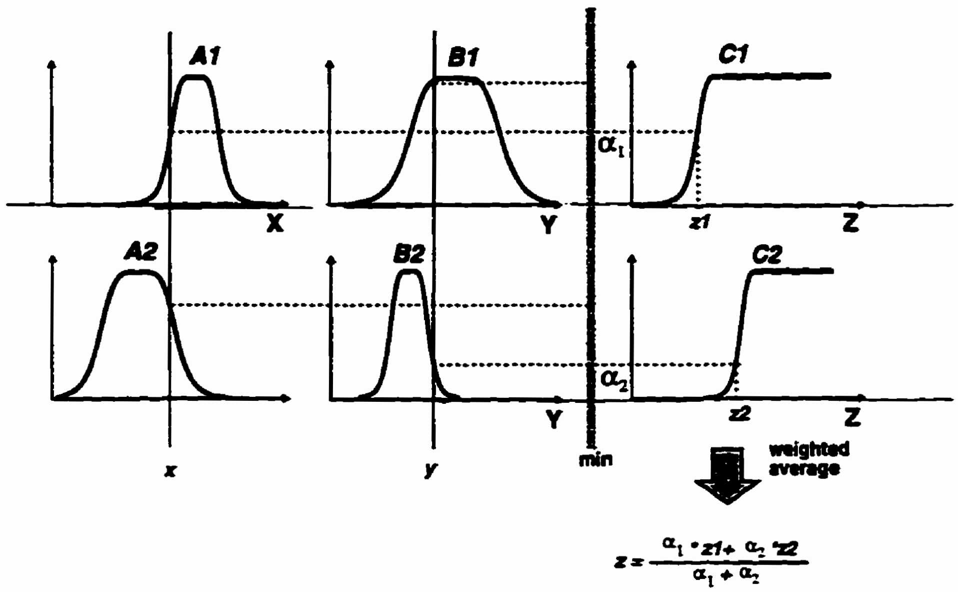

Figure 2.5: Tsukamato's fuzzy reasoning mechanism.

The knowledge base of an FLC consists of a data base. The basic function of the data base is to provide the necessary information for the proper functioning of fuzzification module, the inference engine, and the defuzzification module. This information includes:

- Fuzzy sets (membership functions) representing the meaning of the linguistic values of the process state and the control output variables.

- Physical domains and their normalized counterparts together with the normalization (scaling) factors.

D. Defuzzification Module:

The defuzzification module performs the following functions:

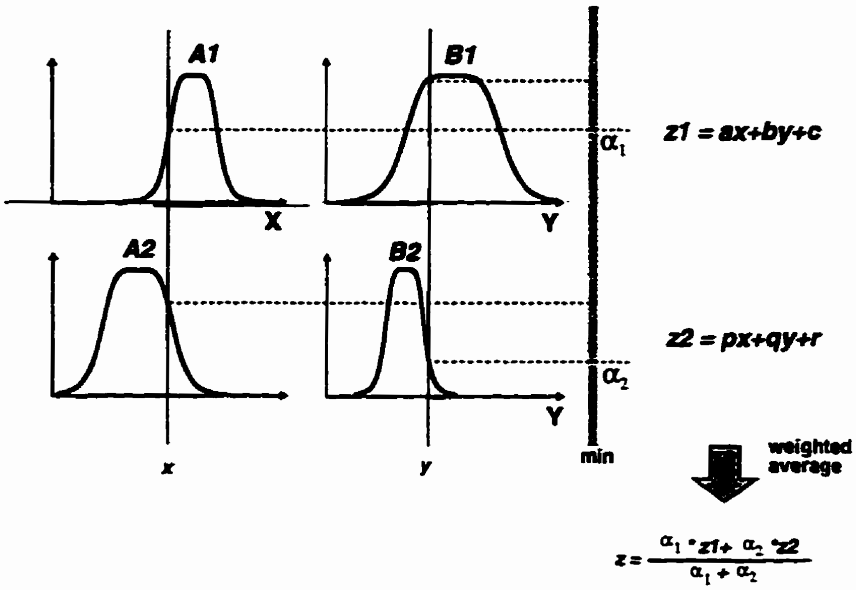

Figure 2.6: Takagi and Sugeno's fuzzy reasoning mechanism.

- converts the set of modified control output values into a nonfuzzy control action;
- performs an output denormalization which maps the range of values of fuzzy sets to the physical domain.

At present, there are three commonly used strategies for defuzzification as the max criterion, the mean of maximum, and the center of area. The widely used center of area strategy generates the center of gravity of the possibility distribution of a control action. In the case of a discrete domain, this method yields:

$$z_0 = \frac{\sum_{j=1}^{n} \mu_z(w_j).w_j}{\sum_{j=1}^{n} \mu_z(w_j)} \tag{2.11}$$

where n is the number of quantization levels of the output.

2.2.6 Characteristics of Fuzzy Logic Control

Considering the existing applications of fuzzy logic controller, which range from very small, micro-controlled based systems in home appliances to large-scale process control systems, the advantages of using fuzzy control usually fall into one of the following categories:

- **Robust nonlinear control** – A fuzzy logic controller, in general, has a non-linear transfer function. In fact, this is the most attractive feature that has made this controller very attractive. Basically, the source of non-linearity comes from rule base, though the fuzzy operators involved in fuzzification and defuzzification are also non-linear in nature. A representation theorem, mainly due to Kosko [49], states that any continuous nonlinear function can be approximated as exactly as needed with a finite set of fuzzy variables, values and rules.

  For the conventional PID controllers, a substantial parameter change or major external disturbance lead to a sharp decrease in performance. In presence of such disturbances, PID systems usually are faced with a trade-off between fast reactions with significant overshoot or smooth but slow reactions. In this case, fuzzy control offers ways to implement simple but robust solutions that cover a wide range of system parameters and that can cope with major disturbances.

- **Implementing expert knowledge** – In many cases of industrial process control, the degree of automation is quite low. There is a variety of conventional control loops, but a human operator is still needed. The knowledge of this operator is usually based on experience. In this case fuzzy control offers a method for implementing the expert's knowledge.

- **Reduction of development time** – Fuzzy control, which works at two levels of abstraction, offers languages at both levels of expertise: the symbolic level is appropriate for describing the application engineer's strategies, while the compiled level is well understood by the control engineers. Since there is a well-defined formal translation between these two levels, a fuzzy based approach can help reduce communication problems.

2.2.7 Limitations of Fuzzy Logic Control

Although, FLC introduces a good tool to deal with complicated, nonlinear and ill-defined systems, it suffers from the following drawbacks:

- At present, there is no systematic procedure for the design of FLC. The most straight forward approach is to define MFs and decision rules subjectively by studying an operating system or existing controller.

- In the case of too complex controlled system, the proper decision rules cannot easily be derived by human expertise.

- Designing and tuning a Multi-Input Multi-Output (MIMO) fuzzy logic controller is so tedious as to be unfeasible.

- In some situations, a reliable expert knowledge may not be available; even with relying on expert knowledge, fine tuning or achieving the optimal FLC is not a trivial task.

- Some significant operating changes, i.e. disturbances or parameter changes , might be outside the expert's experience.

2.3 Artificial Neural Network

2.3.1 History of ANN

Artificial Neural Networks and control community have a long history, which probably began with Weiener's book *Cybernetics*[53]. The first neuro controller was developed by Widrow and Smith in 1963 [54]. A simple ADAptive LINear Element (ADALINE) was taught to reproduce a switching curve in order to stabilize and control an inverted pendulum. This ADALINE was one of the first ANNs and it has a simple architecture that has been used extensively in other ANNs.

During 1970s, Albus proposed the CMAC as a tabular model of the functioning of the cerebellum and used it to control robotic manipulation. Since the early 1980s, the CMAC has been used extensively to model and control highly non-linear processes [55]. In 1980s, many different ANNs and IC architectures were proposed for integrating and extending these algorithms. Reinforcement learning and adaptive critic schemes have been extensively researched [56] and new ANNs such as the MLPs [30], RBFs [57], FLNs [58], and B-spline [59] have been developed. Recurrent networks have been used in optimization schemes and for plant modeling and estimation [60].

ANNs have made a significant impact on the industry, with the applications in non-linear process and human operator modeling, automatic plant knowledge elicitation, fault detection and monitoring, process control and optimization and sensor validation, interpretation and fusion [61].

2.3.2 Basic Elements

Neurons are the basis of the neural networks. A neuron is an information-processing unit that is fundamental to the operation of a neural network. Fig. 2.7 shows the model for a neuron. There are three basic elements of the neuron model, as described here:

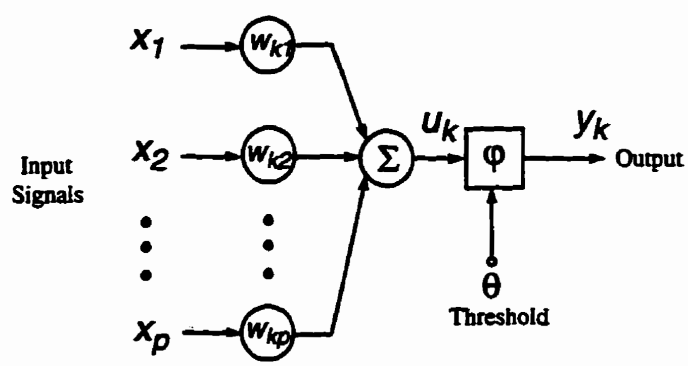

Figure 2.7: Nonlinear model of a neuron

- A set of synapses or connecting links, each of which is characterized by a weight. A signal x_j at the input of the synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}.
- An adder for summing the input signals, weighted by the respective synapses.
- An activation function for limiting the amplitude of the output of a neuron. This limit usually is in the unit interval [0,1] or alternatively [-1,1].

This model also includes an externally applied threshold θ_k that has the effect of lowering the net input of the activation function. In mathematical terms, a neuron k can describe by the following pair of equations:

$$u_k = \sum_{j=1}^{p} w_{kj} x_j \tag{2.12}$$

and

$$y_k = \phi(u_k - \theta_k) \tag{2.13}$$

where $x_1, x_2, \cdots, x_p$ are the input signals; $w_{k1}, w_{k2}, \cdots, w_{kp}$ are the synaptic weights of neuron k, u_k is the linear output; θ_k is the threshold; $\phi(.)$ is the activation function; and y_k is the output signal of the neuron. There are three basic types of activation functions:

1. Threshold Function:

$$\phi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \tag{2.14}$$

2. Piece-wise-Linear Function:

$$\phi(v) = \begin{cases} 1 & \text{if } v \geq \frac{1}{2} \\ v & \text{if } \frac{1}{2} > v \geq -\frac{1}{2} \\ 0 & \text{if } v < -\frac{1}{2} \end{cases} \tag{2.15}$$

3. Sigmoid Function:

$$\phi(v) = \frac{1}{1 + exp(-av)} \tag{2.16}$$

where a is the slope parameter of the sigmoid function.

2.3.3 Network Architectures

The manner in which the neurons of a neural network are connected can be classified into two architectures:

1. **Feedforward Networks** – In this type of network, outputs of every layer are projected to the inputs of the next layer, but not vice versa, as shown in Fig. 2.8.

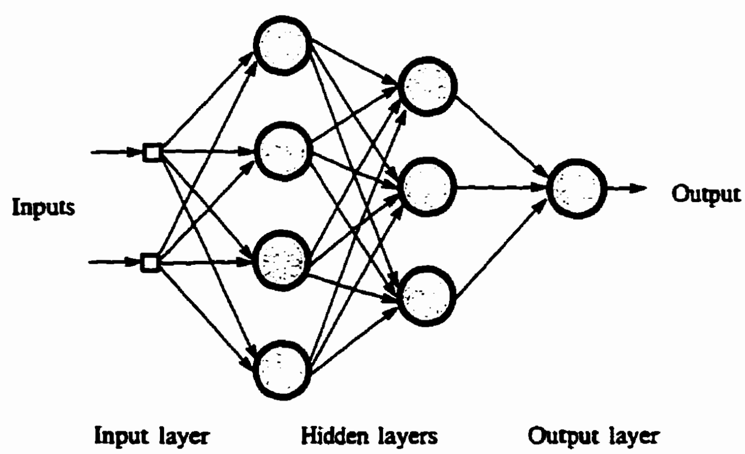

Figure 2.8: Feedforward neural network with two hidden layers.

In other words, this network is strictly of a feedforward type. Usually, the network consists of an input layer, one or more hidden layers and an output layer. By adding one or more hidden layers, a feedforward network is enabled to extract higher-order statistics. The source nodes in the input layer supply the input signals to the network, and the neurons in the output layer constitute the overall response of the network. In term of node's connection, the network can be fully connected or partially connected.

2. **Recurrent Networks** – The main difference between a recurrent neural network and a feedforward neural network is that the recurrent neural network has at least one feedback loop. In Fig. 2.9, the recurrent network is shown with feedback loops. This has a profound impact on the learning capability of the network, and on its performance. Moreover, the feedback loops involve the use of unit-delay element, which results in a nonlinear dynamic behavior of the network.

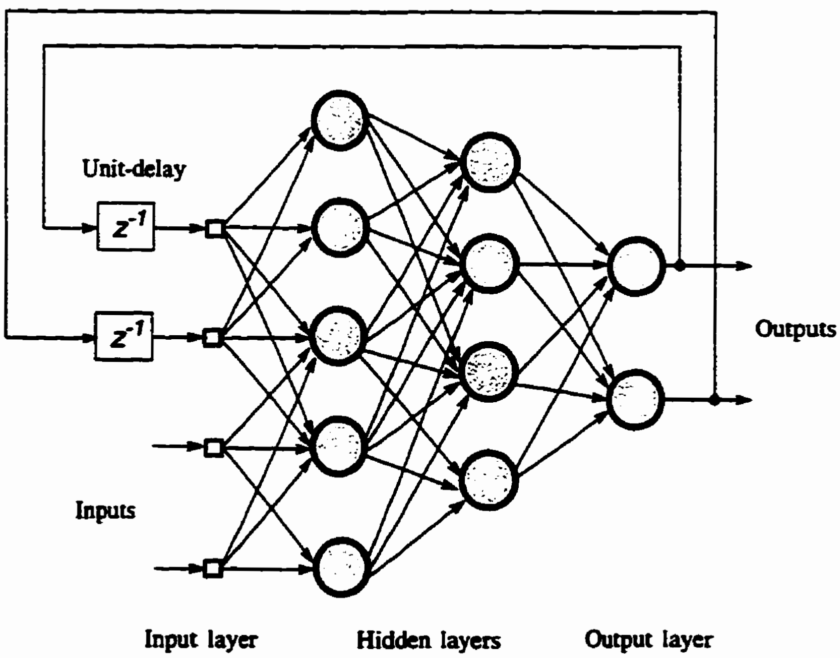

Figure 2.9: Recurrent network with hidden neurons.

2.3.4 Training Algorithms

Among the many properties of a neural network, the property that is of primary significance is the ability of the network to learn from training data, and to improve its performance through learning. There are basically three classes of learning paradigms:

1. **Supervised Learning** – As it implies, supervised learning is performed under the supervision of an external teacher. The network parameters are adjusted under the combined influence of the training data and error signal; the error signal is defined as the difference between the actual response of the network and the desired response.

   Examples of supervised learning algorithms include the Least-Mean-Square (LMS) algorithm [62] and its generalization known as the Back-Propagation (BP) algorithm [63]. The back-propagation algorithm derives its name from the fact that error terms in the algorithm are back-propagated through the network, on a layer-by-layer basis.

   Supervised learning can be performed in an off-line or on-line manner. In the off-line case, once the desired performance is accomplished, the training is frozen, which means the neural network operates in a static manner. On the other hand, in on-line training, learning is accomplished in real time, with the result that the neural network dynamically adjusts the parameters .

2. **Reinforcement Learning** – Reinforcement learning involves the use of a critic that evolves through a trial-and-error process. Compared to supervised

learning, the learning is done on the basis of the reinforcement received from the environment; there is no teacher to supply gradient information during learning. To obtain information, a reinforcement learning system probes the environment through the combined use of trial-and-error and delayed reward. This learning approach is more suited in less-structured situations where it may be possible to improve plant performance over time by means of on-line reinforcement learning [64].

3. **Unsupervised Learning** – Unsupervised learning is performed in a self-organized manner in which no external teacher or critic is required to instruct the network. Rather, provision is made for a task-independent measure of the quality of representation that network is required to learn. In other words, by using unsupervised learning, the network is able to form the underlying structure of the input data in an explicit or simple form. The two most important unsupervised network architectures are Kohonen's Self-Organizing Map [65] and Grossbeerg's ART networks [66].

2.3.5 Different Control Schemes

There are different control schemes to train a neural network to control a plant that is too complex, or about which too little is known. In a typical control problem, one may have desired plant output but not the desired neural network output, which is the control signal. Three basic ways in which the training information required for supervised learning can be obtained are given below:

1. **Copying an Existing Controller** – If there exists a controller capable of controlling the plant, then the information required to train a neural network can be obtained from this controller as shown in Fig. 2.10. The desired network output for a given input is the output of the existing controller for that input. The network learns to copy the existing controller.

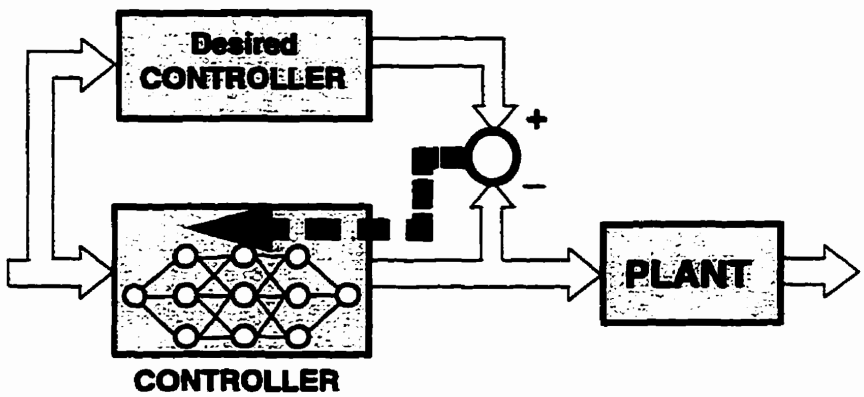

Figure 2.10: Copying an existing controller with a network.

One might question the utility of this method on the ground that if there already exists an effective controller, why would it be useful to have another one in the form of a neural network? Two answers are apparent. First, the existing controller may be a device that is impractical to use; like an artificial intelligent based controller with a large number of inference rules. Second the existing controller may use very complicated algorithms to calculate the control signal, forming a large delay in control response.

2. **Identification of System Inverse** – Fig. 2.11 shows how a neural network can be used to identify the inverse of a plant. The input to the network is the output of the plant, and the desired output is the plant input. If the network

can be trained to match these targets, it will implement a mapping that is a plant inverse. Once one has such an inverse, it can be used for control purposes; the desired plant output is provided as input to the network and the resulting network output is then used as input to the plant.

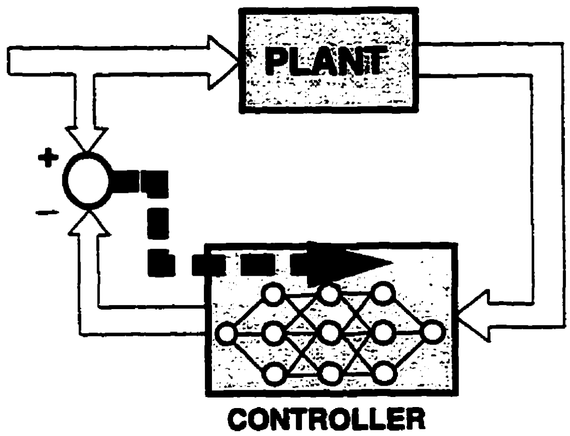

Figure 2.11: Inverse plant modelling using a network.

A major problem with this approach arises when different plant inputs produce the same output, i.e., when the plant's inverse is not well defined. In this case neural network will attempt to map the same network input to many different desired responses.

3. **Differentiating a Model** – This method of training a controller relies more on backpropagation than on general network methods [67]. The method is illustrated in Fig. 2.12. The backpropagation algorithm is used to identify the plant, resulting in a forward model of the plant in the form of a layered network. Thus the derivative of the model's output with respect to its input can be computed by the backpropagation process. Propagating errors between

actual and desired plant outputs back through the forward model produces the error in the control signal. This error is used to train the controller.

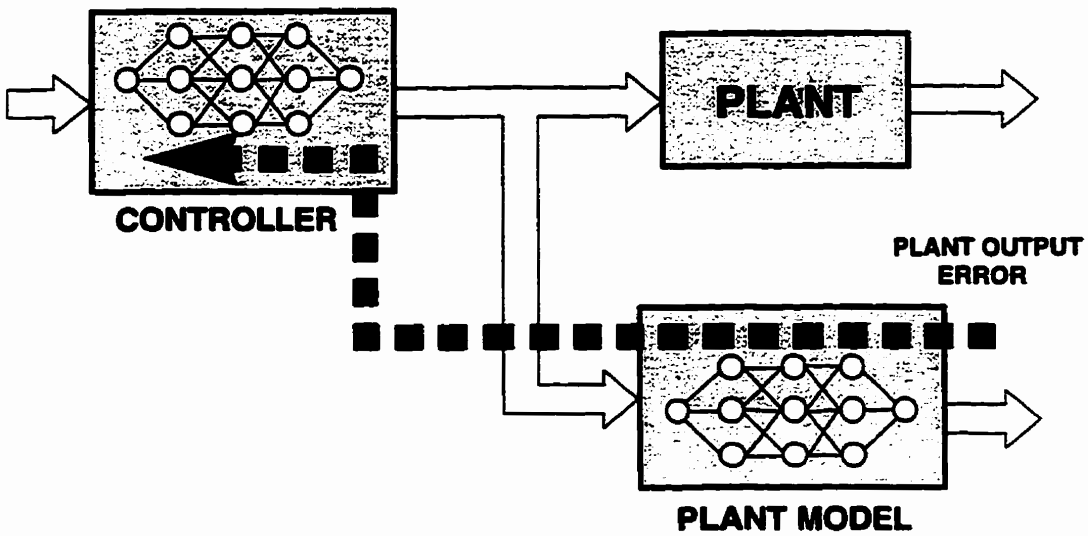

Figure 2.12: Backpropagating through a forward model of the plant.

In Fig. 2.12 this backpropagation process is illustrated by the dashed line passing back through a second neural network. Of course, to apply this idea one needs a model in a form that can be differentiated. This method is discussed in more detail in Chapter 3, using an adaptive fuzzy logic controller.

2.3.6 Characteristic of ANNs

Neural networks offer solutions to problems that are very difficult to solve using traditional algorithms. The potential benefits of a neural approach are:

- **Nonlinearity** – A neuron is basically a nonlinear element. Consequently, a neural network, made up of an interconnection of neurons, is itself nonlinear.

- **Learning** – Neural networks can learn from the interaction with the environment, rather than explicit programming, it learns from the examples by constructing an input-output mapping for the problem at hand.
- **Complex Mapping** – It has the capability of synthesizing complex mappings which may be very difficult or even impossible to be expressed in mathematical form.
- **Generalization** – It is able to generalize the training information to similar situations in which it has never experienced before.
- **Speed** – Due to the parallel mechanism, once an ANN is trained, it can provide the ability to solve the mapping problem much faster than conventional methods and other artificial methods.
- **Robustness and fault tolerance** – Even if the input data are incomplete or noisy, the ANN can still provide satisfactory results. Also, due to distribution of computational load across many simple processing elements, the networks possess some degree of fault tolerance with respect to processor failures.
- **VLSI Implementable** – The massively parallel nature of a neural network makes it ideally suited for implementation using Very Large Scale Integrated (VLSI) technology.

2.3.7 Limitations of ANN

Some of the advantages mentioned above, such as learning ability, cannot be found in the fuzzy logic controllers. However, ANNs do have some limitations as listed below:

- **Black Box** – The major draw back of neural networks is black-box characteristic. It is not easy to understand the knowledge stored in an ANN. Training sets rarely contain a complete description of the desired input-output relationship and once learning has ceased, it may be necessary to modify the stored information. This can only be performed if the knowledge is stored in a transparent fashion.

- **Long training time** – ANNs may require a long training time to obtain the desired performance. The larger the size of ANN and the more complicated the mapping to be performed, the longer the training time required.

- **Network structure** – The selection of number of hidden layers and number of neurons in each layer is not a trivial task. It is, to a large extent, a process of trial and error.

2.4 Genetic Algorithms

Genetic Algorithms are search algorithms which are based on the genetic processes of biological evolution. They work with a population of individuals, each representing a possible solution to a given problem. Each individual is assigned a fitness score according to how well it solves the given problem. For instance, the fitness score might be a performance index for a closed loop control system. In nature, this is equivalent to assessing how effective an organism is at competing for resources. The highly adapted individuals will have relatively large numbers of offsprings. Poorly performing ones will produce few or even no offspring at all. The combination of selected individuals produces superfit offsprings, whose fitnesses are greater than that of the parents. In this way, the individuals evolve to become more and more well suited to their environment.

2.4.1 History of GAs

The underlying principles of GAs were first published by Holland in 1962 [68]. The mathematical framework was developed in the late 1960's, and presented in Holland's pioneering book in 1975 [69]. GA's have been used in many diverse areas such as function optimization [70], image processing [71] and system identification [72] [73]. In the last decade, research devoted to GAs has significantly increased, as attested by the existence of several conferences on the topic. An excellent reference on GAs and their implementation is Goldberg's book [74].

2.4.2 Basic Principles

The standard GA can be represented as shown in Fig. 2.13. In what follows, different steps of the algorithm are briefly explained:

- **Coding** – To translate a problem into a suitable form for a GA, a potential solution should be represented as a set of parameters. These parameters (known as genes), after transformation to binary value, are joined together to form a string of values (known as chromosome). The choice of the genetic coding is crucial to the performance of the genetic algorithm, as the genetic coding defines the window through which the algorithm connects to the actual problem.

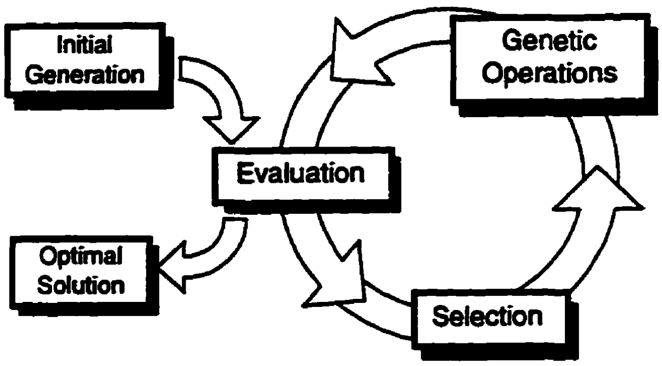

Figure 2.13: Mechanism of genetic algorithm.

- **Evaluation** – The first step in every iteration of a genetic algorithm is to determine how well each individual can solve the problem. A fitness function must be defined and return a single numerical fitness, which is supposed to be proportional to the ability of the individual. For function optimization, the fitness function should be the value of the function. The result of this evaluation is used to specify how many offsprings should be generated by an individual.

- **Reproduction** – In this step, three genetic operators are applied to the current population:

  1. *Selection:* The individuals of higher quality are more likely to be chosen for reproduction than those of lower quality. A number of exact copies are generated with the best individuals producing the most copies. As a result, good individuals might be selected several times while poor ones may not be chosen at all. In this thesis, the selection method called the ranking scheme has been chosen. In this technique, each individual is ranked

based on its fitness. Then depending on its rank, each individual produces a specific number of offsprings. By using this technique, the fittest individuals cannot dominate the population within a single generation.

2. *Crossover:* This operator combines two previously selected individuals as shown in Fig. 2.14 and yields the offsprings. This operator tries to combine vital parts of two individuals in order to create a superior individual. During the crossover operation two points in the strings are randomly chosen and the part, which is enclosed by two points, is swapped. Crossover is not usually applied to all individuals; a random choice is made with a certain probability.

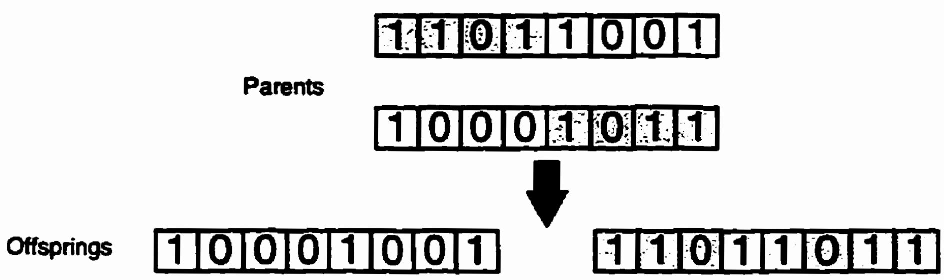

Figure 2.14: Crossover operation.

3. *Mutation:* Mutation is used to introduce new solutions and prevent the population from unrecoverable loss of important information. Mutation is accomplished by flipping single bits of the string as shown in Fig. 2.15 with a certain probability. The probability for mutation is usually kept low to prevent a negative influence on the crossover operation.

While crossover roughly establishes the region of the search space, which contains the solution, mutation is additionally useful for fine tuning at the end of

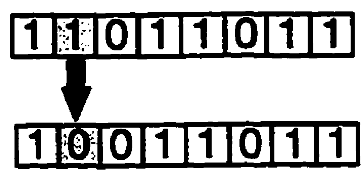

Figure 2.15: Mutation operation.

the optimization.

2.4.3 Characteristic of GAs

- GA is a global optimization method which is specifically useful for discontinuous cost functions. The other optimization techniques, like gradient descent method, rely heavily on the differentiability of the cost function.
- Compared to other conventional search algorithms, genetic algorithm considers many points in the search space simultaneously. Also it uses probabilistic rules not deterministic rules to guide its search. For these two reasons, genetic algorithms have a reduced chance of converging to local optima.

2.4.4 Limitations of GAs

- The optimal solution is usually determined by going through a number of generations. However, the number of generations necessary to ensure that the most-fit individual is found is a priori unknown.
- Since there are many parameters involved in the algorithm, there is no guarantee that the genetic algorithm can reach a near-optimal solution. If the parameters are not properly selected, it can fall into a local optimal point depending on the topology of the search space.

2.5 Summary

The basic concepts and theories of three branches of artificial intelligence, fuzzy logic control, artificial neural networks and genetic algorithms are introduced in this Chapter. The fundamental procedure for each one is explained. Also their benefits as well as their limitations are given.

Nonlinearity and knowledge based are among the most important characteristics of fuzzy logic control. However, it suffers from the drawback of parameter tuning. There are many parameters, including membership functions and rule base, to be tuned. On the other hand, artificial neural network has the capability of learning. It is shown in Chapter 3, by proper combination of these two techniques, the drawbacks of each one can be mostly compensated by the benefits of the other technique.

As mentioned in this Chapter, for designing an artificial neural network, one faces the issue of network structure setup. Selection of a large number of neurons (or hidden layers) leads to the problem of overfitting and long training time as well. Too small network, on the other hand, bring the situation that the network is not able to learn the desired input-output relationship at all. Genetic algorithm, as a powerful technique for multi-criterion optimization problems, can be used to automate the design of neural network architecture.

Chapter 3

Adaptive Fuzzy Logic Controller

3.1 Introduction

In Chapter 2, static fuzzy logic systems are explained. Their success is due to the fact that inherently nonlinear control strategies can be obtained from human expert and then implemented as a fuzzy controller. The strengths and weaknesses of this approach are explained as well. Obtaining the rules from an expert, known as *knowledge elicitation*, is one of the major bottlenecks in the development of fuzzy logic control. Frequently, the fuzzy algorithms provided by experts are not correct, relevant and complete. These problems can be overcome using adaptive fuzzy systems which automatically find an appropriate set of rules and membership functions [75][76].

Adaptive fuzzy system is implemented in the framework of adaptive network architecture and equipped with a training (adaptation) algorithm. Training input data are presented to the network and the network computes its output. Error between the system's output and the desired output is calculated, and finally the error is back-propagated through the whole network to adjust the network parameters such that the output error reduces at each step.

Similar to ANN, there are different approaches to train an adaptive fuzzy controller. The most straight-forward approach is to train the controller using another existing desired controller. However, in a general situation, the desired controller or domain expert may not be available. Therefore, a self-learning approach has to be constructed in order to train adaptive fuzzy controller without resorting to other existing controllers [77].

Although adaptive fuzzy systems offer the potential solution to the knowledge elicitation problem, fuzzy systems still suffer from advanced setting of the structure of fuzzy system. The structure, expressed in term of the number of membership functions and number of inference rules, is usually derived by trial and error. When the number of inference rules is small, the inference rules cannot describe the input-output relationship of given data precisely. On the contrary, when the number of inference rules is large, the generalization capability of the inference rules is sacrified because of the overfitting problem. Therefore, the number of inference rules has to be determined from a standpoint of overall learning capability and generalization capability. In order to solve this problem, a genetic algorithm is employed to automate the design method for optimizing the structure of fuzzy system [78].

In this chapter, first the structure of adaptive fuzzy system is explained, and necessity of using a self-learning algorithm to train the controller is given. Then, the self-learning adaptive fuzzy controller based on back-propagation through time is formulated. Finally, genetic algorithm is described to construct an adaptive fuzzy controller with optimum structure.

3.2 Fuzzy Logic Controller with Learning Ability

Artificial Neural Networks have elicited strong interest among researchers over the last decade. One reason for this resurgent interest is the discovery of a powerful training algorithm for multilayer neural networks - the so-called back-propagation algorithm. In fact, the basic concept of back-propagation algorithm can be applied to any feedforward network. Therefore, if the fuzzy logic systems can be represented as feedforward networks, the idea of back-propagation can be used to train them. This is the motivation of the training algorithm for fuzzy logic controller in this section.

3.2.1 Structure of Adaptive Fuzzy Controller

By observing the functional form of fuzzy controller, it becomes apparent that the fuzzy controller can be represented as a five-layer feedforward network as shown in Fig. 3.1. With this network representation of the fuzzy logic system, it becomes straightforward to apply back-propagation to adjust the parameters in membership functions and inference rules.

For simplicity, assume that the fuzzy controller has two inputs x_1 and x_2 and one output z. Each fuzzy if-then rule is of Takagi and Sugeno's type [36]:

If x_1 is A_j and x_2 is B_k, then $f_i = p_i x_1 + q_i x_2 + r_i$;

where A_j and B_k are linguistic variables, f_i is the output of the ith rule and $\{p_i, q_i, r_i\}$ is the inference rule's parameter set. The node functions in each layer are of the same type function as described below:

Layer 1 – Each node in this layer performs a MF:

$$y_i^1 = \mu_{Ai}(x_i) = exp\left\{-\left[\left(\frac{x_i - c_i}{a_i}\right)^2\right]^{b_i}\right\} \tag{3.1}$$

where x_i is the input of node i, A_i is linguistic label associated with this node and $\{a_i, b_i, c_i\}$ is the parameter set of the bell-shaped MF. y_i^1 specifies the degree to

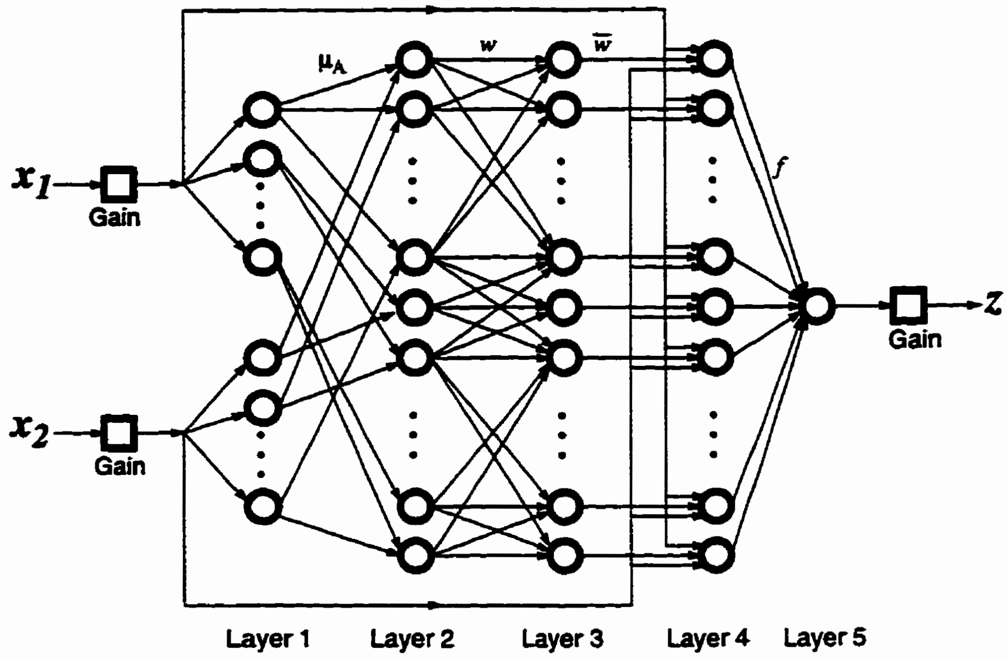

Figure 3.1: Architecture of adaptive fuzzy controller.

which the given input belongs to the linguistic label A_i, with maximum equal 1 and minimum equal to 0. As the values of these parameters change, the bell-shaped function varies accordingly, thus exhibiting various forms of membership functions. In fact, any continuous and piecewise differentiable functions, such as trapezoidal or triangular membership functions, are also qualified candidates for node functions in this layer.

Layer 2 – Every node in this layer represents the firing strength of the rule. Hence, the nodes perform the fuzzy AND operation:

$$y_i^2 = w_i = min(\mu_{Ai}(x_1), \mu_{Bi}(x_2)). \tag{3.2}$$

Layer 3 – The nodes of this layer calculate the normalized firing strength of each rule:

$$y_i^3 = \bar{w}_i = \frac{w_i}{\sum_{i=1}^{n} w_i}. \tag{3.3}$$

Layer 4 – Output of each node in this layer is the weighted consequent part of the rule table:

$$y_i^4 = f_i = \bar{w}_i(p_i x_1 + q_i x_2 + r_i) \tag{3.4}$$

where $\bar{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set.

Layer 5 – The single node in this layer computes the overall output as the summation of all incoming signals:

$$y_i^5 = \sum_{i=1}^{n} f_i \tag{3.5}$$

Thus a fuzzy logic controller with learning capability has been constructed. In order to achieve a desired input-output mapping, these parameters are updated according to the given training data and a gradient-based learning procedure described below.

Assuming that the training data set has P entries and the output layer has only one node, the error measure for the pth entry of training data:

$$E_p = \frac{1}{2}(\hat{y_p} - y_p^L)^2 \tag{3.6}$$

where $\hat{y_p}$ is the pth component of desired vector and y_p^L is the pth component of actual output vector. For each training data, a forward pass is performed and then starting at the output layer, a backward pass is used to compute $\frac{\partial E_p}{\partial y_p}$ for all internal nodes. For the output node:

$$\frac{\partial E_p}{\partial y_p^L} = -(\hat{y_p} - y_p^L) \tag{3.7}$$

and for the internal nodes in layer k:

$$\frac{\partial E_p}{\partial y_{i,p}^k} = \sum_{m=1}^{K1} \frac{\partial E_p}{\partial y_{m,p}^{k+1}} \frac{\partial y_{m,p}^{k+1}}{\partial y_{i,p}^k} \tag{3.8}$$

where $y_{i,p}^k$ is the output of the node in the ith position of kth layer which has K nodes and $K1$ is the number of nodes in $(k+1)$th layer.

Assuming α is a parameter of the adaptive network:

$$\frac{\partial E_p}{\partial \alpha} = \sum_{y^* \in S} \frac{\partial E_p}{\partial y^*} \frac{\partial y^*}{\partial \alpha} \tag{3.9}$$

where S is the set of nodes whose outputs depend on α. The goal is to minimize the overall error $E = \sum_{p=1}^{P} E_p$ by using the general learning rule :

$$\Delta\alpha = -\eta \frac{\partial E}{\partial \alpha} \tag{3.10}$$

in which η is the learning rate and

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \alpha} \tag{3.11}$$

Also, similar to the training of conventional neural networks [30], a momentum term is added for a better convergence:

$$\Delta\alpha(t) = -\eta \frac{\partial E}{\partial \alpha} + \beta \Delta\alpha(t-1). \tag{3.12}$$

where β is the momentum factor and $\Delta\alpha(t-1)$ is the change of α in the last step.

Now, the fuzzy logic system with the above mentioned gradient descent method can be used as an identifier for nonlinear dynamic systems or as a nonlinear controller with adjustable parameters.

3.2.2 Advantages of Adaptive FLC

There are many advantages in using adaptive fuzzy control systems over the static fuzzy control systems and the neural network based controllers:

- In a situation where there is a large uncertainty or unknown variation in plant parameters and structure, a fuzzy logic controller should be able to adjust its parameters to maintain consistent performance of the system. Therefore, fuzzy controller need to be capable of learning.

- Fuzzy rules obtained from human operator are not precise and may not be sufficient for constructing a successful controller. They provide very important information about how to control the system, however they need to be carefully tuned. Adaptive fuzzy control provides a tool for making use of the fuzzy information in a systematic and efficient manner.

- Compared to the conventional adaptive controllers, the major advantage of adaptive fuzzy control is that the ability to incorporate linguistic fuzzy information from a human expert.

- Compared to the conventional artificial neural networks, the parameters of neural network controller have no clear relationships with input-output data, and therefore their initial values are usually chosen randomly. On the other hand, the parameters of adaptive fuzzy controller have clear physical meanings. By incorporating the knowledge base as initial parameters for adaptive fuzzy controller, the speed of convergence is dramatically increased.

3.3 Self-Learning Adaptive Fuzzy Logic Controller

Tt is necessary to know the error in the controller output, $(u_d - u)$, to train an adaptive fuzzy controller. This approach requires the existence of the desired controller which restricts the application domains of adaptive fuzzy controllers. To overcome this problem, a separate adaptive fuzzy identifier is trained to behave like the plant. The block diagram of Fig. 3.2 shows two adaptive fuzzy systems, one acting as the controller and the other acting as the plant identifier. This identification is similar to plant identification in adaptive control theory, except that the plant identification is done automatically by an adaptive fuzzy system capable of modeling non-linear plants.

The utility of this plant identifier is that it can compute the derivative of the plant's output with respect to the plant's input by means of the back propagation process. The final output error of the plant, $(z_d - z)$, is back-propagated through the adaptive fuzzy identifier to obtain the equivalent error for the controller's output. In this figure, back-propagation process is illustrated by the dashed line passing through the forward identifier and continuing back through the adaptive fuzzy controller that uses it to learn the control rule.

Another approach to produce a suitable descent direction at the output of adaptive fuzzy controller, to use the plant Jacobian, or sensitivity derivative [79]. If the cost function is defined as $J(w)$, then, knowing the Jacobian of the plant, the gradient of the cost function with respect to the control output, u, is easily determined as:

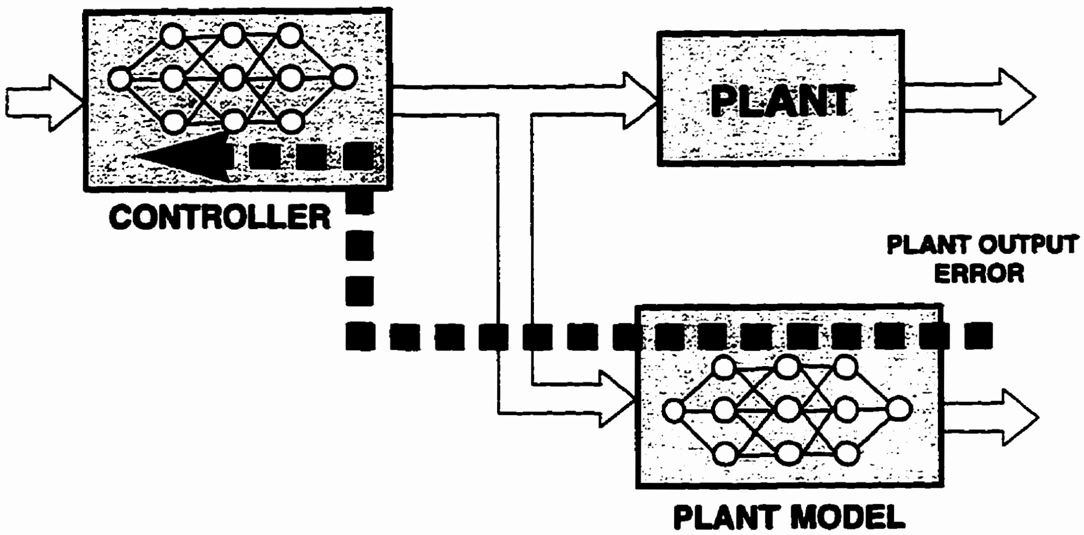

Figure 3.2: Back-propagating through a forward model of the plant.

$$\frac{\partial J(w)}{\partial u} = \sum_{i=1}^{n} \frac{\partial J(w)}{y_i} . \frac{\partial y_i}{\partial u}. \tag{3.13}$$

where y_i is the ith plant output. If little is known about the plant, it would be difficult to obtain an analytical expression for the plant Jacobian. Numerical differentiation could be used to form an approximation to the Jacobian, but would suffer from the large errors that plague such a technique.

Another technique [80] is to use the sign of the Jacobian, instead of its real value, for the training of adaptive fuzzy controller. This is often available simply from qualitative knowledge of the system in question. The plant backpropagation equation then becomes:

$$\frac{\partial J(w)}{\partial u} \simeq \sum_{i=1}^{n} \frac{\partial J(w)}{y_i} .SGN(\frac{\partial y_i}{\partial u}). \tag{3.14}$$

Among these methods, using an adaptive fuzzy identifier is preferred because of two reasons. First, the backpropagation mechanism can be employed to adjust the plant identifier as well as the controller. Therefore, the plant identifier can follow any change or large disturbance in the actual plant, hence the backpropagated error becomes more accurate. Secondly, by incorporating the prior knowledge of the plant into the adaptive fuzzy identifier, the training time for identifier is decreased dramatically.

There are two learning paradigms for training the self-learning adaptive fuzzy controller. With the on-line training, the parameters of the controller are updated immediately after each sampling time has been passed. On the other hand, in back-propagation through time, the parameters are updated after a certain elapsed time [77] [81].

Given the state of the plant at time $t = k * h$, adaptive fuzzy controller will generate an input to the plant and the plant will evolve to the next state at time $(k+1) * h$. By repeating this process starting from $t = 0$, a plant state trajectory is determined by the initial state and the parameters of adaptive fuzzy controller. The state transition from $t = 0$ to $m * h$ is shown in Fig. 3.3, which contains m sampling states of the plant.

Accordingly, the back-propagation gradient descent is applied to minimize the difference between the plant trajectory output and the desired trajectory. In this way, the corresponding error to be minimized is:

$$E = \sum_{k=1}^{m} \|z(h * k) - z_d(h * k))\|^2 + \lambda * \sum_{k=0}^{m-1} \|u(h * k)\|^2, \tag{3.15}$$

where $z_d(h * k)$ is the desired trajectory, $u(h * k)$ is the controller's output at time $t = h * k$. By a proper selection of λ, a compromise between trajectory error and

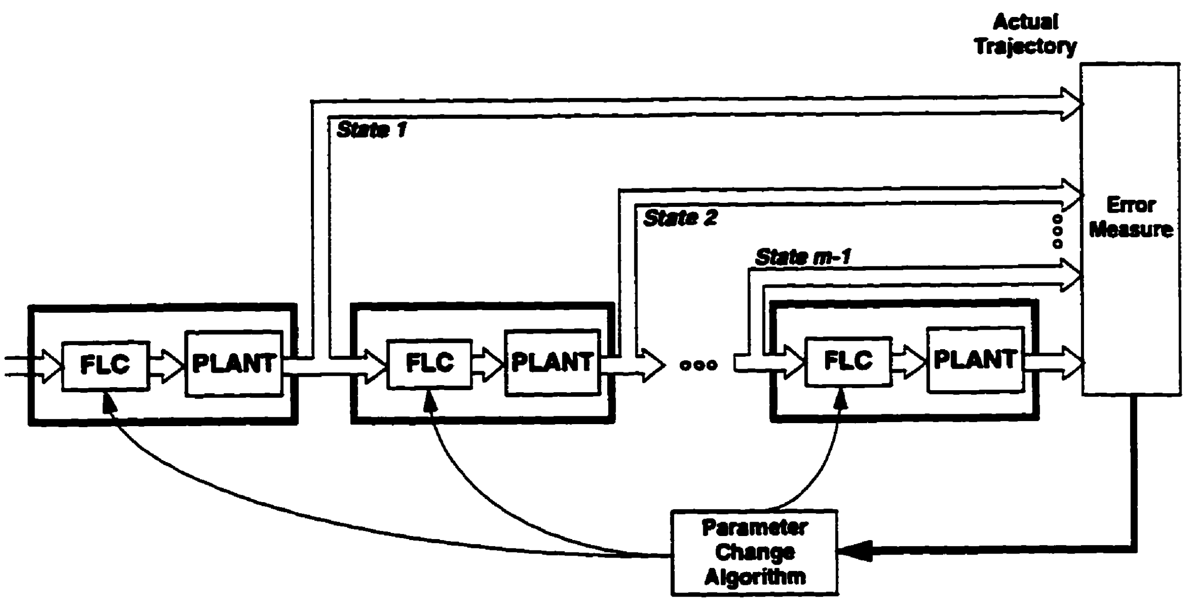

Figure 3.3: The plant trajectory of self-learning adaptive fuzzy controller

control signal can be obtained.

The parameter changes from all the stages obtained from the back-propagation algorithm are added together and then added to the controller's parameters.

3.4 Genetic Optimization of Adaptive Fuzzy Controller

3.4.1 Problem of Structure

Generally, training of any type of adaptive network involves the selection of an optimal network structure. Usually the designer has to search for the optimal structure by trial and error. This search causes a large number of experiments. If the selected network is too large, it may fail to generalize because it has too many degrees of freedom. A large number of parameters often allows a network to initially learn to detect global features of the input-output mapping, and as a consequence generalize quite well. However, after prolonged training the network will start to recognize each individual example of input-output pair rather than settling for parameters that describe the mapping for all cases in general. This problem is called *overfitting* (or overtraining) [82].

When that happens the network gives exact answers for the training set, but is no longer able to respond correctly for input not contained in the training set. Much research is being done to find the optimal network architecture. An overall review is presented in [83] and a comprehensive bibliography can be found in [84].

The same problem arises for an adaptive fuzzy controller. The result of the learning depends on the number of membership functions and inference rules. When the number of inference rules is small, the inference rules cannot express the input-output relation for given data. On the contrary, when the number is large, the generalization capability of the inference rules is sacrified because of overfitting. Therefore, the number of inference rules has to be determined from a standpoint of overall learning capability and generalization capability.

Fig. 3.4 shows generalized relations between the number of inference rules and the training and checking errors. The larger the number of inference rules, the smaller the training error obtained. However, the checking error becomes larger for a larger number of inference rules after it exceeds a certain threshold value. Genetic algorithm can be used to overcome this problem as is described in the next section.

3.4.2 Algorithm to Optimize Inference Rules

By applying genetic algorithm the number of inference rules and the shape of membership functions can be determined. Fig. 3.5 shows the encoding of membership

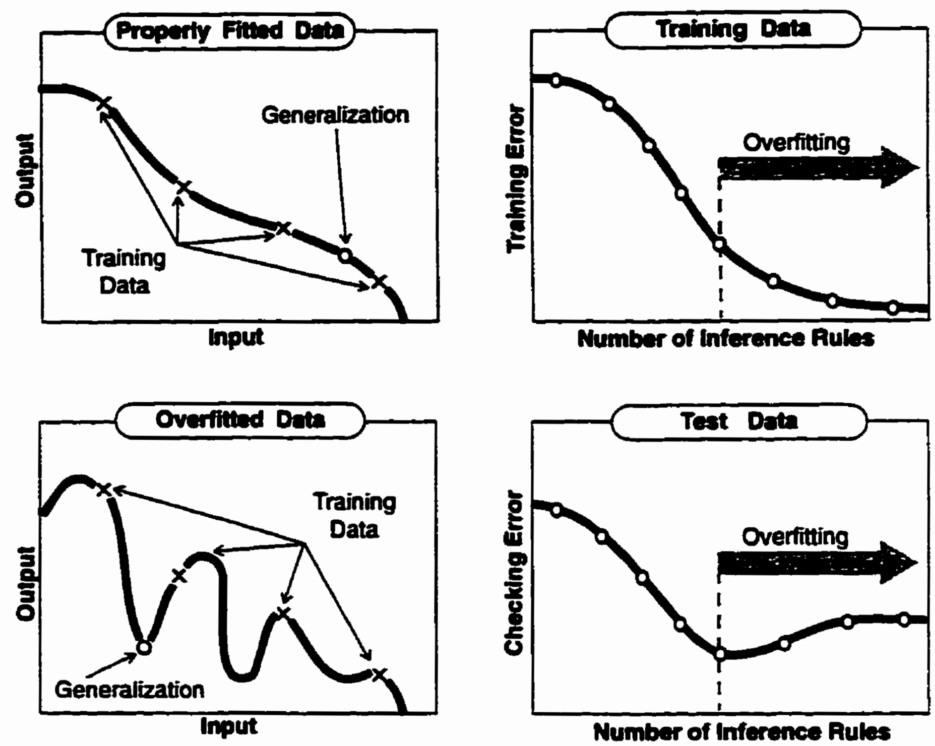

Figure 3.4: Effect of number of inference rules on training and checking error

functions to a bit string. The membership function takes triangular shape, and the width of each membership function is defined to be the length between the centers of neighboring membership functions.

The number and the shapes of membership functions can be expressed in term of strings consisting of "0' and "1", wherein the center position of each membership function is expressed by "1". By using this kind of encoding, the optimal number of membership functions and the center positions of these are searched for each input variable by the genetic algorithm. Since there is a direct relation between the number of membership functions and the number of inference rules, one can obtain the optimum number of inference rules from optimum number of membership

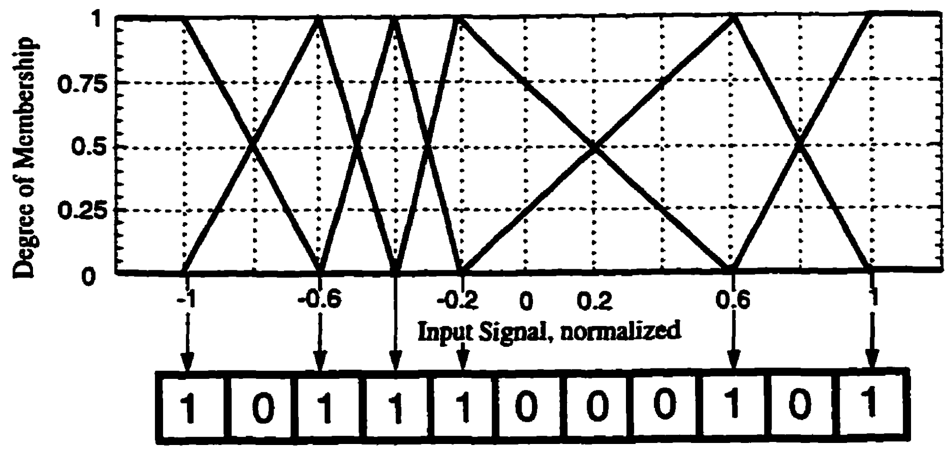

Figure 3.5: Encoding the membership functions to a bit-string

functions.

The four stages of genetic optimization are as follows:

1. **Initialization** – First, the domain of each input is divided into twelve sections, and a string with the length of 11 entries is associated with that interval. Each entry takes one of two possible values, 0 denoting the absence and 1 indicating the presence, of a triangular MF. To make sure that the MFs exist for both ends of the input domain, the first and the last entries are set to 1. The length between the centers of two neighboring MFs defines the width of each MF.

Given the strings for both input variables, the genotype (each individual in the population) is constructed by concatenating two strings, to yield a bit-string of length 22. The GA starts with a random population of such a bit-string, each string representing a network structure.

2. **Evaluation** – To qualify each individual, a fitness function is defined as:

$$F_i = \Phi_{max} - \Phi_i \tag{3.16}$$

where F_i is the fitness function for the ith individual, Φ_i defines the objective function which should be minimized for that individual, and Φ_{max} is the maximum objective function in the whole population. In this way, the best individual receives the maximum fitness.

The objective function is a combined objective function:

$$\Phi_i = k_1 N_i + k_2 E_{tri} + k_3 E_{chki} \tag{3.17}$$

where N_i is the number of adaptive nodes in the net-work, E_{tri} is the network error obtained from training data and E_{chki} is the network error as a result of checking data. The weighting parameters, k_1, k_2 and k_3 are mostly dependent on the problem and the desired solution. The checking error is included with the overall objective function in order to avoid problems with overfitting. If just the training error is used, then a network that has been overfitted might have a higher fitness than a network that cannot generalize well at all.

3. Selection – The individuals from the population are copied to a mating pool. Highly fit individuals are likely to be copied more than once. Unfit or poor performing individuals may be removed from the population. The behavior of the GA very much depends on how individuals are chosen to go into the mating pool. In this thesis, the fitness ranking technique is employed. Individuals are sorted in order of raw fitness, and then reproductive fitness values are assigned according to the rank of the individuals.

4. Genetic Operations – The selected individuals are recombined using crossover and mutation. During the crossover operation, two points in the strings are randomly chosen and the part enclosed by these two points is swapped. Mutation is accomplished by alteration of a single bit at a particular string position.

If the GA has been correctly implemented, the population will evolve over successive generations, so that the fitness of the individuals in each generation increases toward the global optimum. The population is said to have converged when 95% of the population shares the same value.

3.4.3 Combination with Self-learning Method

By employing both genetic algorithm and adaptive fuzzy controller, the inference rules' parameters can be tuned and also the number of membership functions can be optimized at the same time. This optimization contains two major process:
a) Search for the optimum number of rules and shape of MFs by using GA.
b) Train the network to determine the consequent parts of rule base by the gradient descent algorithm.

These processes and their interactions are shown in Fig. 3.6 and are described below:

- The cycle starts with a uniformly distributed random population of strings.
- Each string in the current population is decoded to an adaptive fuzzy network.
- Each network is trained to determine the consequent parts of fuzzy if-then rules.

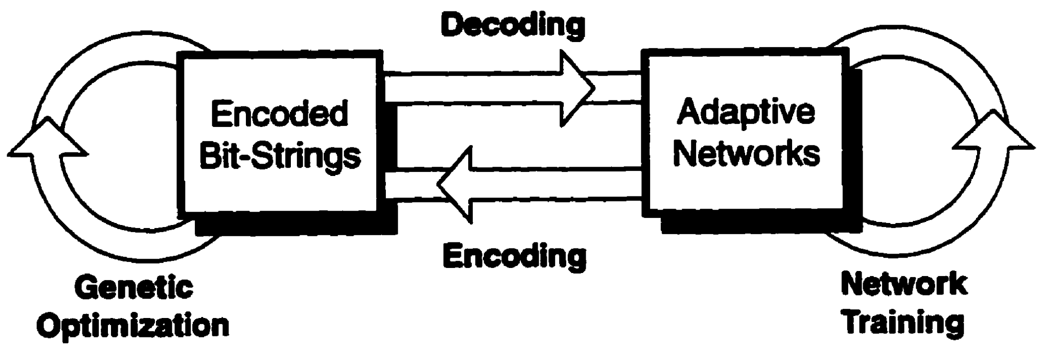

Figure 3.6: Training and optimization processes.

- After a certain number of epochs, training process is stopped and the total mean square error, E_{tri}, between the actual outputs and the desired outputs is calculated.

- Checking data is applied to each network and checking error, E_{chki}, is derived in the same fashion.

- The fitness for each individual is computed by using eqn. 3.16.

- With probability according to the fitness, a number of children are produced in the current generation.

- Genetic operations, crossover and mutation are applied to child individuals produced above and the new generation is formed.

3.5 Summary

In this Chapter, two major drawbacks of conventional fuzzy systems, the parameter tuning and finding the optimum structure, are discussed. Fuzzy system's parameters are in membership functions and inference rules. Adaptive fuzzy systems as a candidate to solve this problem is presented. Each step in the fuzzy system is implemented to a layer of a network containing the adjustable parameters of the fuzzy system. Then back-propagation algorithm is applied to the network and the parameters are tuned in such a way that the overall error of the network is minimized.

Using adaptive fuzzy system, one can copy any existing non-linear controller as a desired controller. However in general, the desired controller may not be available. A self-learning approach for training the adaptive fuzzy controller is presented. In this approach without using any desired controller, the error at the output of the plant is back-propagated through a plant identifier to obtain the error signal at the output of the controller. The adaptive fuzzy controller is trained using this error signal. To make the plant follow a desired trajectory, plant output is traced for m sampling period and compared with the desired trajectory. The derivative of error to a parameter for each sampling time is calculated and finally the changing rule is applied to the network.

The second problem, finding the optimum structure, can be overcomed by employing genetic algorithm. By encoding the membership functions to a bit-string, genetic algorithm starts from a random population of such a strings. Each string, representing an adaptive fuzzy controller, goes under the genetic operators such as cross-over and mutation. If the optimization process converges, it means the optimum solution has been found based on the fitness criteria. The fitness criteria consists of three elements: number of nodes in the network, final error due to the training data and final error due to the checking data. The last one assures that the adaptive fuzzy controller does not to become overfitted.

So far, the basics of FLC, ANNs and GAs (Chapter 2); and employing ANN and GA techniques to make Fuzzy Logic Controller more effective (Chapter 3) have been given. In the next part of dissertation, the potential of applying Fuzzy Logic Control with learning ability to power system will be discussed.

Part II

Simulation Studies

Chapter 4

Adaptive Fuzzy Logic Power System Stabilizer

4.1 Introduction

Studies have shown that the use of supplementary control signal in the excitation system and/or governor system of a generating unit can provide extra damping for the system and thus improve the unit's dynamic performance [8]. This method of stability improvement is cheap, flexible and easy to implement. A variety of PSS algorithms have been proposed and studied extensively in recent decades, among which some have been used successfully in the industry [85].

The most commonly used PSS, referred to as Conventional PSS (CPSS), is a fixed parameter analog-type device. The CPSS, first proposed in 1950's, is based on the linear model of the power system at some operating point to damp the low frequency power oscillations in the power system. This type of PSS is widely used in power systems and has made a great contribution in enhancing power system dynamic stability [17].

With the development of power systems and the increasing demand for quality electricity, it is worthwhile looking into the possibility of using modern control techniques. Power system's configuration keeps changing either due to switching actions in the short term or system enhancement in the long term. Therefore, it would be more suitable to use adaptive control techniques that can track the operating conditions and changes in the system. An adaptive PSS (APSS) can adjust its parameters on-line according to the environment in which it works and can provide good damping over a wide range of operating condition [26, 27].

The response time of the controller is the key factor to a good closed-loop performance. The APSS employs complicated algorithms for parameter identification and optimization which require significant amount of computing time. The higher the order of the discrete model of the controlled system used in identification, the more computing time is needed. To develop a quick response PSS, it is necessary to investigate alternative techniques.

In recent years, Fuzzy Logic Control (FLC) and Artificial Neural Network (ANN), as two branches of Artificial Intelligence (AI), have attracted considerable attention as candidates for novel control strategies because of the variety of advantages that they offer over the conventional computational systems. Unlike other classical control methods, FLC and ANN are model-free controllers, i.e. they do not require an exact mathematical model of the controlled system. Moreover, rapidity and robustness are the most profound and interesting properties in comparison to the classical schemes.

Designing power system stabilizers (PSSs) based on FLC has become an active area and satisfactory results have been obtained [40, 41]. Although, FLC introduces a good tool to deal with complicated, nonlinear and ill-defined systems, it suffers from a drawback - the "parameter tuning" for the controller. At present, there is no systematic procedure for the design of FLC. The most straightforward approach is to define Membership Functions (MFs) and decision rules subjectively by studying an operating system or an existing controller. Therefore, there is a need for an effective method for tuning the MFs and rules so as to minimize the output error or maximize the performance index.

Similarly, research on ANN application in power system stability has been reported [32, 33]. Besides the advantages mentioned above, ANN has the powerful capability of learning and adaptation, the advantages that can not be found in the FLC. However, one of the drawbacks of using conventional ANN is its "black-box" characteristic. It is difficult for an outside observer to understand or modify the network decision making process. For this reason initial values are chosen randomly.

In this Chapter, both the FLC and the ANN have been employed together to design a new PSS, Adaptive-Network based Fuzzy Logic PSS (ANF PSS). In this approach, a fuzzy PSS with learning ability has been constructed and is trained directly from the input and output data of the generating unit [75, 76]. Because the ANF has the property of learning, fuzzy rules and MFs of the controller can be tuned automatically by the learning algorithm. Learning is based on the error evaluated by comparing the output of the ANF controller and a desired controller. For studies in this Chapter, a self-optimizing pole-shifting APSS [28] has been chosen as the desired controller.

4.2 Adaptive-Network based FLC PSS

Essentially, an adaptive network is a superset of a multi-layer feedforward neural network with supervised learning capability. An adaptive network consists of nodes and directional links through which the nodes are connected. Each node performs a particular function which may vary from node to node. The choice of each node function depends on the overall input-output function which the adaptive network is required to perform. Whereas in an ANN, the adaptive parameters pertain to the links between the nodes, here the links only indicate the direction of flow of signals and part or all of the nodes contain the adaptive parameter(s). These parameters are specified by the learning algorithm and should be updated to achieve a desired input-output mapping. Similar to the ANN with supervised learning algorithm, the learning rule of adaptive network is based on gradient descent [30].

A class of adaptive networks which are functionally equivalent to FLC is referred to as Adaptive-Network based FLC. This scheme combines the idea of FLC and adaptive network structure and as a result an FLC network is constructed automatically by learning from the training examples itself.

In this study, an Adaptive-Network based FLC structure is employed to design a new fuzzy logic PSS (ANF PSS) for the system. The FLC is considered to have two inputs, the generator speed deviation $\Delta\omega$ and its derivative $\Delta\dot{\omega}$, and one control output, U_{pss}. Moreover, the rule base contains the fuzzy if-then rules of Takagi and Sugeno's type [36], in which the output of each rule is a linear combination of input variables plus a constant term:

$$\text{If } \Delta\omega \text{ is } A_i \text{ and } \Delta\dot{\omega} \text{ is } B_i \text{ then } U_{pss} = p_i\Delta\omega + q_i\Delta\dot{\omega} + r_i$$

and the final output is the weighted average of each rule's output. The architecture of the ANF PSS is shown in Fig. 4.1 , where node functions in each layer are as described below:

Layer 1 − Each node in this layer performs a MF:

$$y_i^1 = \mu_{Ai}(x_i) = exp\left\{-\left[\left(\frac{x_i - c_i}{a_i}\right)^2\right]^{b_i}\right\} \tag{4.1}$$

where x_i is the input of node i, A_i is the linguistic label associated with this node and $\{a_i, b_i, c_i\}$ is the parameter set of the bell-shaped MF. y_i^1 specifies the degree to

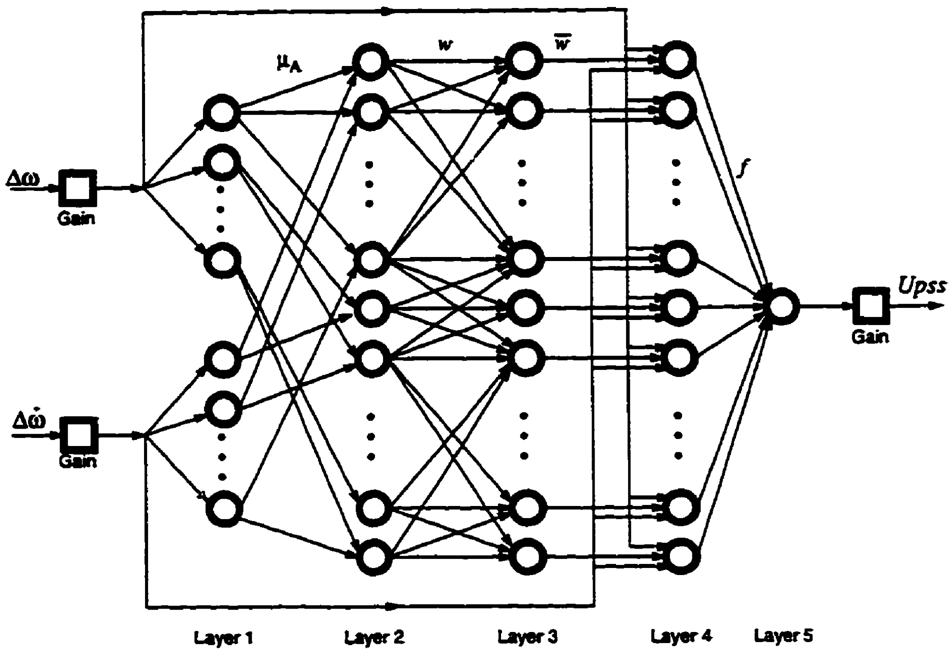

Figure 4.1: Architecture of ANF PSS.

which the given input belongs to the linguistic label A_i, with maximum equal to 1 and minimum equal to 0.

Layer 2 – Every node in this layer represents the firing strength of the rule. Hence, the nodes perform the fuzzy AND operation:

$$y_i^2 = w_i = min(\mu_{Ai}(\Delta\omega), \mu_{Bi}(\Delta\dot{\omega})). \tag{4.2}$$

Layer 3 – The nodes of this layer calculate the normalized firing strength of each rule:

$$y_i^3 = \bar{w}_i = \frac{w_i}{\sum_{i=1}^{n} w_i}. \tag{4.3}$$

Layer 4 – Output of each node in this layer is the weighted consequent part of the rule table:

$$y_i^4 = f_i = \bar{w}_i(p_i\Delta\omega + q_i\Delta\dot{\omega} + r_i) \tag{4.4}$$

where $\bar{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set.

Layer 5 – The single node in this layer computes the overall output as the summation of all incoming signals:

$$y_i^5 = \sum_{i=1}^{n} f_i \tag{4.5}$$

Thus an adaptive network has been constructed which is functionally equivalent to a fuzzy logic PSS. This structure can update MF and rule base parameters according to the gradient descent update procedure.

4.3 ANF PSS Training

In a conventional FLC, parameters (MFs and rules) are specified by an expert who is familiar with the system. In the ANF based PSS, however, it is assumed that no expert is available; the initial values of MFs parameters are equally distributed along the universe of discourse and all consequent parts of rule table are set to zero. In this manner, the ANF PSS starts from zero output and during the training process it gradually learns to function as close to the desired controller as possible. However, in practice, a priori knowledge in the form of the untuned fuzzy if-then rules is employed. As a result, the training starts from much less error.

In the studies presented in the next section, the ANF PSS was trained by the self-optimizing pole shifting APSS [28] as the desired controller. The sampling period, T_s, for APSS is set to $50ms$ and no computation time is assumed. The training was performed over a wide range of conditions for the generating unit, i.e., the generator output ranging from 0.1 $p.u.$ to 1.0 $p.u.$, and the power factor ranging from 0.7 lead to 0.1 lag. Similarly, a wide spectrum of possible disturbances was used for the training. These disturbances are: reference voltage and infinite bus voltage disturbances in the range of -0.05 $p.u.$ to 0.05 $p.u.$, governor input torque variations from -0.3 $p.u.$ to 0.3 $p.u.$, one transmission line outage, and three phase fault on one line of the double circuit transmission line connected to the generating unit. A total of 18000 input-output data pairs were obtained for the training of ANF PSS.

The number of MFs for each input variable is determined by the complexity of the training data and by trial and error. It is similar to choosing the number of neurons in the hidden layers of an ANN; too many neurons will result in wasting the computer memory and computing time and too few neurons will not give the appropriate control effect. Based on earlier experience, seven linguistic variables for each input variable were used to get the desired performance. The MFs for two inputs, $\Delta\omega$ and $\Delta\dot{\omega}$, before and after training are shown in Fig. 4.2. As Fig. 4.2 shows, the universe of discourse for both input variables is normalized and the gain parameters are chosen based on the input-output space:

$$\Delta\omega \text{ gain} = 1.2 \ , \ \Delta\dot{\omega} \text{ gain} = 0.1 \ , \ U_{pss} \text{ gain} = 0.1$$

4.4 System Configuration and Model

A number of studies have been performed to investigate the effect of the proposed stabilizer and the results are compared with those of the CPSS and ANF PSS. In all the following tests, CPSS is chosen to be an analog type PID controller and ANF PSS is considered to be a digital controller with $T_s = 50\ ms$.

This study is based on a detailed model of a generating unit connected to a constant voltage bus through two parallel transmission lines. A schematic diagram

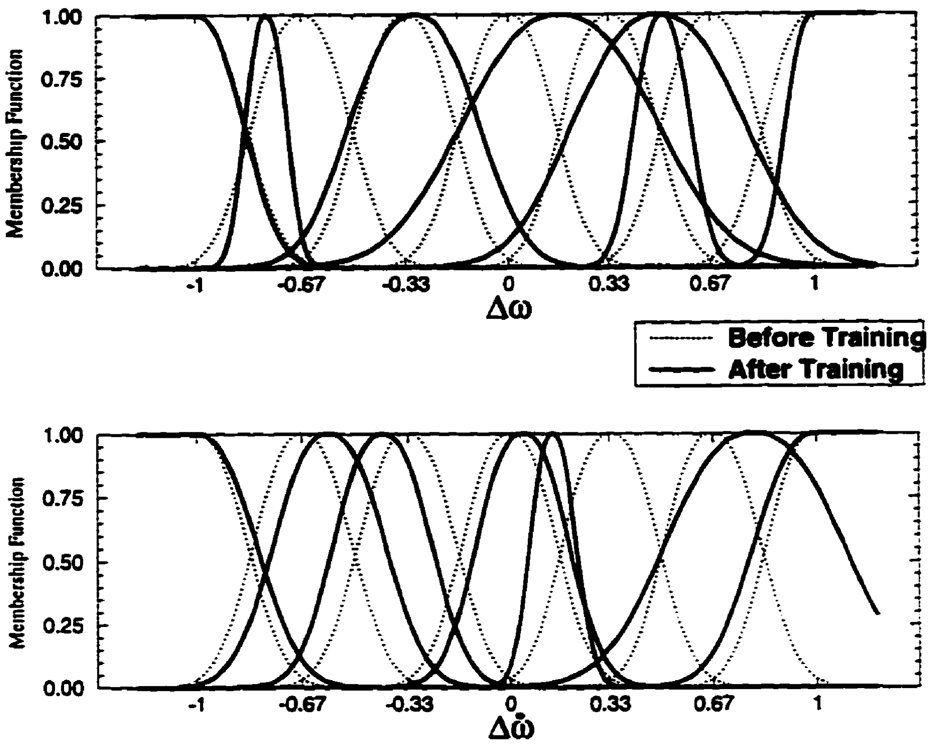

Figure 4.2: Membership functions before and after training.

of the system is shown in Fig. 4.3. For comparison the CPSS was also included in the studies. A switch is used to achieve the changes between the stabilizers. The differential equations used to simulate the generating unit, the transfer functions of the governor, AVR and CPSS, and also the system parameters are given in Appendix A.

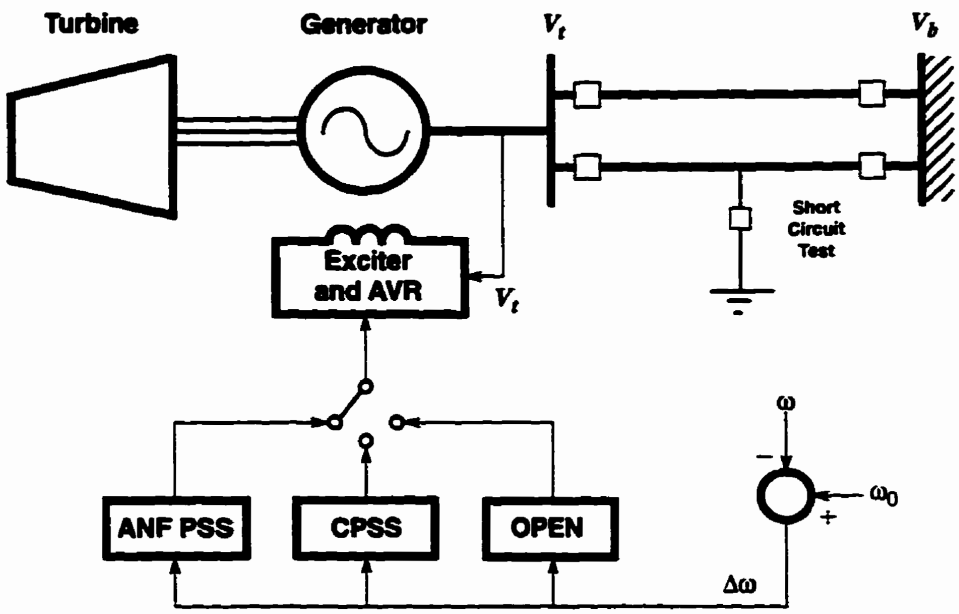

Figure 4.3: Basic system model configuration.

4.5 Test Results

4.5.1 CPSS Parameter Tuning

With the generator operating at a power of 0.7 *p.u.*, 0.85 power factor lag, a 0.05 *p.u.* step in input torque reference is applied at time 1 *s*. At time 5 *s*, the change in torque reference is removed and the system returns to its previous operating condition.

Under these conditions, the CPSS with the transfer function given in Appendix A was carefully tuned for the best performance, i.e. the overshoot and settling time were minimized by the CPSS damping effect. The parameters of the CPSS were then kept unchanged for all the tests described in this Chapter.

Results of the study with the ANF PSS, CPSS and without a stabilizer are shown in Fig. 4.4. It can be seen from the figure that the ANF PSS damps out the low frequency oscillations very quickly.

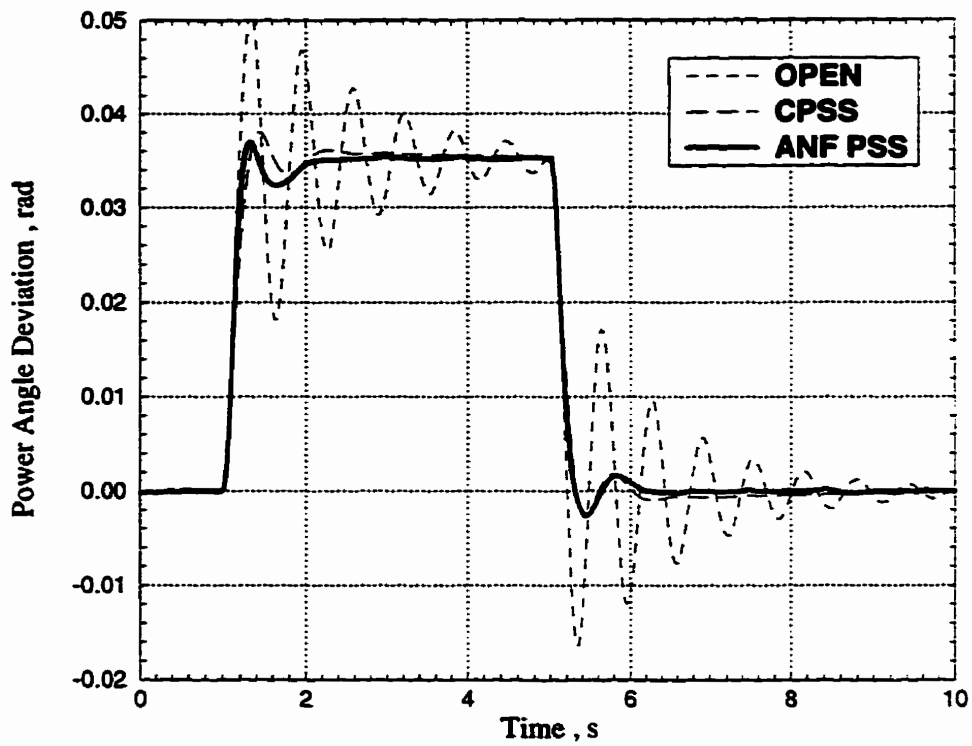

Figure 4.4: Response to a 0.05 *p.u.* step increase in torque and return to initial conditions.

4.5.2 Light Load

The system condition is the same as the previous case except that the generator is now operating under a light load condition: the power is 0.30 *p.u.* with 0.85 power factor lag, and the disturbance is a 0.15 *p.u.* step increase in input torque reference. The disturbance is large enough to cause the system to operate in a nonlinear region.

System response for these non-linear conditions is shown in Fig. 4.5 for both the CPSS and ANF PSS. Because the CPSS is designed for best performance at another operating condition, it is not able to provide as effective a damping at this operating condition. However, the ANF PSS still provides very effective performance.

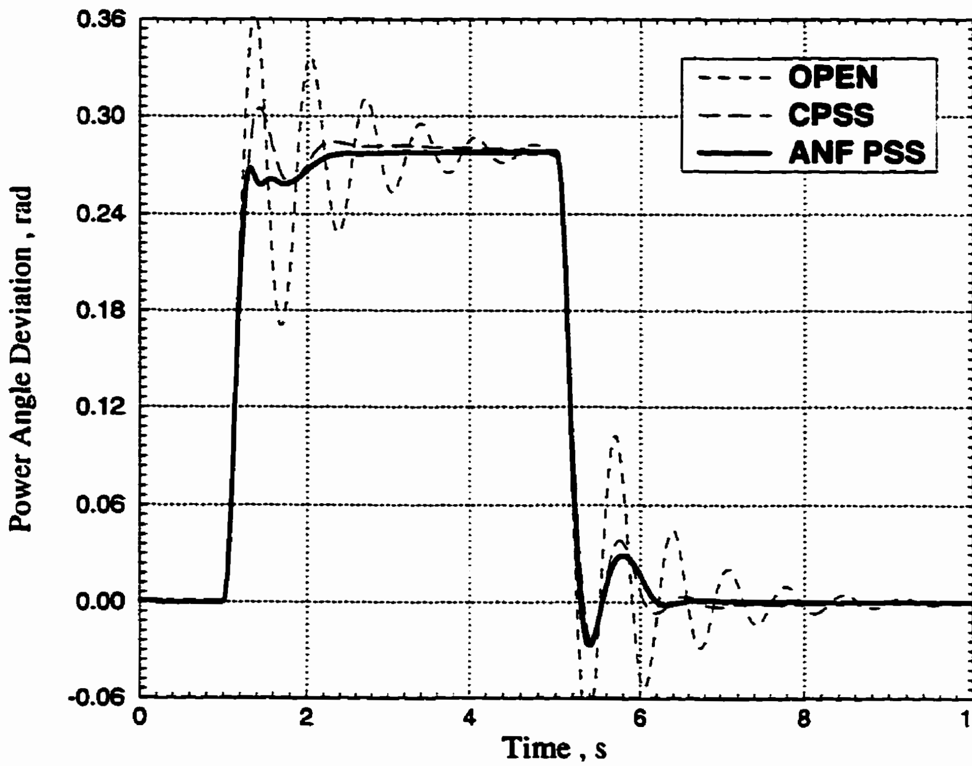

Figure 4.5: Response to a 0.30 *p.u.* step increase in torque and return to initial condition in light load test.

4.5.3 Leading Power Factor Test

When the generator is operating at a leading power factor, it is a difficult situation for the controller because the stability margin is reduced. However, in order to absorb the capacitive charging current in a high voltage power system, it may become necessary to operate the generator at a leading power factor. It is, therefore, desirable that the controller be able to guarantee stable operation of the generator under leading power factor condition.

With the generator operating at a power of 0.3 *p.u.* with 0.9 *p.f.* lead, a 0.2 *p.u.* step increase in torque reference was applied. The results given in Fig. 4.6 show that the oscillation of the system is damped out rapidly and demonstrates the effectiveness of the ANF PSS to control generator under leading power factor operating conditions.

4.5.4 Voltage Reference Change

With the generator operating at 0.2 *p.u.* active power, 0.8 *p.f.* lag and 1.04 *p.u.* terminal voltage, a 5% step decrease in reference voltage was applied at 1 *s* and removed at 5 *s*. The generating unit power angle deviation responses are shown in Fig. 4.7. In the open loop system without any PSS, the severity of the oscillations increases as the reference voltage drops, since the system stability margin decreases as the reference voltage drops for a certain active power output. It can be seen from Fig. 4.7 that the oscillations are effectively damped by ANF PSS for both reference voltage decrease and increase, which means that the system stability margin is enhanced by using ANF PSS.

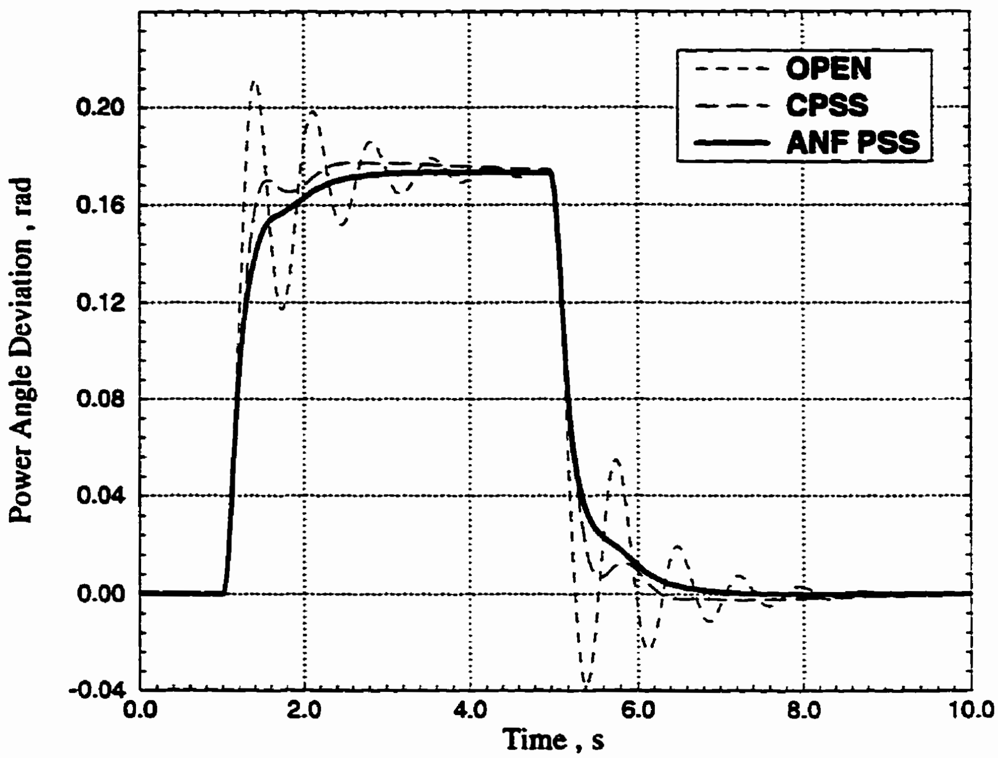

Figure 4.6: Response to a 0.2 $p.u.$ step increase in torque under leading power factor conditions.

4.5.5 Fault Test

The behavior of the proposed ANF PSS under transient conditions was further verified by applying a fault. For this study, the equivalent reactance of the double circuit transmission line was set at 0.4 $p.u.$ instead of 0.6 $p.u.$. The response of the power system to a three phase to ground short circuit at the middle of one transmission line, cleared 200 ms later by the disconnection of the faulted line and successful reclosure after 4 s is shown in Fig. 4.8 . The results show that ANF PSS minimizes

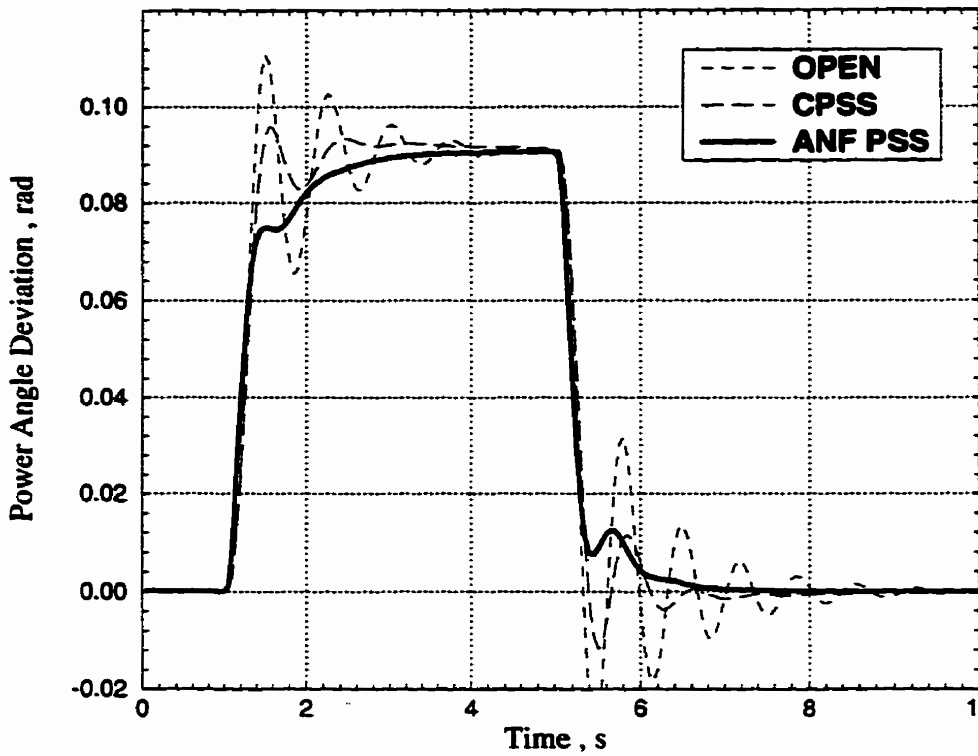

Figure 4.7: Response to a 0.05 *p.u.* step decrease in reference voltage and return to initial condition.

the deviation of the power angle of the generator after the fault and helps the system to reach the new operating point very quickly.

4.5.6 Stability Margin

Besides the improvement of the dynamic performance by introducing the supplementary controller, the stability margin has also been increased. To demonstrate this effect, a simulation study was conducted with the initial operating conditions of

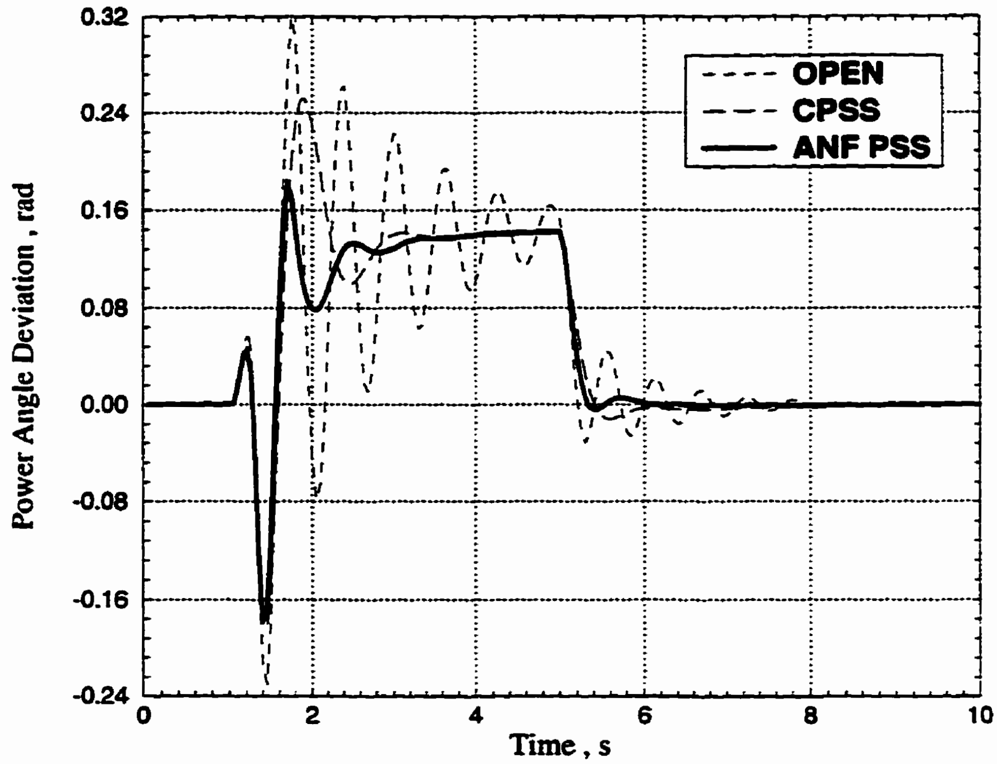

Figure 4.8: Response to a three phase to ground fault at the middle of one transmission line.

0.95 *p.u.* , 0.9 *p.f.* lag, and the input torque reference was increased gradually. The dynamic stability margin is described by the maximum power output at which the system losses synchronism. The results for the system without stabilizer, with CPSS and with ANF PSS are given in Table 4.1. ANF PSS provides the largest output power, which indicates that the dynamic stability margin of the system is improved most by the ANF PSS.

4.6 Comparison of ANF PSS and APSS

The purpose of training in this Chapter is to make the ANF PSS function as close to the APSS as possible. In Fig. 4.9, typical comparison curves of the closed-loop system response with ANF PSS and APSS are given. The generator operating point and the applied disturbance are the same as Test 1; i.e. $P = 0.7\ p.u.$, $p.f. = 0.85\ lag$ and a 0.05 $p.u.$ step in input torque reference is applied. In this figure, APSS-1 is the system response using an adaptive PSS with no computation time and APSS-2 is the system response using the same APSS, but with an assumed computation time of 20 ms for the control signal. Although in general, there is not much difference between the performance of the ANF-PSS and APSS-1, some differences can be seen between ANF PSS and APSS-2 performances.

If the computation of control signal in APSS takes more time (such as in the case of MIMO systems), the difference between these two stabilizers will become even more significant.

4.7 Comparison of training time with ANN

As mentioned in Section 3.2.2, by incorporating the knowledge base as initial parameters of ANF PSS, the training time can be reduced drastically. Artificial neural

	OPEN	CPSS	ANF PSS
Maximum Power	1.95 $p.u.$	2.85 $p.u.$	3.30 $p.u.$
Maximum Rotor Angle	1.18 $rad.$	2.10 $rad.$	2.45 $rad.$

Table 4.1: Dynamic stability margin results.

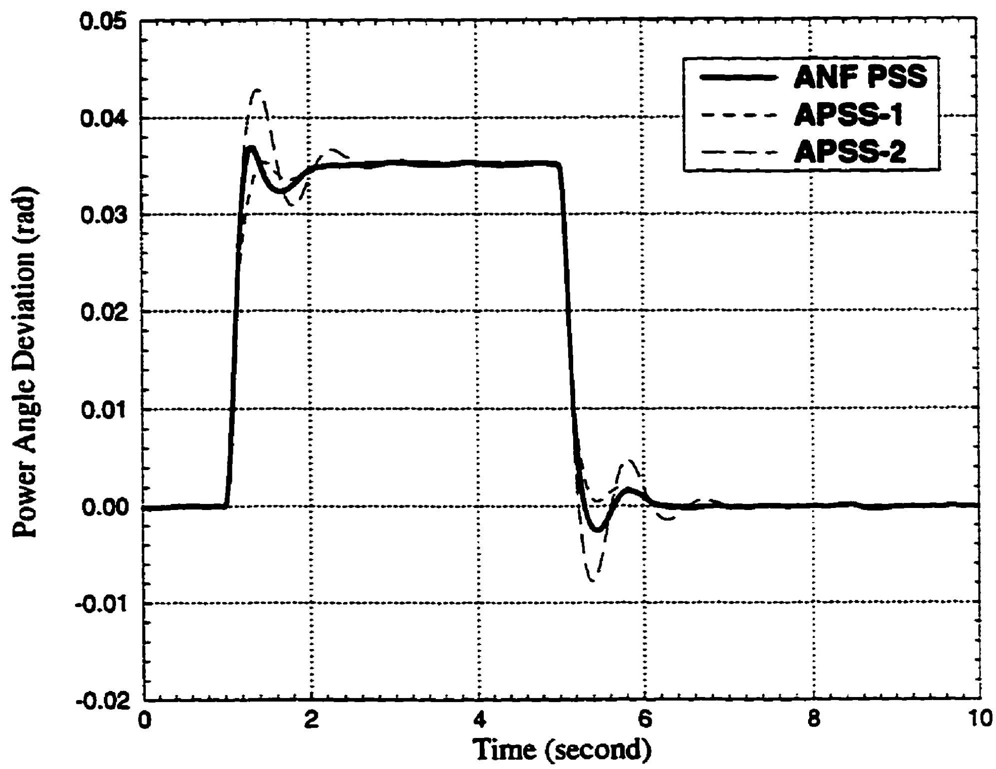

Figure 4.9: Comparison of ANF PSS and APSS (same conditions as in Fig. 4.4).

networks suffer from long traing time since their training starts with small random initial parameters. To demonstrate this property, an ANN with two hidden layers and 40 neurons in the first layer and 20 neurons in the second layer was chosen for comparison with the ANF PSS described in Section 4.2. Both networks have the same number of inputs and outputs. Moreover, exactly the same training data obtained from the APSS is given to both networks.

Before training starts, the ANF PSS contains only 13 major rules. These rules are located on the center row and center column of fuzzy rule table as shown in Fig. 4.10. This figure shows the values of c_i parameter in the consequent part of Sugeno's inference rules. The other parameters, a_i and b_i, are set to zero. Although all 49 rules can be assigned the corresponding values, it is assumed that the only available knowledge about the input-output relation of the stablizer is when one of the input signals is zero and the other one varies. As shown in Fig. 4.10, the consequent part of the inference rule table is equally distributed from NB to PB.

$\Delta\dot{\omega}$

$\Delta\omega$	NB	NM	NS	ZO	PS	PM	PB
NB				NB			
NM				NM			
NS				NS			
ZO	NB	NM	NS	ZO	PS	PM	PB
PS				PS			
PM				PM			
PB				PB			

Figure 4.10: Initial fuzzy rule table of parameter c_i before training

Fig. 4.11 shows the sum of squared error curves for both the ANN PSS and the ANF PSS. The following can be concluded from this figure:

- Initial error for ANF PSS starts from much smaller value, since the stabilizer has a rough a priori knowledge at the beginning.

- After about 50 epochs, ANF PSS training has converged. However, in the

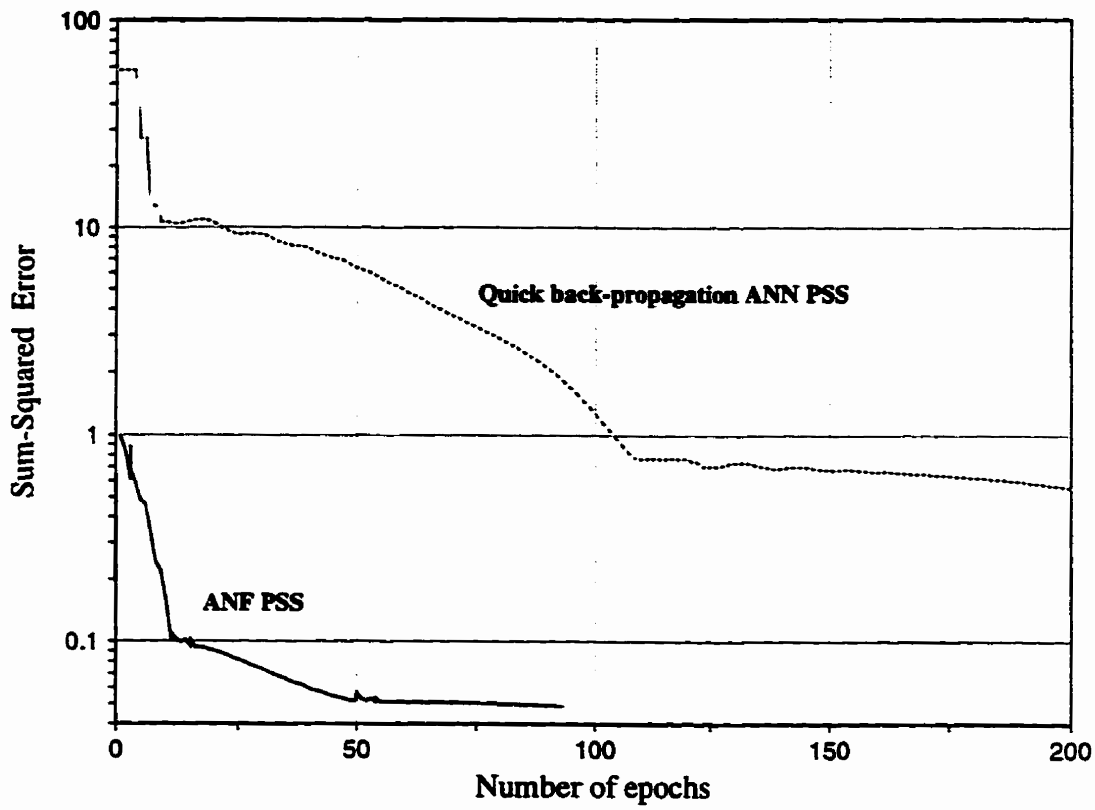

Figure 4.11: Comparison of ANF PSS and APSS (same conditions as in Fig. 4.4).

ANN PSS case, even after 150 epochs, the network error is still reducing.

•• Obviously, final error is lower for ANF PSS than that of ANN PSS.

4.8 Summary

A new design method for power system stabilizer employing adaptive-network-based fuzzy logic controller and its application to a power system are described in this Chapter. The proposed PSS employs a multi-layer adaptive network. The network is trained directly from the input and the output of the generating unit. The algorithm combines the advantages of the Artificial Neural Networks and Fuzzy Logic Control schemes.

The following conclusions can be drawn from the results.

- The proposed method retains all the advantages of artificial neural networks and fuzzy logic controller, such as simplicity, adaptability, rapidity and robustness.

- In this method, by using neural network as a structure for the fuzzy logic controller, the design time of conventional FLC can be significantly reduced; fuzzy rules and membership functions are generated automatically to meet the prespecified performance; i.e. the tuning problem has been eliminated.

- Compared to a conventional neural network, the training time is dramatically decreased, since a priori knowledge in the form of fuzzy if-then rules can be employed. It means that the initial parameters of the adaptive network can be chosen in such a way that the training of the network starts from a much less error at the output of the network than that of a neural network with random initial parameters. Also, the parameters of the proposed controller have physical interpretations unlike the "black-box" characteristic of the neural network.

- Test results for various conditions show that the proposed stabilizer is able to function as close to the adaptive PSS as possible. However, the longer computation time is one of the major limitations of the adaptive control strategy.

- Simulation results show that the ANF PSS can provides good damping over a wide operating range and can significantly improves the dynamic performance of the system.

Chapter 5

A Self-Learning Fuzzy Logic Power System Stabilizer

5.1 Introduction

In Chapter 4 [86][87], both the FLC and the ANN have been employed together to design a new PSS, Adaptive-Network-Based Fuzzy Logic PSS (ANF PSS). In this approach, a fuzzy PSS with learning ability has been constructed and is trained directly from the input and output data of the generating unit [75, 76]. Because the ANF has the property of learning, fuzzy rules and MFs of the controller can be tuned automatically by the learning algorithm. Learning is based on the error evaluated by comparing the output of the ANF controller and a desired controller which in this case has been chosen as a self-optimizing pole-shifting Adaptive PSS (APSS) [28].

The ANF PSS presented in this Chapter is based on a self-learning FLC [77]. In other word, without resorting to another existing controller, it is proposed to construct an FLC that performs a prescribed task. Similar to the first approach, the learning method is basically a special form of the gradient descent (back propagation), which is used for the training of ANN. To train the controller, the back-propagation method is employed to propagate the plant output error signal through different stages in time [81].

5.2 Self-Learning ANF PSS

In Chapter 4, the ANF PSS was trained by the self-optimizing pole shifting APSS [26] as the desired controller. However, in a typical situation, the desired controller may not be available. The ANF PSS presented in this Chapter is trained from the performance of the generating unit output which is the generator speed deviation.

5.2.1 Structure of ANF PSS

In this approach, before finding a controller to control the plant states, a function approximator (or model) is needed to represent the input-output behavior of the plant. To model the plant, an adaptive-network-based fuzzy logic model is employed, which has the same structure as the controller. The utility of this plant model is that it can compute the derivative of the model's output with respect to its input by means of the back propagation process. Consequently, propagating errors between actual and desired plant outputs back through the model produces the error in the control signal, which can be used to train the controller. The block diagram of Fig. 5.1 shows an adaptive network containing two subnetworks, the fuzzy controller and the plant model.

The training process for the controller starts from an initial state at $t = 0$. Then the FLC and the plant model generate the next states of U_{pss} and $\Delta\omega$ at time $t = h$. The process continues till the plant state trajectory is determined. The objective of the learning process is the minimization of:

$$E = \sum_{k=1}^{m} [\Delta\omega(h * k) - \Delta\omega_d(h * k))]^2 + \lambda * \sum_{k=0}^{m-1} U_{pss}(h * k)^2 \tag{5.1}$$

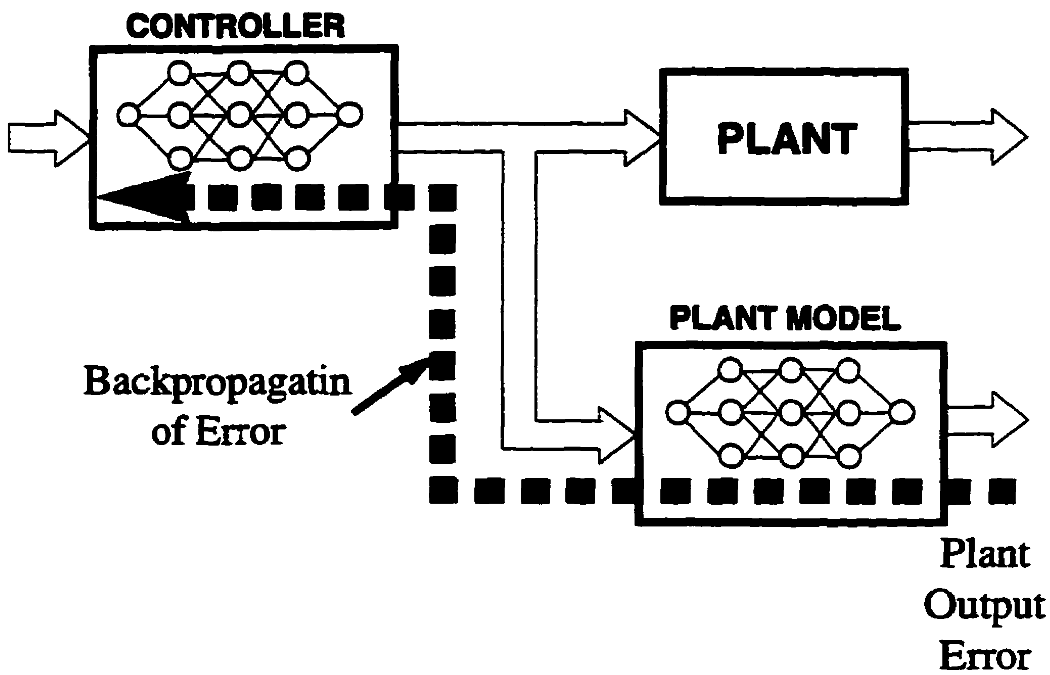

Figure 5.1: Error back-propagation through plant model

where $\Delta\omega_d$ is the desired output trajectory, which is always zero and the tuning parameter λ is selected to improve the plant trajectory.

5.3 Training of ANF PSS

The training was performed over a wide range of conditions for both the controller and the plant model with the generator output ranging from 0.1 $p.u.$ to 1.0 $p.u.$, and the power factor ranging from 0.7 lead to 0.1 lag. Similarly, a wide spectrum of possible disturbances was used to obtain the training data. These disturbances are: reference voltage and infinite bus voltage disturbances in the range of -0.05 $p.u.$ to 0.05 $p.u.$, governor input torque variations from -0.3 $p.u.$ to 0.3 $p.u.$, one transmission line outage, and three phase fault on one circuit of the double circuit transmission

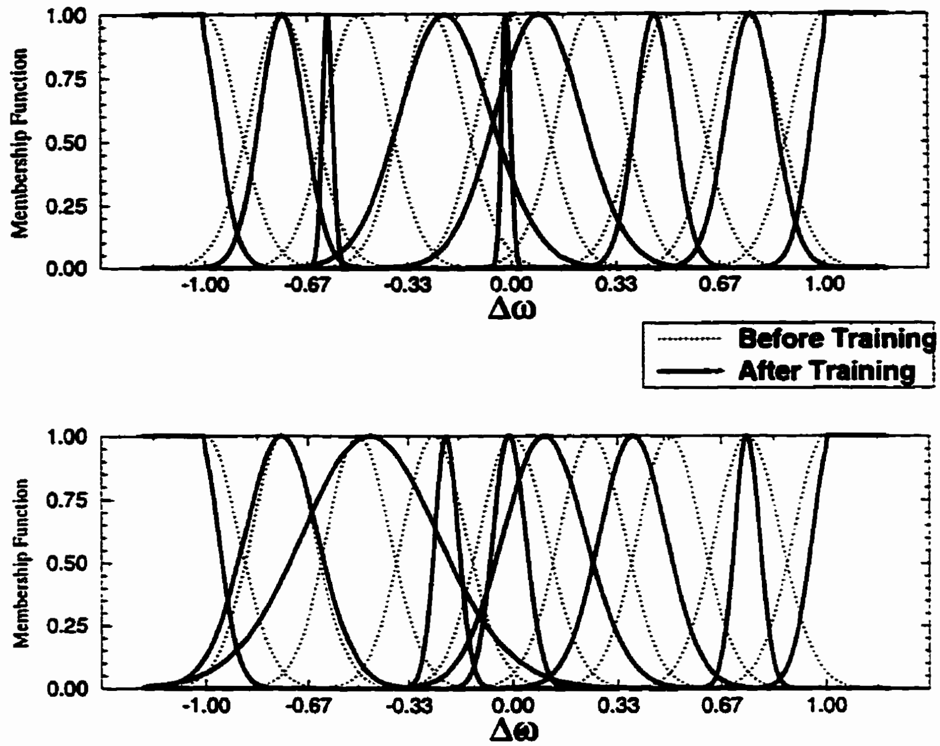

Figure 5.2: Membership functions before and after training

line.

The MFs for two inputs of the controller, $\Delta\omega$ and $\Delta\dot{\omega}$, before and after training are shown in Fig. 5.2.

5.3.1 System Configuration and Model

A number of studies have been performed to investigate the effect of the proposed stabilizer and the results are compared with those of the CPSS and ANF PSS.

This study is based on a detailed 7th order model of a generating unit connected to a constant voltage bus through two parallel transmission lines. A schematic diagram of the system is shown in Fig. 4.3 For comparison the CPSS was also included in the studies. A switch is used to achieve the changes between the stabilizers. The differential equations used to simulate the generating unit, the transfer functions of the governor, AVR and CPSS are given in the Appendix A.

5.4 Test Results

5.4.1 Tuning the parameter λ

With the generator operating at 0.7 $p.u.$ power, 0.85 $p.f.$ lag, a 0.15 $p.u.$ step in input torque reference is applied at time 1 s. At time 5 s, the change in torque reference is removed and the system returns to its previous operating condition. Under these conditions, the performance of the self-learning ANF PSS was investigated. The results are shown in Fig. 5.3 for different values of λ, and for the subsequent studies $\lambda = 1$ is used.

5.4.2 Light Load

With the system now operating under a light load condition of 0.20 $p.u.$ power, 0.85 $p.f.$ lag, a 0.30 $p.u.$ step increase in input torque reference is applied. The disturbance is large enough to cause the system to operate in a nonlinear region. System response for these conditions is shown in Fig. 5.4 for both the CPSS and ANF PSS. Because the CPSS is designed for best performance at another operating condition, it is not able to provide as effective a damping at this operating condition. However, the ANF PSS still provides very effective performance.

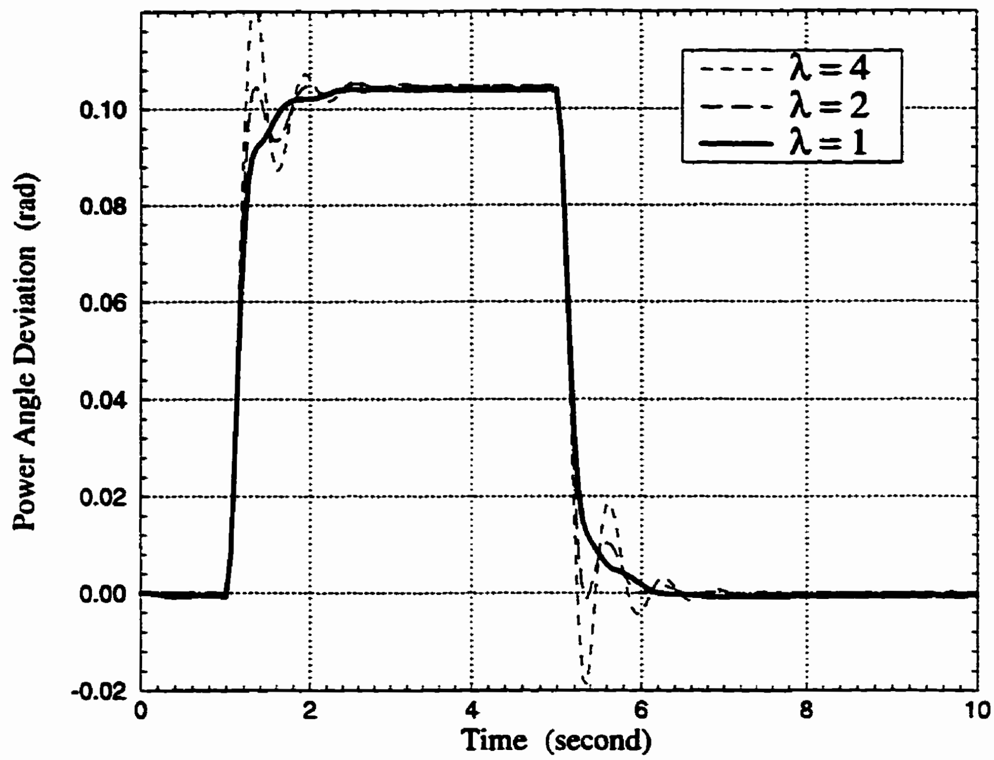

Figure 5.3: Response to a 0.15 $p.u.$ step increase in torque with different values of λ

5.4.3 Leading Power Factor Test

When the generator is operating at a leading power factor, it is a difficult situation for the controller because the stability margin is reduced. However, the controller must guarantee stable operation under these conditions also.

With the generator operating at 0.3 $p.u.$ power, 0.9 $p.f.$ lead, a 0.20 $p.u.$ step increase in torque reference was applied. The results given in Fig. 5.5 show that the oscillation of the system is damped out rapidly. It demonstrates the effectiveness of

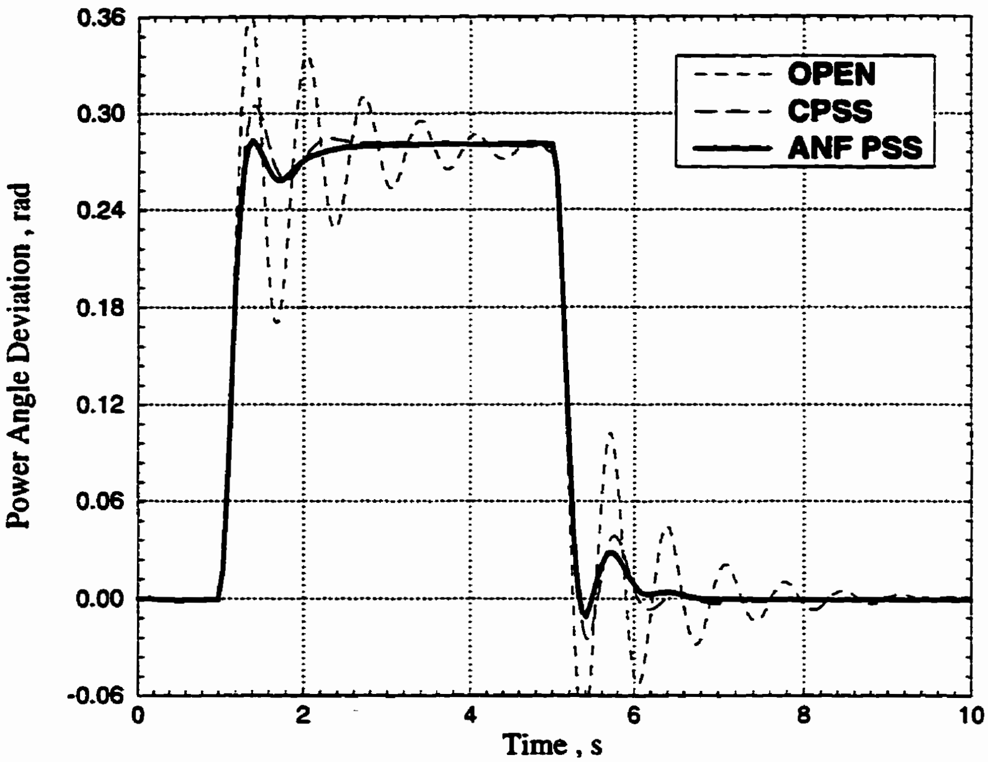

Figure 5.4: Response to a 0.30 *p.u.* step increase in torque and return to initial condition in light load test

the ANF PSS to control generator under leading power factor operating conditions.

5.4.4 Fault Test

The behavior of the proposed ANF PSS under transient conditions was further verified by applying a fault. The response of the power system to a three phase to ground short circuit at the middle of one transmission line, cleared 200 *ms* later by the disconnection of the faulted line and successful reclosure after 4 *s* is shown in

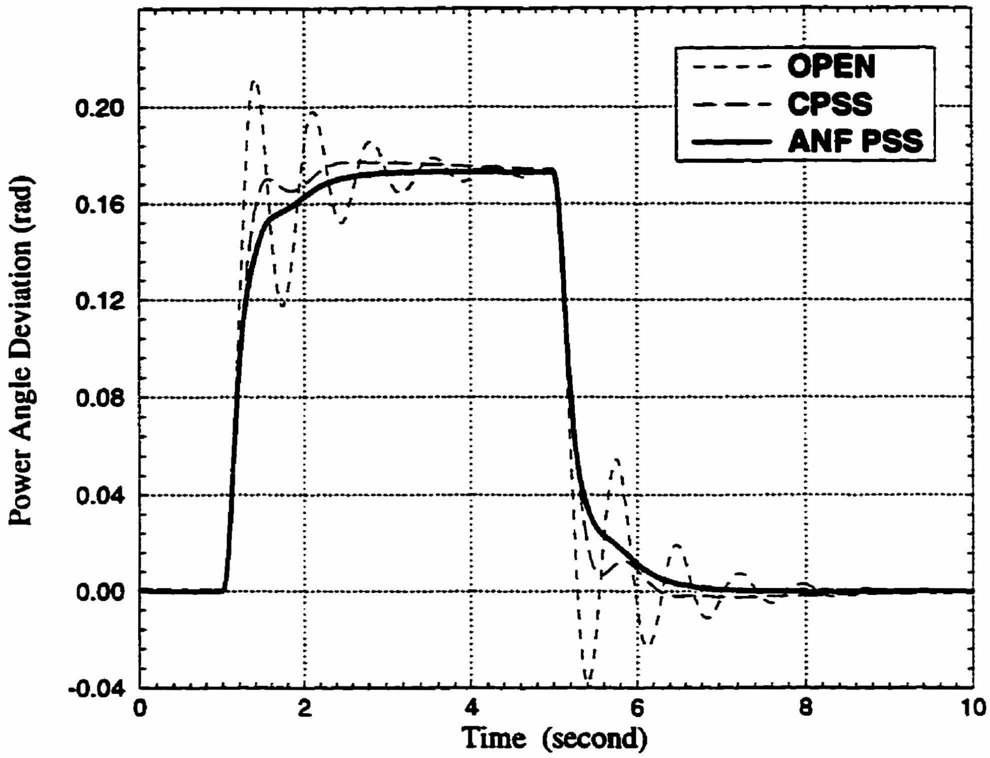

Figure 5.5: Response to a 0.20 *p.u.* step increase in torque under leading power factor condition

Fig. 5.6 . The results show that ANF PSS minimizes the deviation of the power angle of the generator after the fault and helps the system to reach the new operating point very quickly.

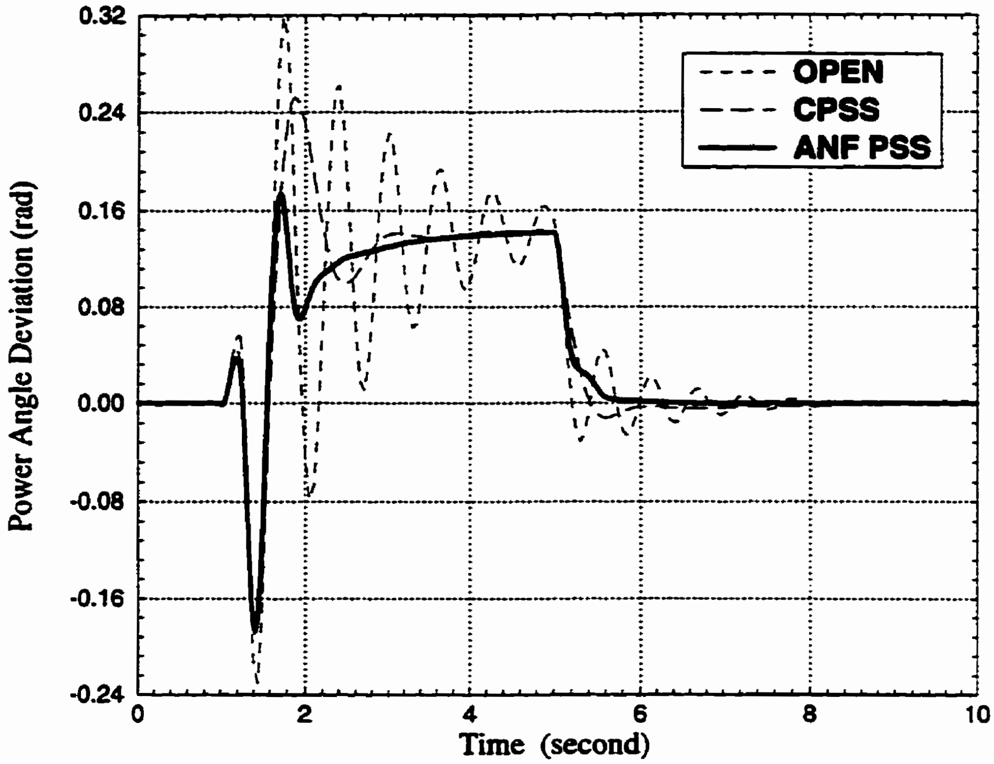

Figure 5.6: Response to a three phase to ground fault at the middle of transmission line.

5.5 Summary

A new design method for power system stabilizer employing adaptive-network-based fuzzy logic controller with self-learning capability and its application to a power system are presented in this Chapter. The proposed method retains all advantages of artificial neural network and fuzzy logic controller, such as simplicity, adaptability, rapidity and robustness. Compared to a conventional neural network, the training time is dramatically decreased , since a prior knowledge in the form of fuzzy if-then rules can be employed. The ANF PSS presented in this Chapter is trained directly from the performance of the generating unit and thus was independent of other PSS. It provides good damping over a wide range and significantly improves the dynamic performance of the system.

Chapter 6

Genetically Optimized Fuzzy Logic Power System Stabilizer

6.1 Introduction

In Chapter 4, use of an Artificial Neural Network (ANN) to design an Adaptive-Network-Based Fuzzy Logic PSS (ANF PSS) is described [87], [88]. Because the ANF has the property of learning, fuzzy rules and MFs of the controller can be tuned automatically by the learning algorithm. However, the selection of the number of inference rules in these methods is not a trivial task. Finding the optimum number of rules for a specific application is, to a large extent, a process of trial and error, relying mostly on past experience with similar application. Also, the size of the adaptive network grows exponentially with the increasing number of MFs, requiring more training time. This problem becomes more crucial when the number of input variables increases.

In order to solve this problem, the Genetic Algorithm (GA) [69], as a global optimization technique, is employed to construct an ANF PSS with optimum structure. Since the number of rules depends, in a direct manner, on the number of MFs, the number and shape of MFs are determined first by applying the GA [78]. Then the parameters in the consequent part of the rule table are specified by the learning algorithm which is a special form of the gradient descent (back propagation).

6.2 Genetically Optimized ANF PSS

Although the adaptive-network-based FLC can solve the problem of tuning MFs and inference rules, the selection of the number of rules is still tedious trial and error work. Two trivial algorithms, constructive and destructive, are usually employed. Both methods, however, are guided by a predefined heuristic, as it is computationally expensive to try out all possible networks.

The more powerful technique for efficiently searching the space of all possible networks is the genetic algorithm. By encoding the center of MFs to a bit string as shown in Fig. 6.1, the shape and number of MFs can be optimized by means of the GA [69]. Since the number of rules is proportional to the number of MF's for each input, the optimum number of rules (and eventually optimum network structure) will be achieved.

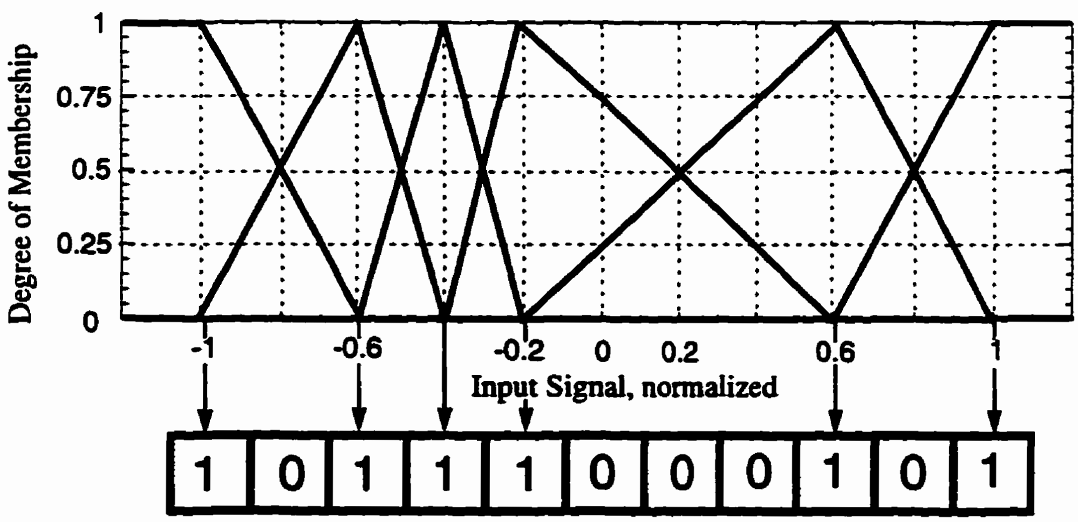

Figure 6.1: Encoding the MFs to a bit-string.

The four stages involved in the genetic search process are described in Chapter

3. In the initialization step, the domain of both input variables, $\Delta\omega$ and $\Delta\dot{\omega}$, are divided into twelve sections. A bit string, containing 0 or 1, is associated with each section.

The fitness function is defined to be:

$$F_i = \Phi_{max} - \Phi_i \tag{6.1}$$

where F_i is the fitness function for the ith individual, Φ_i defines the objective function which should be minimized for that individual, and Φ_{max} is the maximum objective function in the whole population. In this way, the best individual receives the maximum fitness.

The objective function is a combined objective function:

$$\Phi_i = k_1 N_i + k_2 E_{tri} + k_3 E_{chki} \tag{6.2}$$

where N_i is the number of adaptive nodes in the network, E_{tri} is the network error obtained from training data and E_{chki} is the network error as a result of checking data. The weighting parameters, k_1, k_2 and k_3 are mostly dependent on the problem and the desired solution. Their values are chosen as:

$$k_1 = 0.02, \; k_2 = 1, \; k_3 = 20.$$

The checking error is included with the overall objective function in order to avoid problems with overfitting. If just the training error is used, then a network that has been overfitted might have a higher fitness than a network that cannot generalize well at all.

6.3 Training and Optimization Processes

Genetic optimization of the ANF PSS contains two major processes:

a) Search for the optimum number of rules and shape of MFs by using GA.

b) Training the network to determine the consequent parts of rule base by the gradient descent algorithm.

Data from not only the typical operating conditions but also over as wide a range of operating conditions as the system is likely to encounter, must be used for proper training of an ANF PSS. In this case, the training data and checking data were obtained over the generator output ranging from 0.1 *p.u.* to 1.0 *p.u.*, 0.7 *p.f.* lead to 0.1 *p.f.* lag, and a wide spectrum of disturbances with the self-optimizing pole-shifting APSS [26] acting as a non-linear power system stabilizer and with all appropriate excitation limits in place. These disturbances are: reference voltage and infinite bus voltage disturbances in the range of $\pm$ 0.05 *p.u.*, governor input torque variations of $\pm$ 0.3 *p.u.*, one transmission line outage, and three phase fault on one circuit of the double circuit transmission line.

The optimum triangular MFs for two inputs of the controller obtained by GA are shown in Fig. 6.2, four MFs for $\Delta\omega$ and six MFs for $\Delta\dot{\omega}$. In genetic optimization, the probability of crossover and mutation operators are chosen to be 80% and 5% respectively and the population is 25.

Since for each generation all networks have to be trained individually with the whole set of training data, this takes a very long computation time. The whole set of training data was reduced to 15% during genetic optimization. After the convergence of GA, by which the structure of the optimum network is achieved, the complete

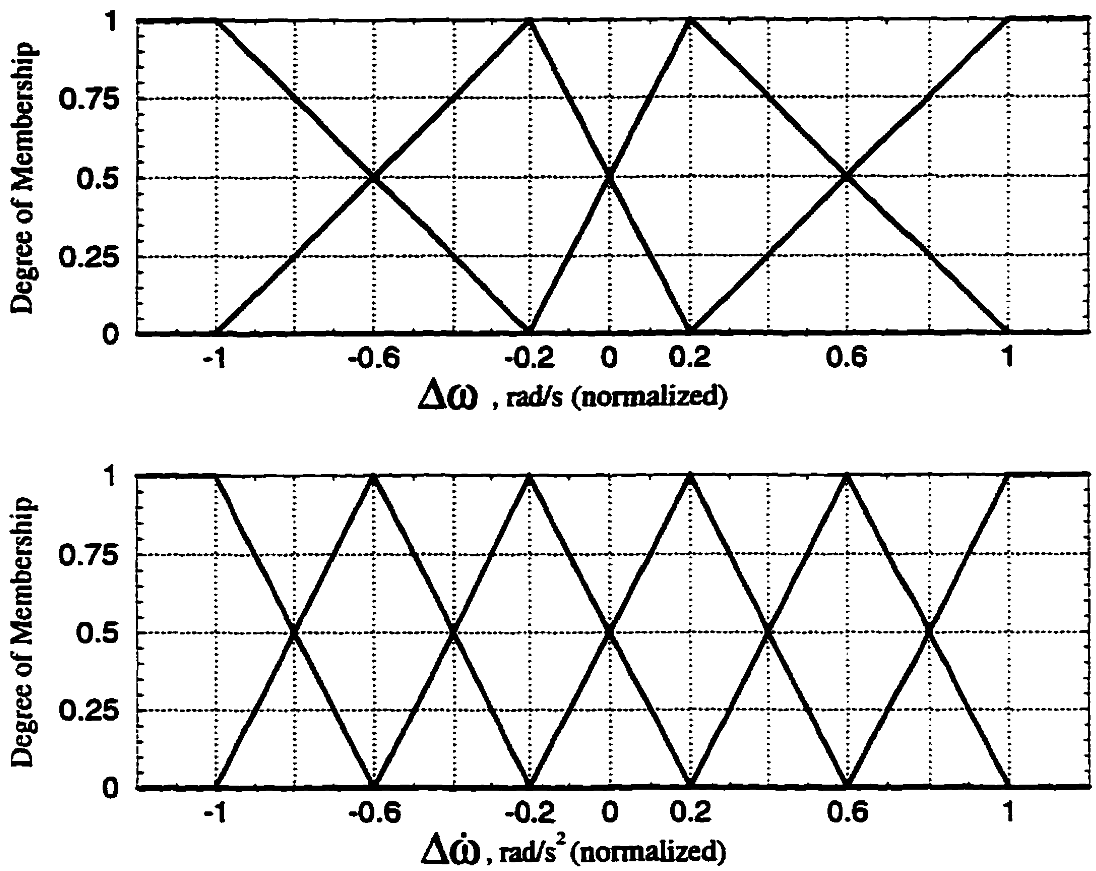

Figure 6.2: Optimized Membership Functions.

training data (15000 pairs) was employed to fine tune the MFs and inference rules with the initial values of MFs obtained from GA optimization. The final set of MFs shown in Fig. 6.3 has the same number for both input signals, but the shape of MFs is changed from the triangular to bell function with smoother characteristics.

The initial control action surface and the final control action surface after complete training are shown in Fig. 6.5 and Fig. 6.4 respectively. The final control action surface after complete training depicted pictorially the effect of non-linearity in controller. The control action before optimization is completely flat.

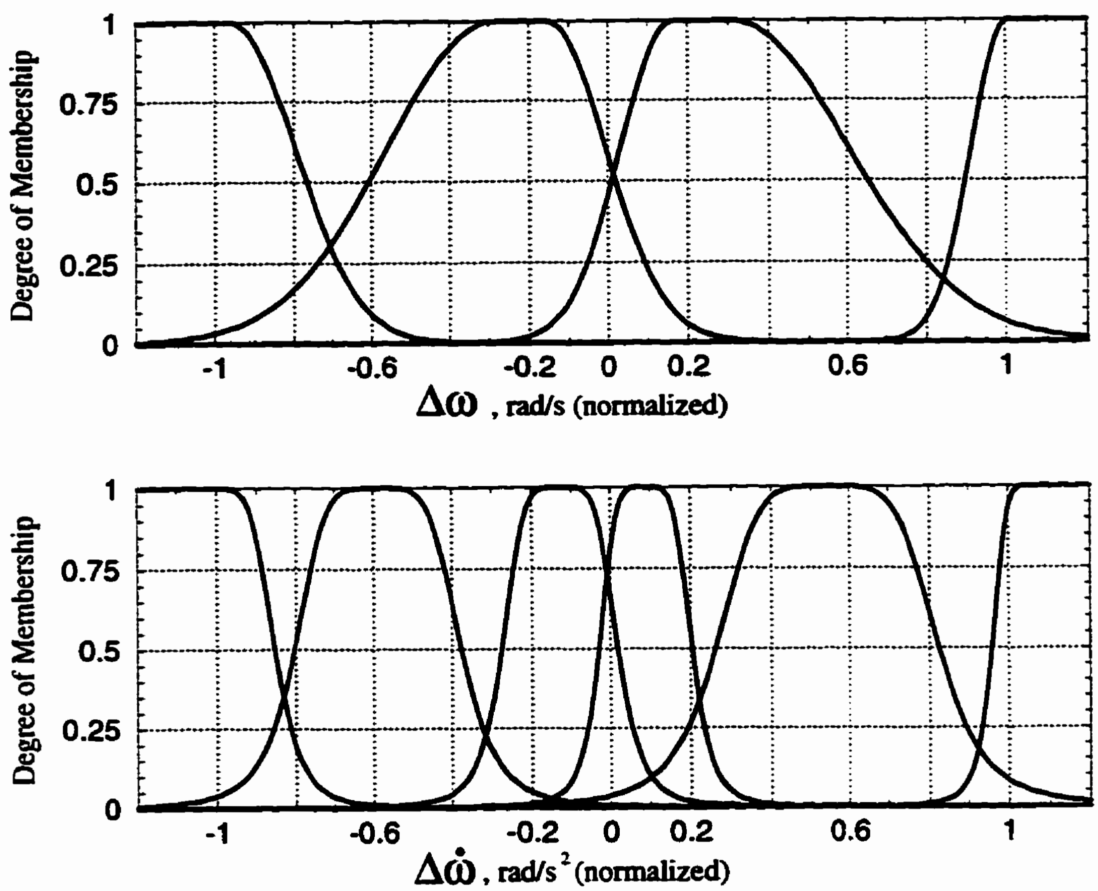

Figure 6.3: Optimized MFs after complete training.

6.4 Test Results

A number of studies have been performed to investigate the effect of the proposed stabilizer and the results are compared with those of the CPSS. The system used for the studies is a 7th order non-linear model of a generating unit as described in Chapter 4.

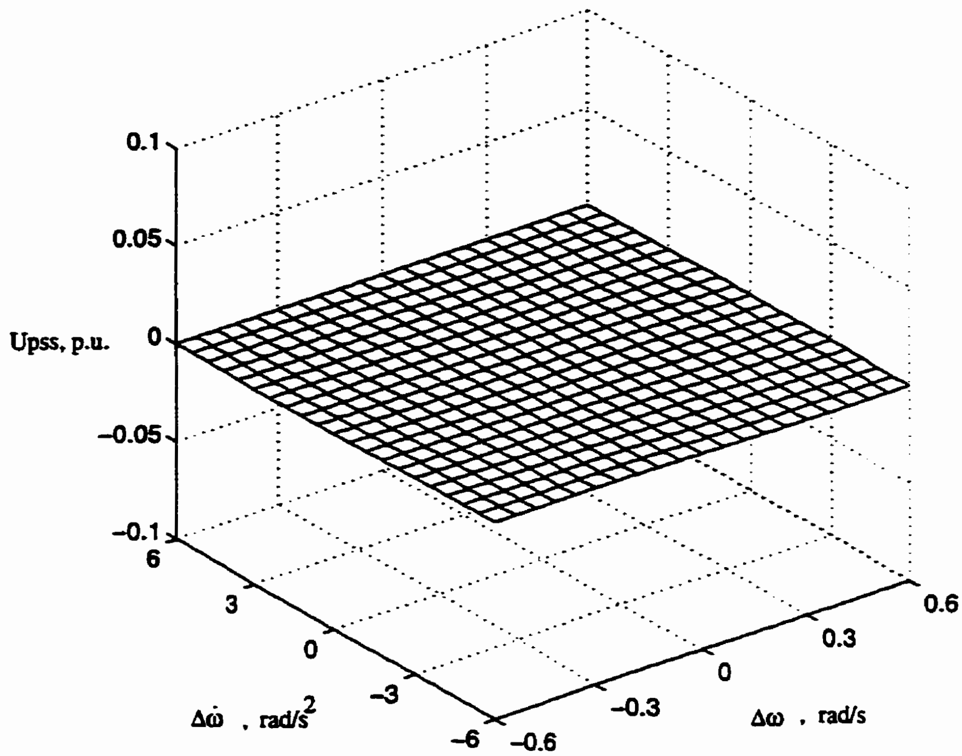

Figure 6.4: Control action surface before training.

6.4.1 CPSS Parameter Tuning

With the generator operating at 0.7 *p.u.* power, 0.85 *p.f.* lag, a 0.05 *p.u.* step in input torque reference is applied at time 1 *s*. At time 5 *s*, the system returns to its initial operating condition. Under these conditions, the CPSS was carefully tuned for the best performance, i.e. the overshoot and settling time were minimized by the CPSS damping effect. The parameters of the CPSS were then kept unchanged for all the tests described in this Chapter. Results of the study with the ANF PSS, CPSS and without a stabilizer given in Fig. 6.6 show that the ANF PSS damps out the low frequency oscillations very quickly.

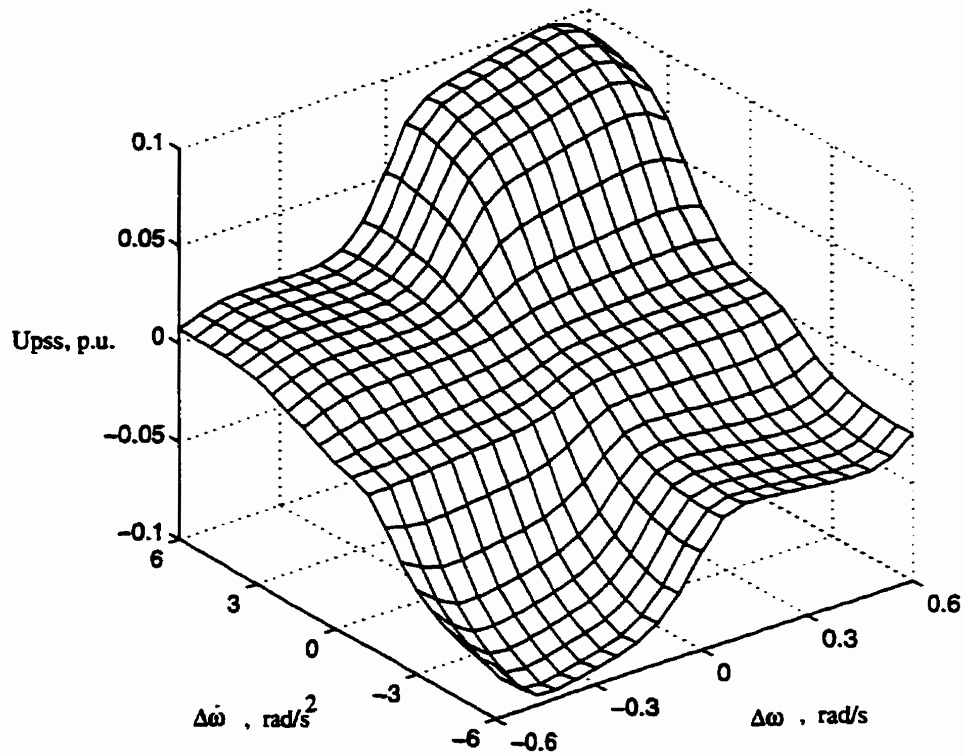

Figure 6.5: Control action surface after complete training.

6.4.2 Light Load

With the system now operating under a light load condition of 0.20 *p.u.* power, 0.85 *p.f.* lag, a 0.15 *p.u.* step increase in input torque reference is applied. System response for these conditions is shown in Fig. 6.7. Despite a large change in the operating conditions the ANF PSS still provides very effective performance.

6.4.3 Fault Test

The behavior of the proposed ANF PSS under transient conditions was further verified by applying a fault. The response of the power system to a three phase to

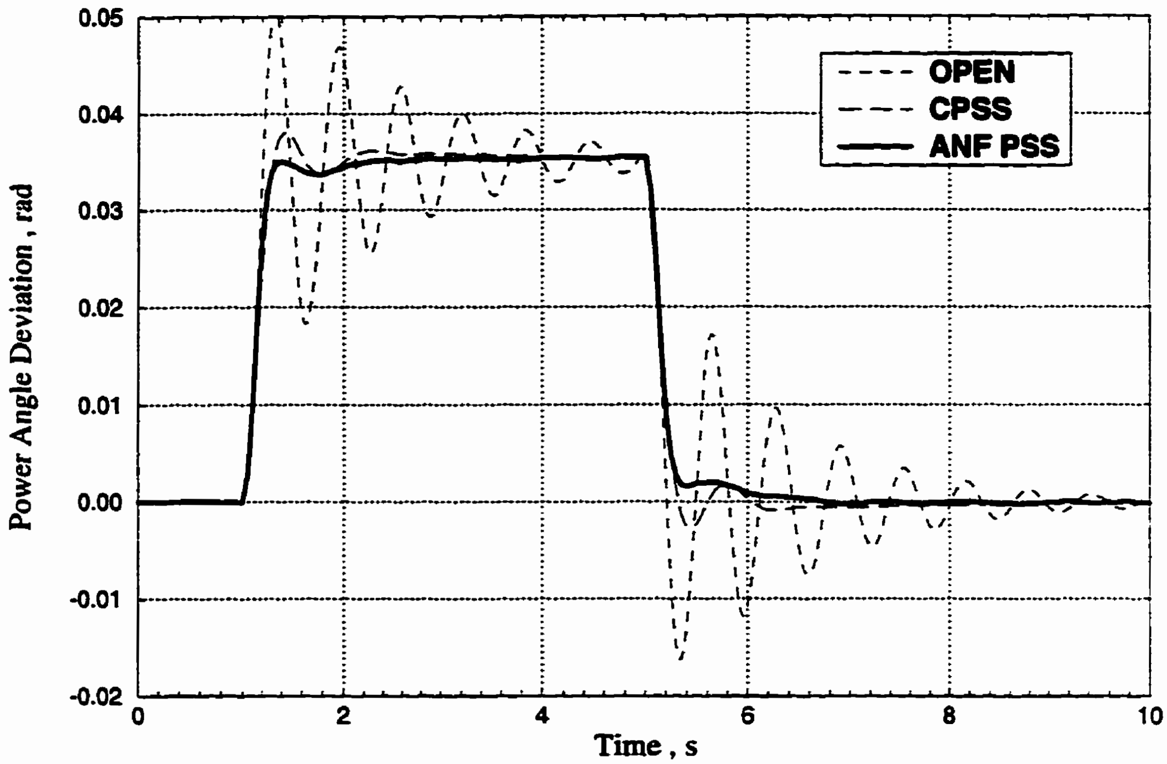

Figure 6.6: Response to a 0.05 $p.u.$ step increase in torque and return to initial conditions.

ground short circuit at the middle of one transmission line, cleared 200 ms later by the disconnection of the faulted line and successful reclosure after 4 s is shown in Fig. 6.8. The results show that ANF PSS minimizes the deviation of the power angle of the generator after the fault and helps the system to reach the new operating point very quickly.

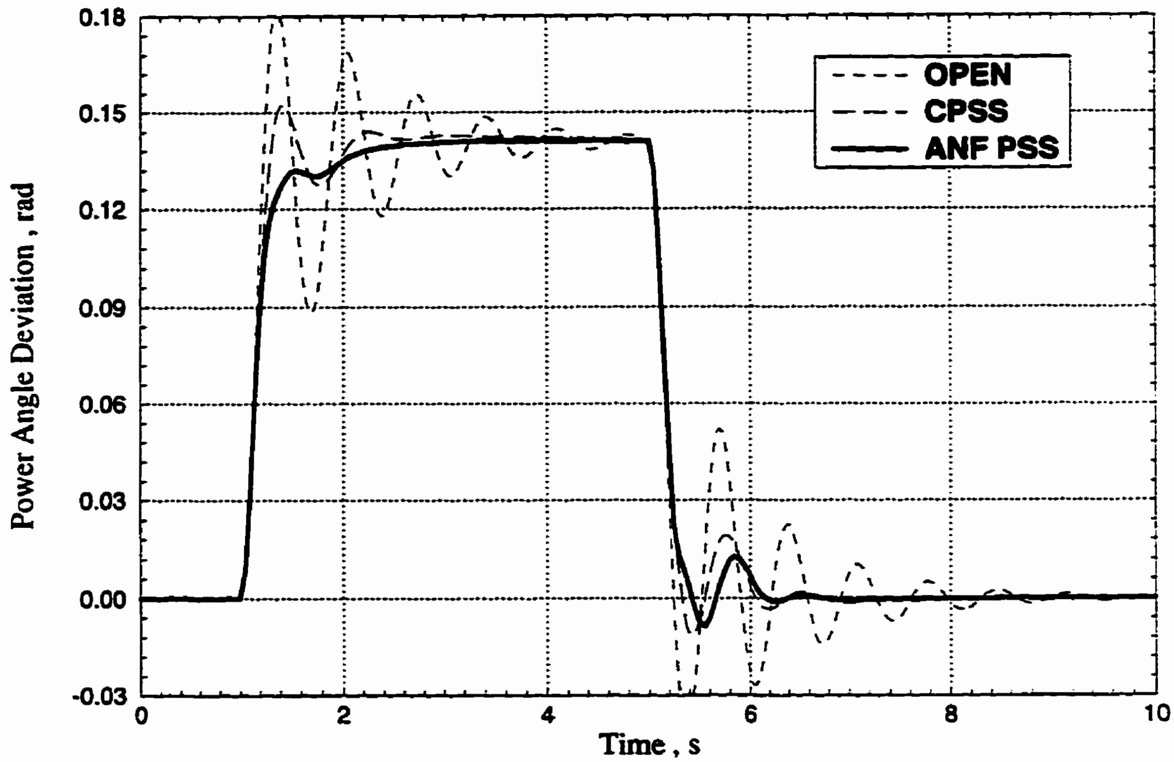

Figure 6.7: Response to a 0.15 $p.u.$ step increase in torque and return to initial conditions.

6.4.4 Control Signals

The ANF PSS supplementary control signals for the three previous tests are shown in Fig. 6.9.

6.4.5 Effect of optimization with GA

With the generator operating at 0.3 $p.u.$ power, 0.9 $p.f.$ lead, a 0.10 $p.u.$ step increase in torque reference was applied. Closed-loop system response of two ANF PSSs with

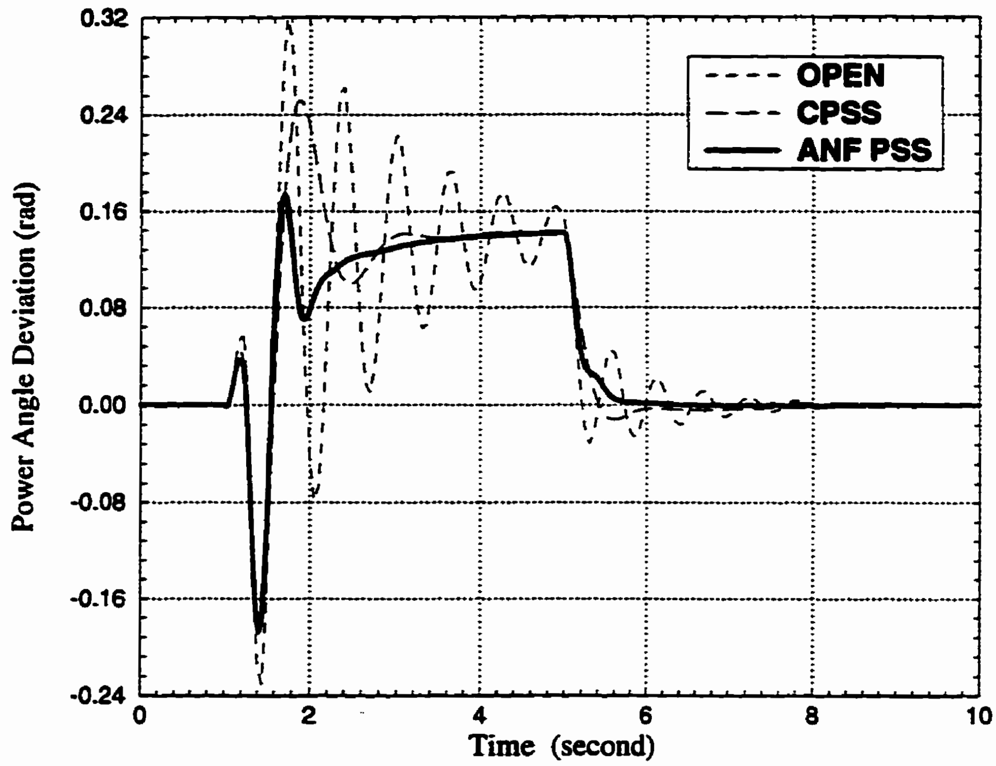

Figure 6.8: Response to a three phase to ground fault at the middle of one transmission line.

different number of MFs is given in Fig. 6.10. The structure for the first one is chosen to be 7 MFs for each input, resulting in 49 inference rules. The structure for the second ANF PSS is as Fig. 6.3 obtained from GA optimization; 4 and 6 MFs respectively for the two inputs, $\Delta\omega$ and $\Delta\dot{\omega}$, resulting in 24 inference rules.

Although in general, there is not much difference between the performance of these two stabilizers, the second has a reduced structure, requiring less memory and less computation time.

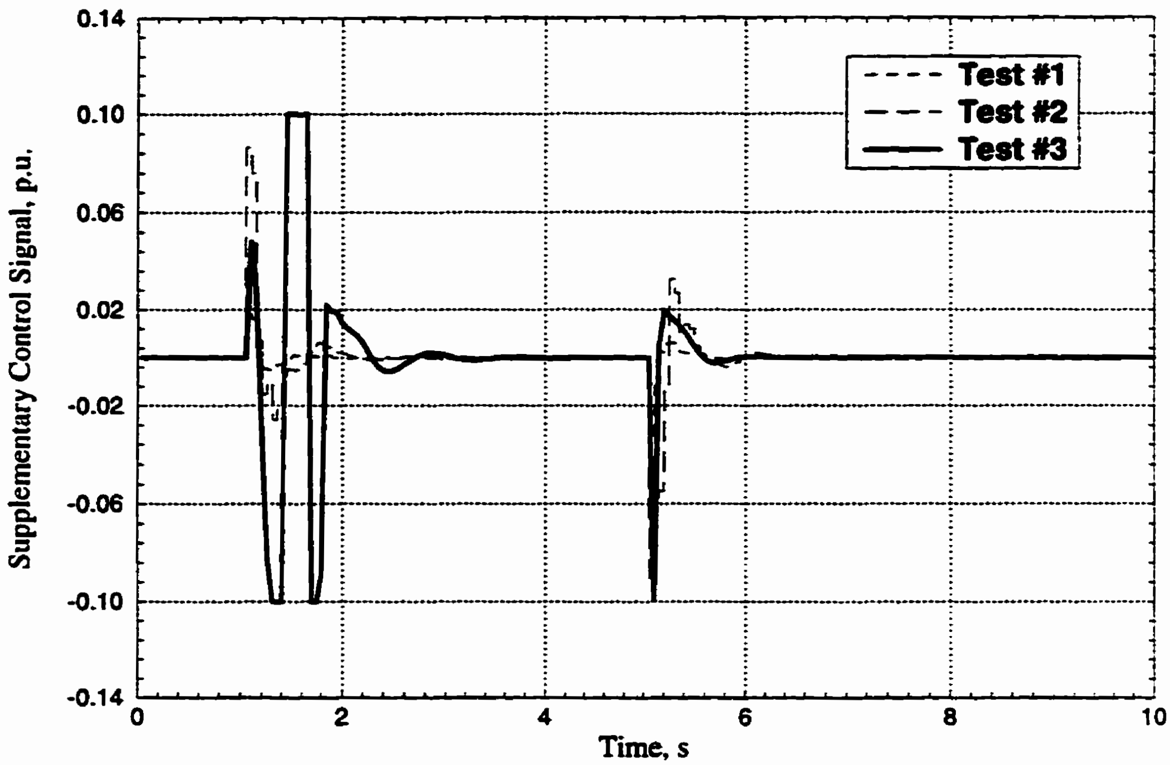

Figure 6.9: Supplementary control signal of ANF PSS for previous tests.

6.4.6 Stability Margin

The introduction of the supplementary controller not only improves the dynamic performance, but also increases the stability margin. To demonstrate this effect, the input torque reference was increased gradually from the initial value, 0.95 *p.u.*, 0.9 *p.f.* lag. During this test the terminal voltage remained constant as long as the system was stable. The dynamic stability margin is described by the maximum power output at which the system losses synchronism. The result for the system without

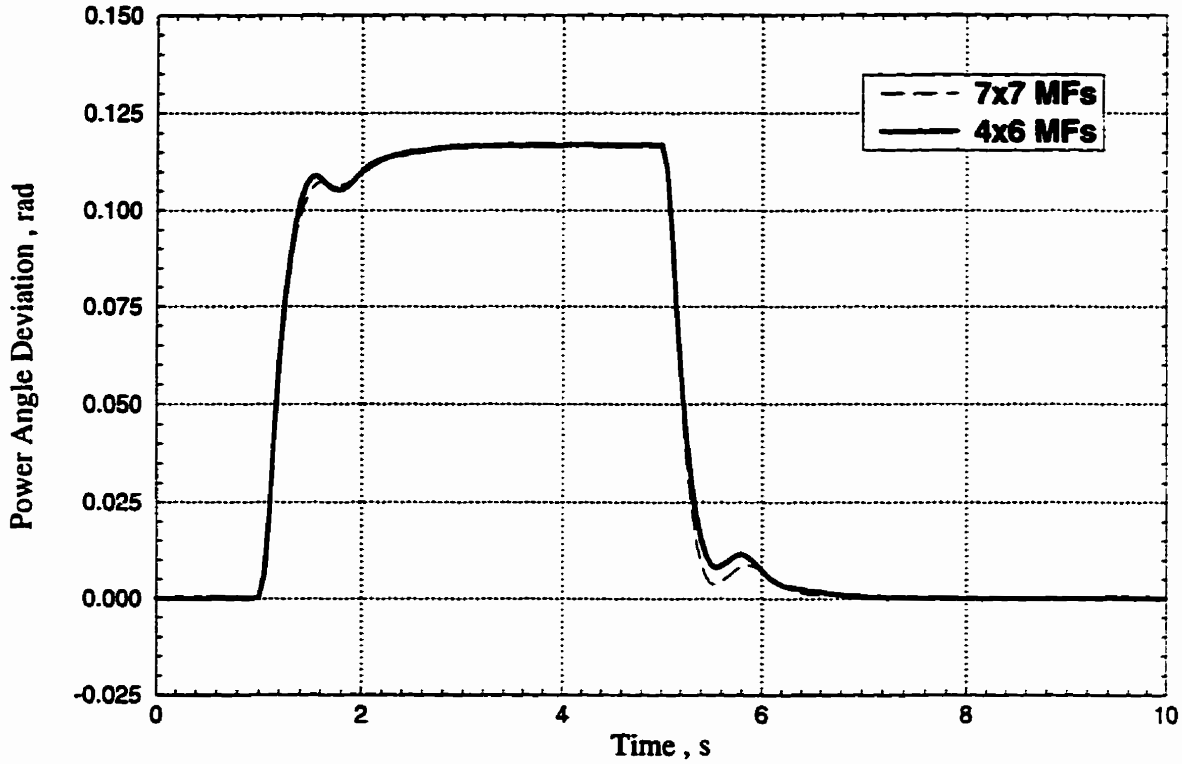

Figure 6.10: Comparison of two ANF PSSs, with 4x6 and with 7x7 MFs.

stabilizer, with CPSS and with optimized ANF PSS are given in Table 6.1. ANF PSS provides the largest output power, which indicates that the system dynamic stability is improved most by the ANF PSS.

6.5 Summary

A genetic approach for optimization of adaptive-network-based fuzzy logic controller and its application to a power system are presented in this Chapter. By employing

Table 6.1: Dynamic stability margin results.

	OPEN	CPSS	ANF PSS
Maximum Power	1.35 $p.u.$	2.85 $p.u.$	3.15 $p.u.$
Maximum Rotor Angle	1.05 $rad.$	1.78 $rad.$	1.95 $rad.$

an adaptive network, fuzzy logic controller is able to learn the input-output behavior of a complex controller. However, selecting the optimum number of inference rules is still a tedious task. Genetic algorithm, as a global optimization technique, is employed to determine the shape and the number of membership functions which in turn defines the number of rules. The results show that the proposed optimized ANF PSS provides good damping over a wide operating range and significantly improves the dynamic performance of the system.

When the proposed ANF PSS is to be used in a multi-machine power system environment, it faces some other problems, such as multi-mode oscillations, coordination with other PSSs of the same or different type, etc. Use of the proposed ANF PSS in a multi-machine power system environment is investigated in the next Chapter.

Chapter 7

Self-Learning Adaptive Fuzzy Logic PSS in Multi-Machine Power System

7.1 Introduction

Simulation studies in Chapters 4, 5 and 6 showed that properly trained ANF PSSs can provide an effective damping of the power system [87][88][89]. These studies were on the single-machine infinite-bus environment. The effectiveness of the ANF PSS to damp multi-mode oscillations in multi-machine environment needs to be verified.

The effectiveness of the ANF PSS to damp multi-mode oscillations in a multi-machine environment is investigated in this Chapter. A five machine power system is used in this study and its transient response to a large disturbance is presented with the multi-mode oscillation phenomenon.

Multi-mode oscillations appear in a multi-machine power system in which the interconnected generating units have quite different inertia and they are weakly connected by transmission lines. These oscillations are generally analyzed in three main oscillation modes, i.e. local, inter-area and inter-machine modes. Depending upon their location in the system, some generators participate in only one oscillation mode, while others participate in more than one mode [17].

Speed deviation, $\Delta\omega$, and accelerating power, ΔP_e, are chosen as the inputs to ANF PSS. It is demonstrated by the simulation results that when installed on different machines, the proposed ANF PSS can adjust itself to provide good damping for different oscillation modes, such as the local and inter-area mode. Also, the self-coordination capabilities of the ANF PSS with other ANF PSSs and conventional PSSs are demonstrated.

7.2 Power System Multi-Mode Oscillations

There are three modes of oscillation in a multi-machine power system:

Local Mode – usually refers to oscillations occurring in plant transients stemming from generator rotors oscillating relative to the combined equivalent inertia of the system. This is also described as the generator swinging relative to an infinite bus formed by the combined equivalent inertia external to a particular generator as shown in Chapter 4, 5 and 6. Frequency magnitudes are directly related to the equivalent rotational inertia of the generator and the prime mover, and to the synchronous torque coefficient linking the generator to the fixed bus. Local mode oscillations are in the range of 0.8 to 2 Hz.

Inter-Machine Modes – this describes frequencies related to closely coupled generators swinging relative to each other. This can occur at a plant that has a diverse mix of generators and controllers or at neighboring plants that are linked with inter-ties such that the machines are relatively closely coupled. Inter-machine frequencies are related to the equivalent machine inertia of the closely coupled generator groups and are in the range of 0.3 to 1 Hz.

Inter-Area Modes – these frequencies stem from coherent groups of generators in one area swinging relative to a number of other coherent groups in other areas. Inter-area frequencies are in the range of 0.1 to 0.7 Hz and these frequencies may overlap with frequencies described under the other two modes.

7.3 A Multi-Machine Power System Model

A detailed 5th order five-machine power system without infinite bus, as shown in Fig. 7.1, is used to test the proposed ANF PSS. Five generators are connected through a transmission network. Generators G_1, G_2 and G_4 have much larger capacities than G_3 and G_5. Parameters of all generators, governors, AVRs, transmission lines and loads operating conditions are given in Appendix B. G_3, G_2 and G_5 maybe considered to form one area, and G_1 and G_4 a second area. The two areas are connected together through a tie line connecting buses 6 and 7. Under normal conditions, each area serves its own load and is almost fully loaded with a small load flow over the tie line.

When this system is disturbed, multi-mode oscillations arise because of the different sizes of the generators and the network configuration. The multi-mode oscillations can be clearly observed in Fig. 7.2. A 0.10 $p.u.$ step decrease in the mechanical input torque reference of G_3 is applied at 1 s, and the system returns to the original condition at 10 s. Under the above mentioned disturbances without any PSS installed, the local mode oscillation at a frequency of about 1.3 Hz and the inter-area mode of about 0.65 Hz are quite distinct. This is because of the large difference in the inertia of the generators. The speed difference between G_2 and G_3 exhibits mainly local mode oscillations, while the speed difference between G_1 and G_2 shows

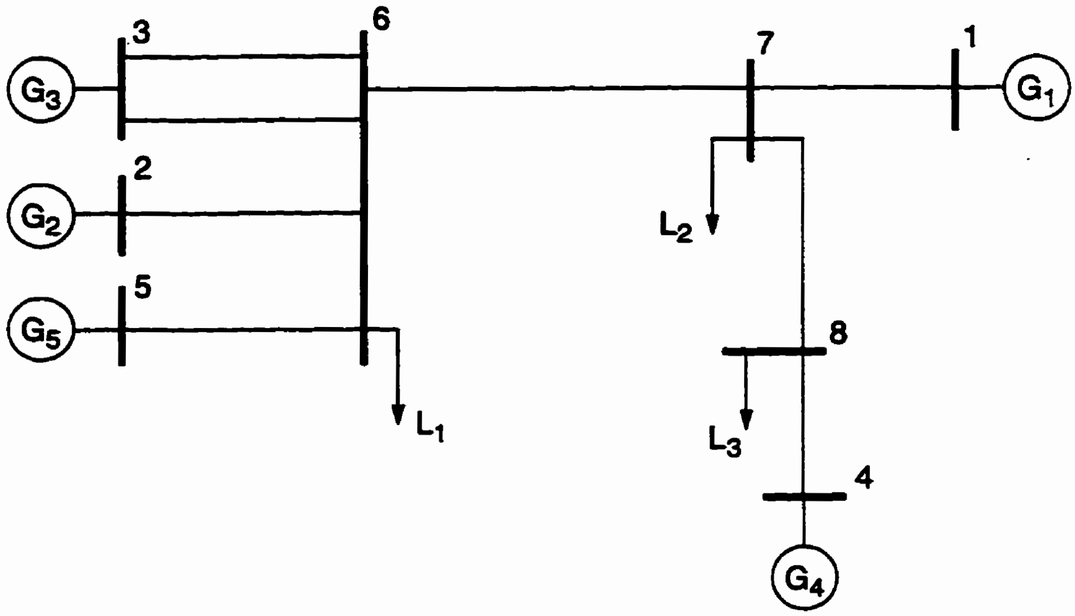

Figure 7.1: A five machine power system configuration

the inter-area mode oscillations. Both local and inter-area oscillations exist in the speed difference between G_1 and G_3.

7.4 The Effectiveness of ANF PSS in Damping Multi-Mode Oscillations

The ANF PSSs employed in this test are the same as those developed in Chapter 3 and tested in the single-machine infinite-bus environment in Chapter 5. Accelerating power, ΔP_e, and speed deviation, $\Delta\omega$ are used as the inputs to the stabilizer as shown in Fig. 7.3. Since there is no infinite bus in the system, speed deviation has a DC offset value. A washout filter is utilized to remove the DC value before speed deviation signal is fed to the stabilizer.

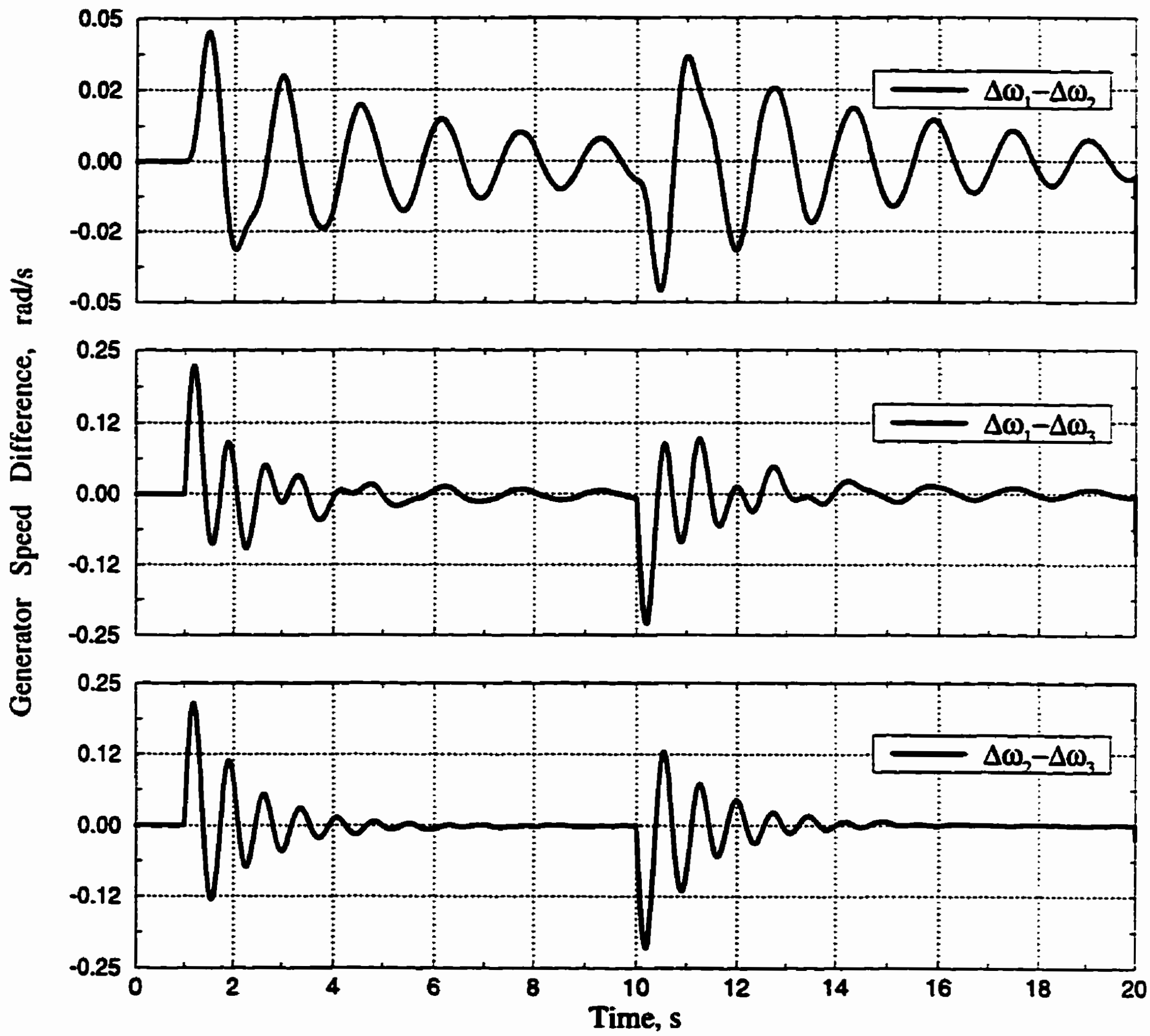

Figure 7.2: Multi-mode oscillations of the five-machine power system.

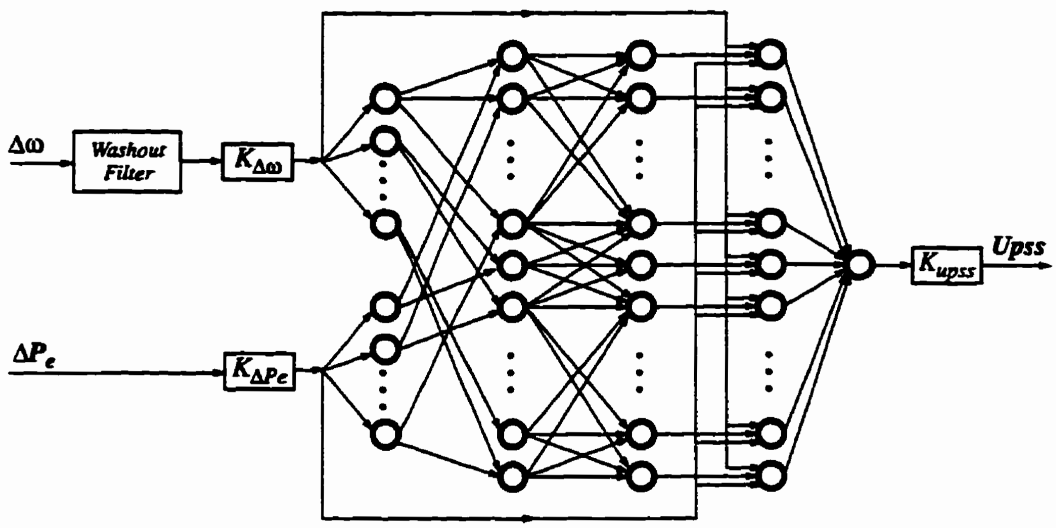

Figure 7.3: Structure of ANF PSS used in multi-machine tests

7.4.1 Only One PSS Installed

Under the same disturbance, 0.10 *p.u.* step decrease in the mechanical input torque reference of G_3, the proposed ANF PSS was installed on G_3 and none of the other generators was equipped with PSS. The speed deviation, $\Delta\omega$, and the accelerating power, ΔP_e, were sampled at the rate of $20Hz$.

The training was performed over a wide range of conditions for both the controller and the plant model. The generator output ranging from 0.1 *p.u.* to 1.0 *p.u.*, and the power factor ranging from 0.7 lead to 0.1 lag. Similarly, a wide spectrum of possible disturbances was used to obtain the training data. These disturbances are: reference voltage disturbances in the range of -0.05 *p.u.* to 0.05 *p.u.*, governor input torque variations from -0.15 *p.u.* to 0.15 *p.u.*, one transmission line outage, and three phase fault on one circuit of the double circuit transmission line.

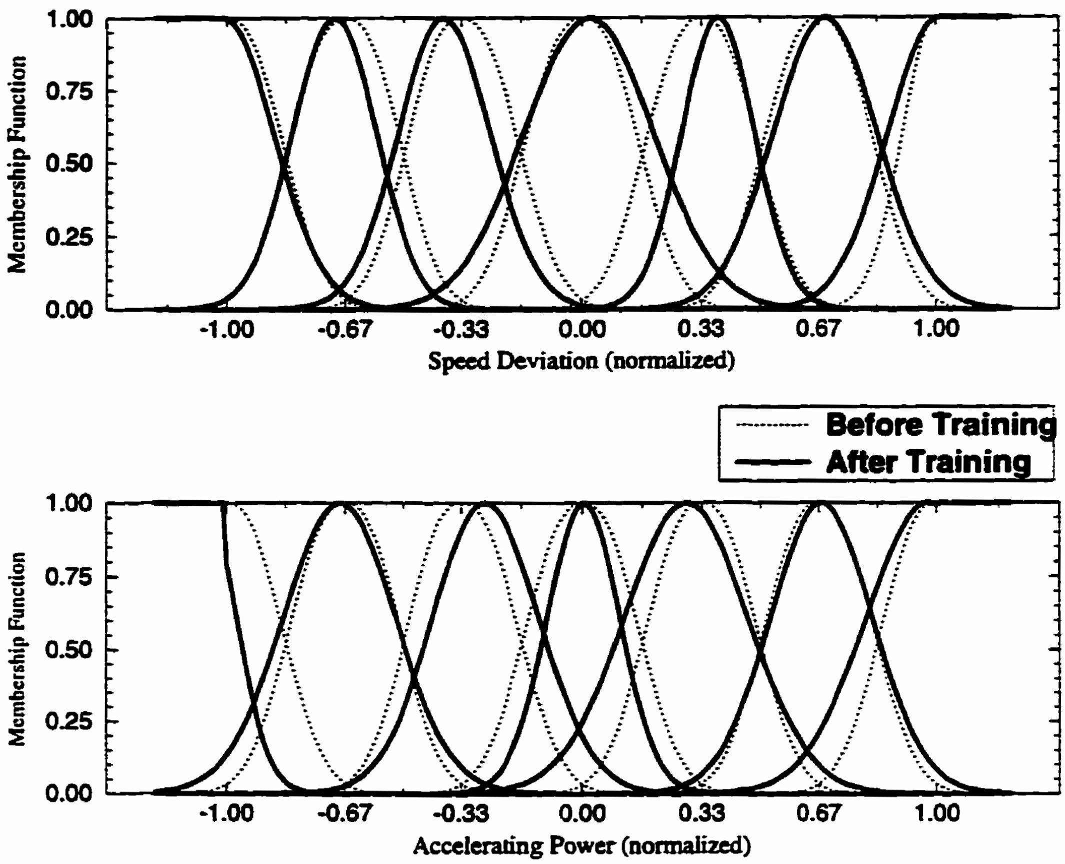

Figure 7.4: Membership functions before and after training

The MFs for two inputs of the controller, $\Delta\omega$ and ΔP_e, before and after training are shown in Fig. 7.4.

After the complete training, as shown in Fig. 7.5, the ANF PSS damps the local mode oscillations very effectively. However, as expected, it has little influence on the inter-area mode oscillations. This is because the rated capacity of G_3 is much less than G_1 and G_2; and G_3 does not have enough power to control the inter-area mode oscillations.

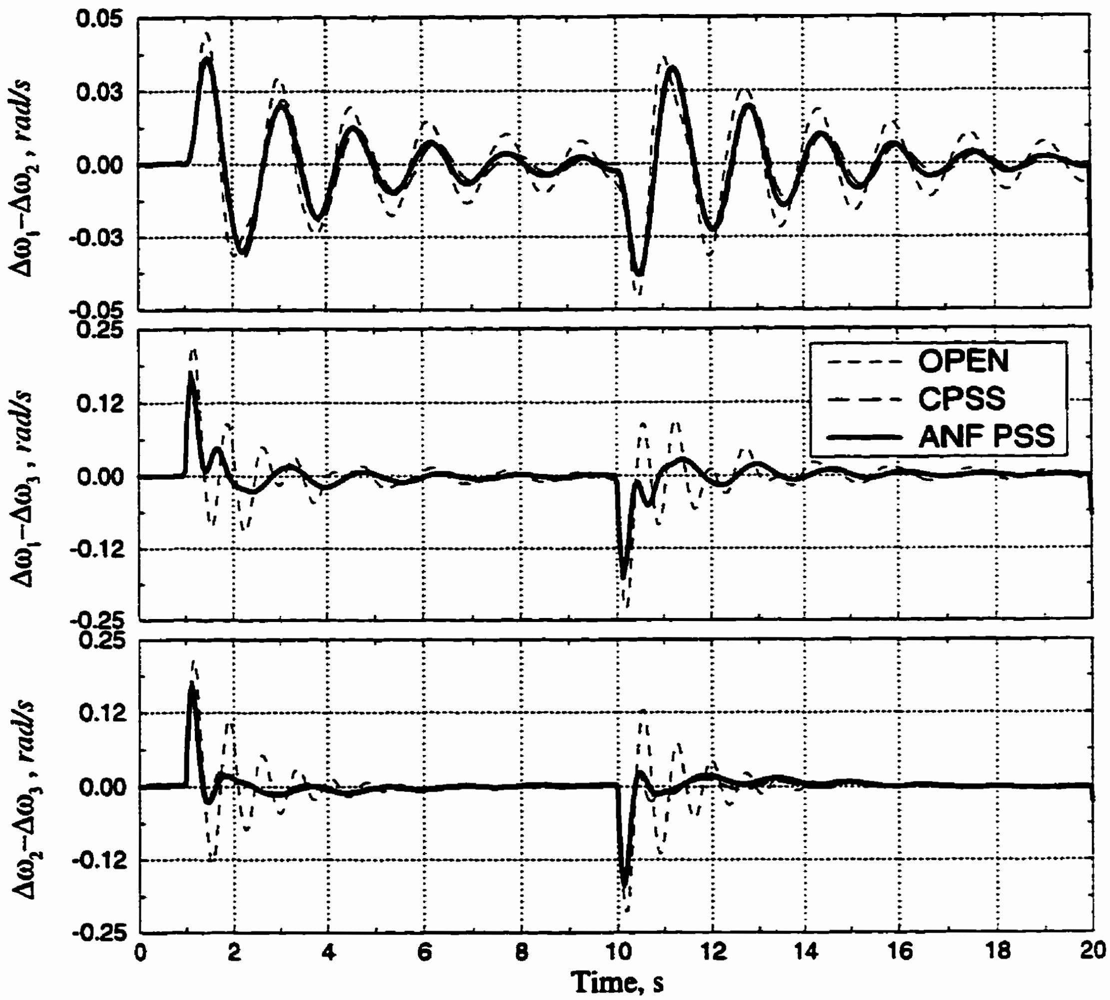

Figure 7.5: System response with ANF PSS and CPSS installed on G_3.

For comparison, in a separate test a conventional PSS (CPSS) with the following transfer function [90] was installed on the same generator:

$$U_{pss}(s) = K_s \frac{sT_5}{1+sT_5} \frac{1+sT_1}{1+sT_2} \frac{1+sT_3}{1+sT_4} \Delta P_e(s) \tag{7.1}$$

After careful parameter tuning, the CPSS with the following parameter set performs almost similar to the ANF PSS.

$$K_s = 1.0,\ T_1 = T_3 = 0.3,\ T_2 = T_4 = 0.10,\ T_5 = 0.4$$

The results are shown in Fig. 7.5. The same conclusion as that for the ANF PSS can be drawn for the CPSS.

7.4.2 With Three PSSs Installed

To damp both local and inter-area modes of oscillations, the ANF PSSs were installed on G_1, G_2 and G_3. The ANF PSS on G_3 was kept the same as the previous test, however, due to the different size of input signals, the ANF PSS on G_1 was retrained with the initial parameters acquired from ANF PSS on G_3. Then the latest parameters were duplicated to that of ANF PSS on G_2. Responses given in Fig. 7.6 show that both modes of oscillations are damped out effectively.

Fig. 7.6 also depicts the system response when CPSSs are installed on G_1, G_2 and G_3. The proper parameter set for the CPSS on G_1 and G_2 is:

$$K_s = 1.0,\ T_1 = T_3 = 0.3,\ T_2 = T_4 = 0.01,\ T_5 = 0.4$$

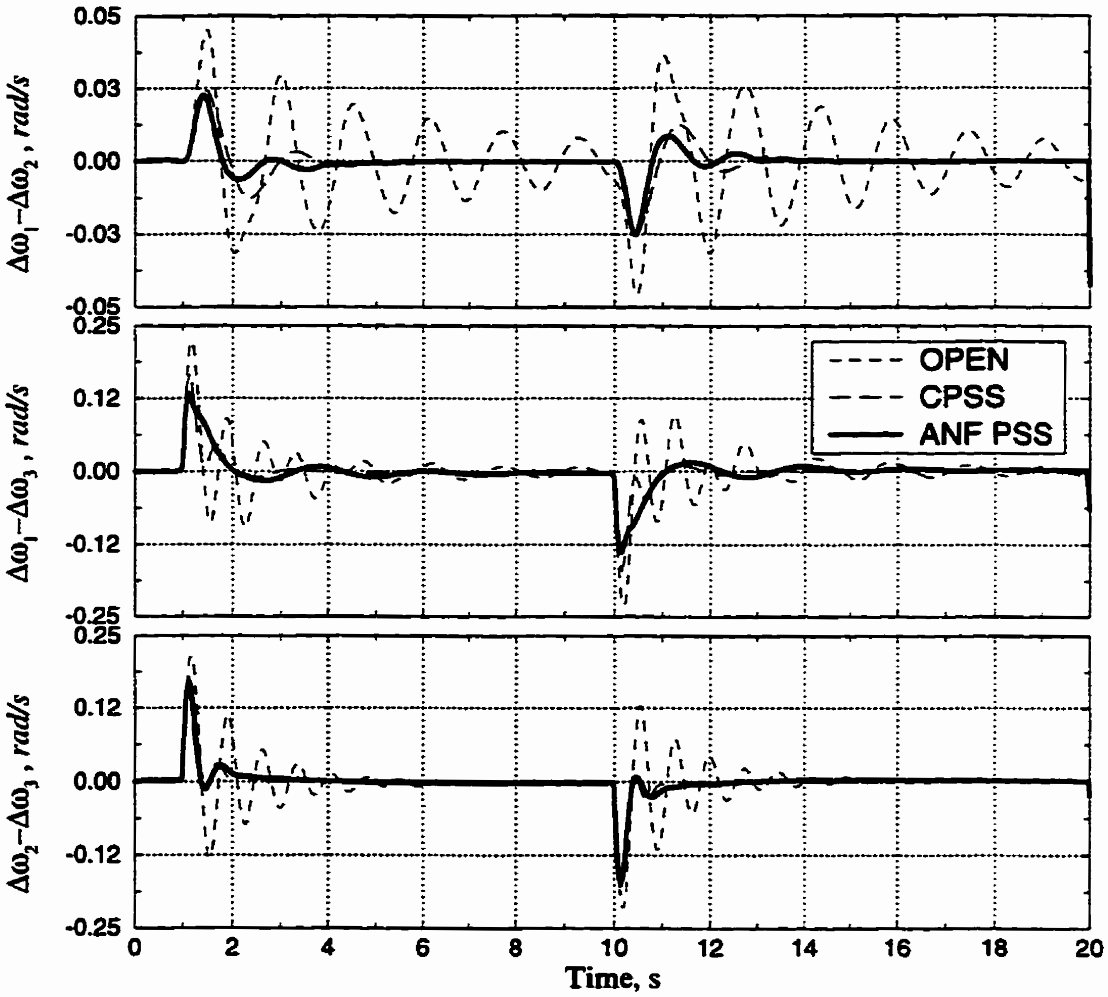

Figure 7.6: System response with ANF PSS and CPSS installed on G_1, G_2 and G_3.

7.4.3 Coordination Between ANF PSS and CPSS

In practice, the newly installed ANF PSS will have to work together with CPSSs which already exist in a power system. For the five machine power system, the proposed ANF PSS was installed on G_1 and G_3, with CPSSs on G_2, G_4 and G_5. Fig. 7.7 shows the system performance and demonstrates that the two types of PSSs can work co-operatively.

7.4.4 Three Phase to Ground Fault Test

So far, the parameters of CPSSs are tuned under the disturbance of 0.10 *p.u.* step change in the mechanical input torque reference of G_3. It has been shown that performance of the CPSS at a specific operating point can be satisfactory if its parameters are tuned properly at that operating point and under the same disturbance.

To compare the performance of ANF and CPSS under different disturbances, a three phase to ground fault was applied at the middle of one transmission line between buses #3 and #6 at 1 *s* and cleared 100 *ms* later. At 10 *s*, the faulted line was restored successfully. The disturbance is large enough to cause the system to work in the nonlinear region. Fig. 7.8 shows the system response when the proposed ANF PSSs are installed on G_1, G_2 and G_3. It shows the closed-loop response of the system when CPSSs are installed on the same generating units.

From these two system responses, it can be concluded that because the CPSS is designed for best performance for the small disturbances, it is not able to provide as effective a damping. However, despite a large change in the operating conditions the ANF PSS provides very effective performance.

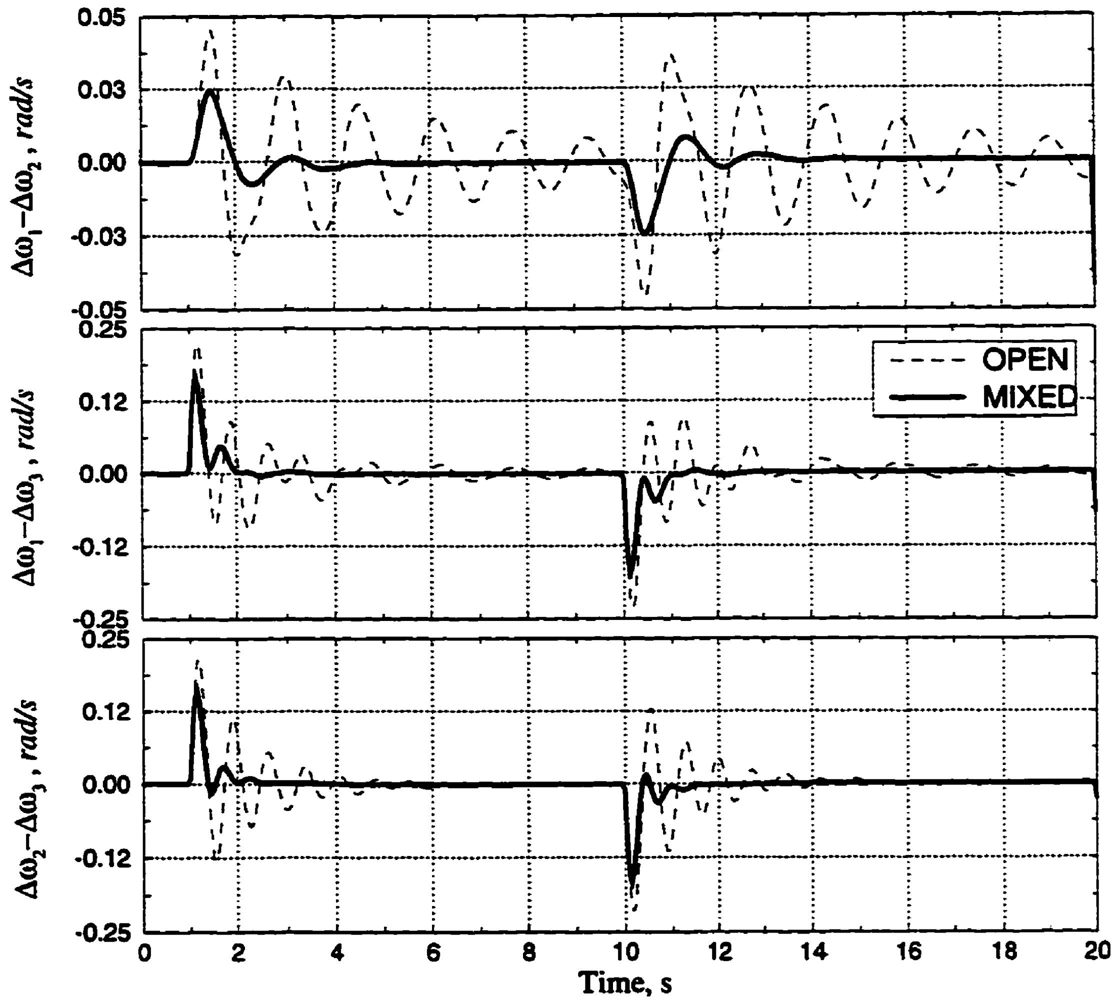

Figure 7.7: System response with ANF PSSs on G_1 and G_3 and CPSSs on G_2, G_4 and G_5.

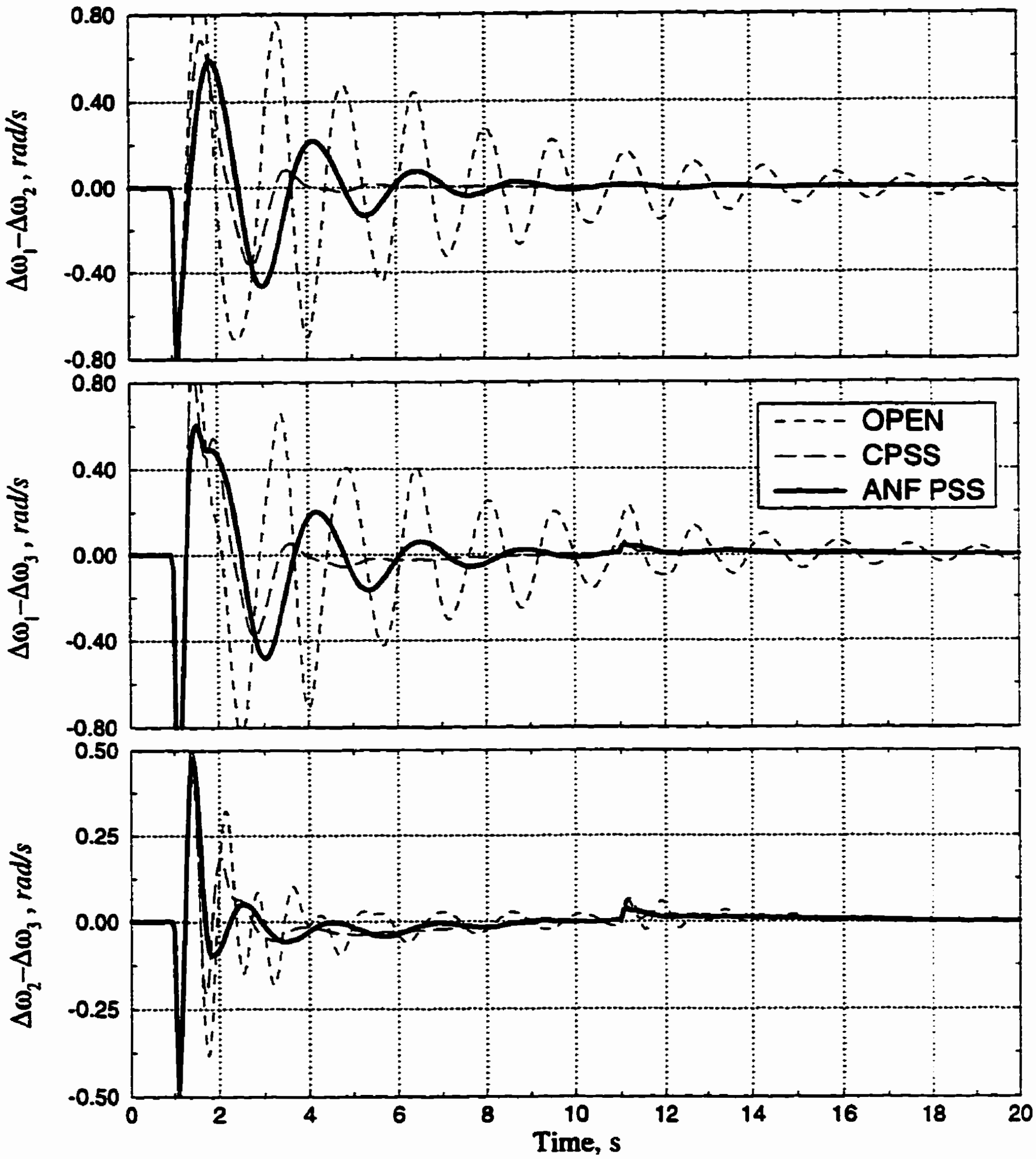

Figure 7.8: System response to three phase to ground test

7.4.5 New Operating Condition Test

The behavior of ANF PSS and CPSS under other operating condition is investigated in this test. The new system operating point is given in Appendix B. With ANF PSSs and CPSSs installed on G_1, G_2 and G_3, Figs 7.9 shows the system response under a three phase to ground fault. Again, it is shown that the system response with ANF PSSs is further improved than with CPSS. The reason is that the ANF PSS is designed to capture the nonlinearity of the power system, whereas the CPSS is a linear controller.

7.5 Summary

In this Chapter, the effectiveness of an ANF PSS in damping the multi-mode oscillations of a five machine power system environment is investigated. The accelerating power and speed deviation of the generating unit are used as the inputs to ANF PSS. Training procedure for the proposed stabilizer is based on a self-learning technique; i.e. independent of other PSS. The ANF PSS also was trained over the full working range of the generating unit with a large variety of disturbances to capture the non-linear behavior of the power system. This is a desirable characteristic the conventional PSS lacks. Also, the coordination of the ANF PSS with the CPSS is well demonstrated.

From the simulation results in this part of dissertation, it can be seen that the proposed ANF PSS can produce satisfactory performance when it is used in both the single-machine power system and the multi-machine power system. The behavior of the proposed ANF PSS applied to a physical system is studied in the next part.

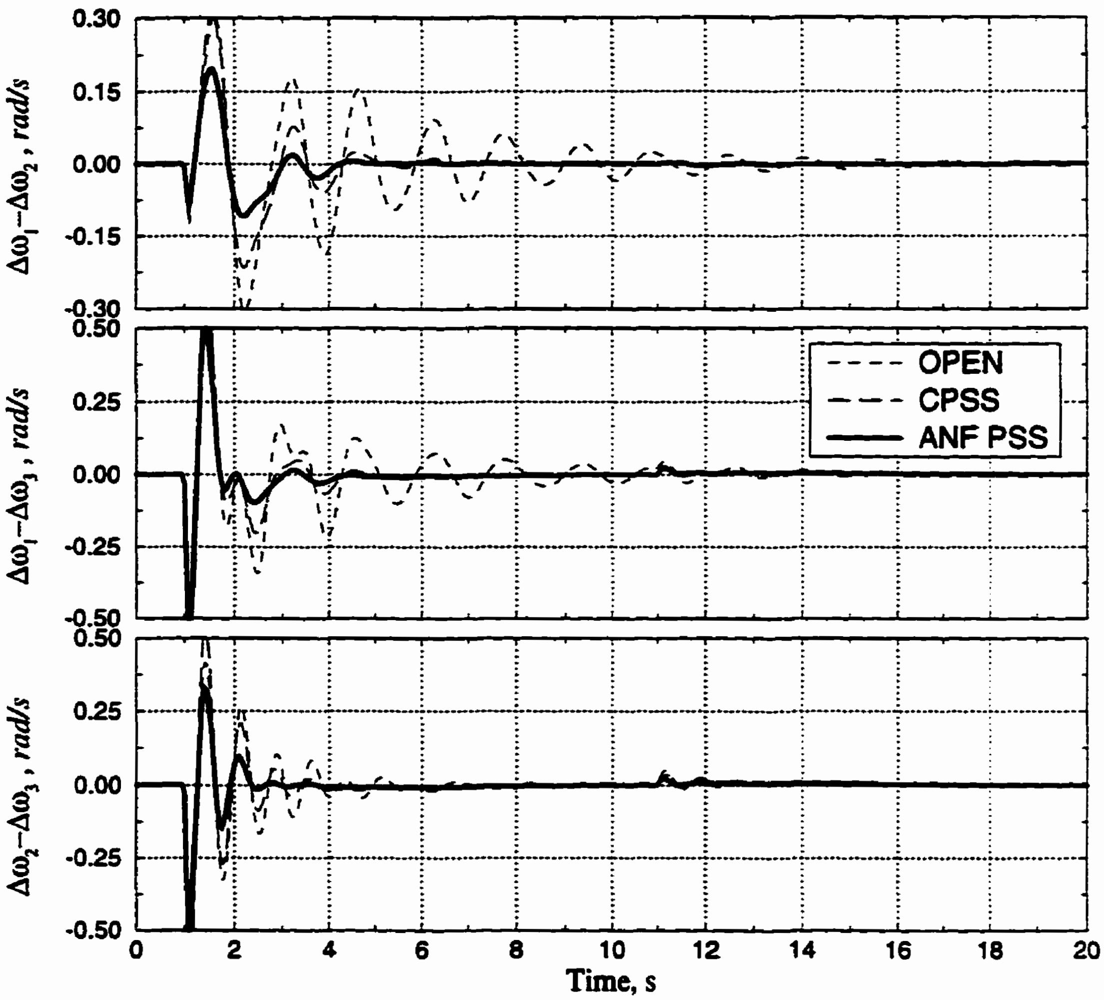

Figure 7.9: System response to three phase to ground test for the new operating condition.

Part III

Experimental Tests

Chapter 8

Experimental Studies with a Self-Learning Adaptive Fuzzy Logic PSS

8.1 Introduction

Theoretical development of a self-learning Adaptive-Network based Fuzzy Logic Power System Stabilizer (ANF PSS) is described in Chapter 3[87]. It has been simulated on a single-machine infinite-bus power system (Chapter 4, 5 and 6) and a multi-machine power system (Chapter 7). The proposed fuzzy controller has the powerful capability of learning and adaptation. In this approach a fuzzy PSS with learning ability has been constructed. Fuzzy rules and MFs of the controller can be tuned automatically by the learning algorithm. Moreover, it is not dependent on another existing controller. In other words, it employs a self-learning scheme in which the ANF PSS is trained from the performance of the generating unit output and not the controller output [88].

After the theoretical development and computer simulation studies, the performance of the ANF PSS is investigated further on a physical model of a power system. Scaled physical model is able to emulate the behavior of the actual power plant in the laboratory environment. The ANF PSS has been implemented on a Digital Signal Processor (DSP) mounted on a PC.

For comparison, a digital conventional PSS (CPSS) was implemented in the same environment on the DSP board. Details of implementation along with the experimental studies are described in this Chapter. The results demonstrate that the proposed self-learning ANF PSS provides good damping over a wide range of operating conditions and improves the stability margin of the system.

8.2 Physical Model of a Power System

A single-machine infinite bus power system was physically modeled in the Power Research Laboratory at the University of Calgary. An overall schematic diagram of this model is shown in Fig. 8.1. The parameters are given in the Appendix C. A three phase 3 kVA, 220 V synchronous micro-alternator driven by a dc motor is employed to model the generating unit.

The transmission line was modeled by a lumped element physical model. This simulates the performance of a 500 kV, 300 km long double circuit transmission line connected to a constant voltage bus. Consisting of six π sections, the transmission line gives a frequency response that is close to the actual transmission line response up to 500 Hz. A Time Constant Regulator (TCR) was used to change the effective field time constant of the generator in order to simulate a large generating unit. By using the TCR, the effective generator field time constant can be changed up to 10 s.

In this setup, an ABB AVR implemented on a PHSC2 Programmable Logic Controller (PLC) is used to control the terminal voltage of the alternator. Three phase ac terminal voltages and currents are stepped down, rectified and filtered with a cut-off frequency of 8 Hz to form six dc signals proportional to the terminal

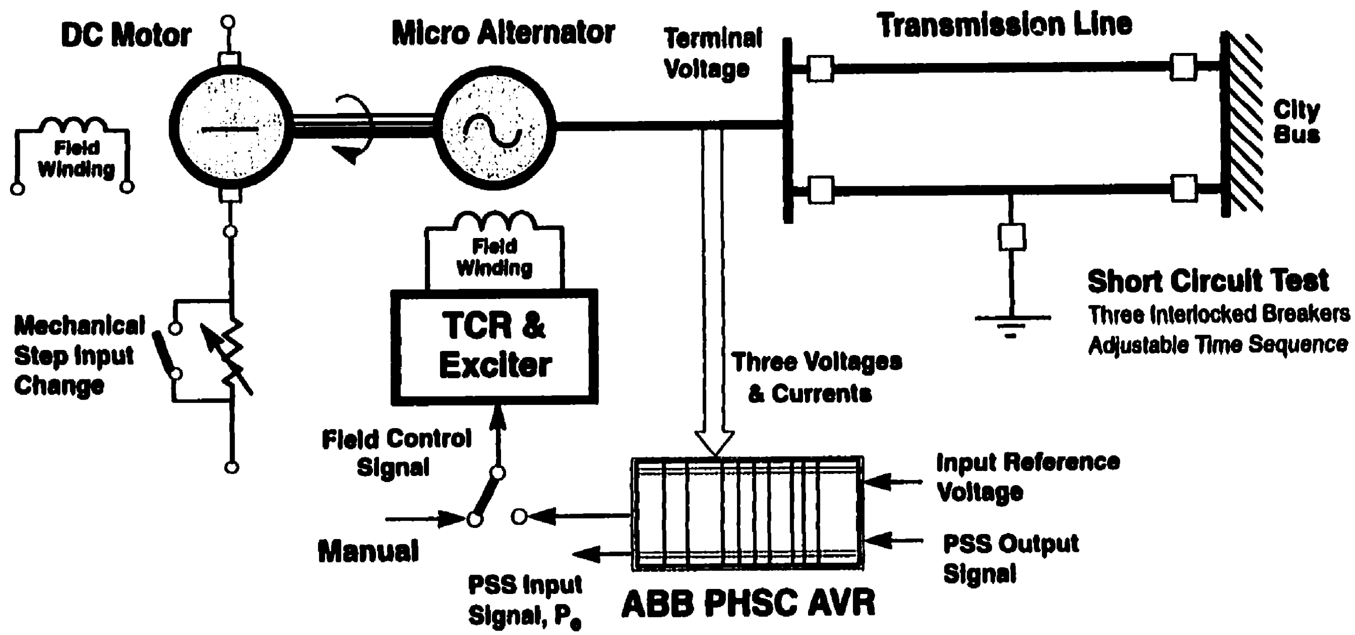

Figure 8.1: Structure of physical model of power system.

voltages and currents. As shown in Fig. 8.1, the PLC accepts these analog voltage and current signals as the inputs and provides the required field control signal as the output which is fed to the TCR. At the same time, the PLC calculates various electrical signals; among them is active power signal, P_e, applied as input signal to the stabilizer. PHSC2 PLC has been programmed using a function block programming language called FUPLA. Using a PC, FUPLA program is compiled and downloaded to the PLC.

Various disturbances can be applied to the model power system. The generator terminal voltage can be stepped up and down by changing the voltage reference setting of AVR. Similarly, by changing the armature current of the dc motor, the active power of the micro-alternator can be changed. Different types of faults can be applied on the transmission line to simulate large disturbances.

8.3 DSP Based Real-Time ANF PSS

8.3.1 Hardware Requirement

Structure of the real-time digital ANF PSS environment is shown in Fig. 8.2.

Development of the real-time digital control environment is based on a DSP board supplied by SPECTRUM Signal Processing Inc. The board contains a TMS320C30 DSP chip which is a 32-bit floating-point device with 60 ns single cycle instruction execution time. Its performance is further enhanced through its large on-chip memories, concurrent DMA controller, two external interface ports and instruction cache. Furthermore, the two 200 kHz, 16-bit analog I/O channels on board, coupled with direct access to all the serial and parallel I/O channels of DSP chip, provide the exte-

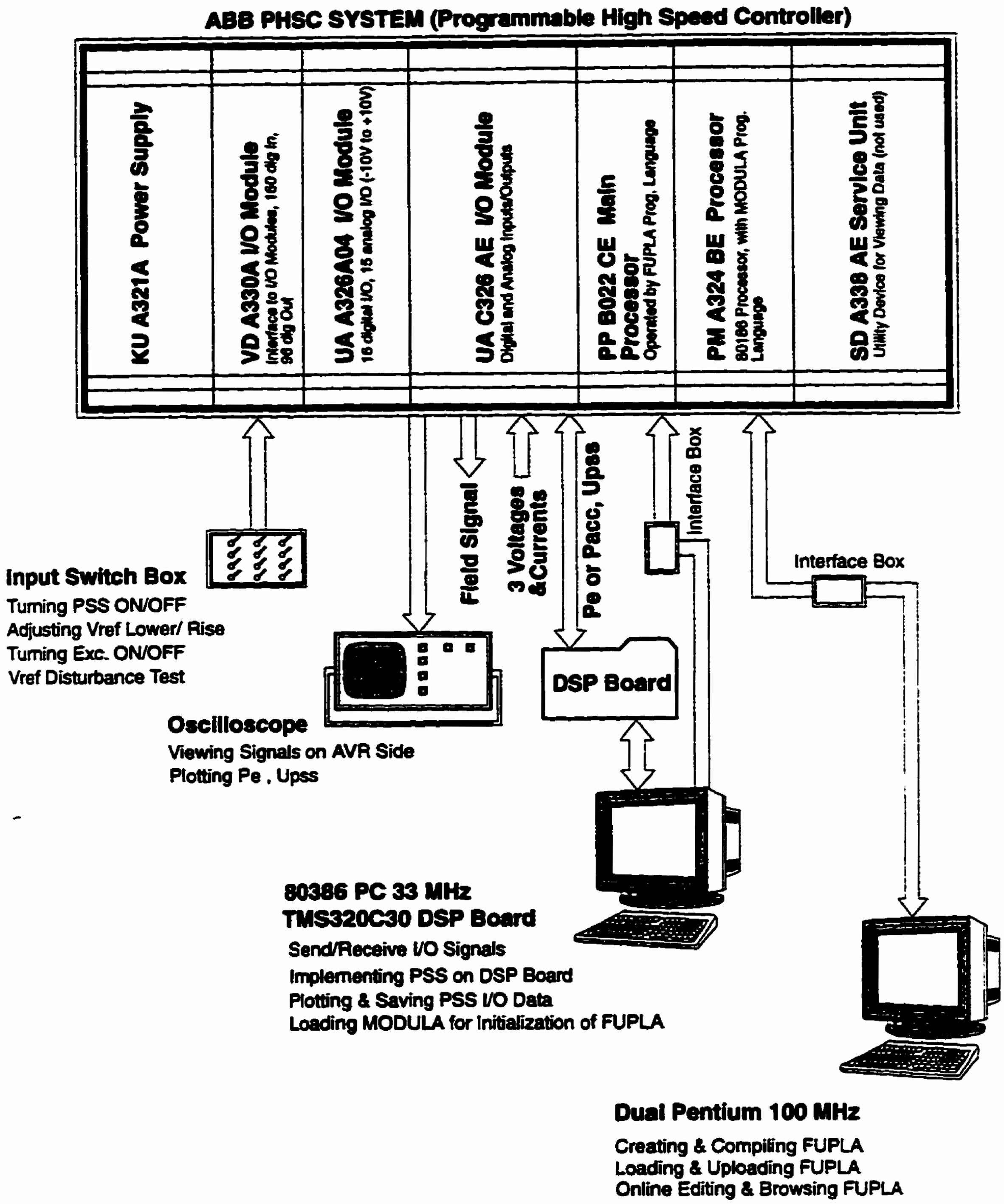

Figure 8.2: Structure of the real-time digital control environment.

rior input-output functions. The DSP board is installed on a PC with corresponding development software and debugging application program.

The output of the physical plant, P_e, is calculated by PLC and sent through analog output port. The A/D input channel of DSP board receives P_e signal, samples at a 5 ms sampling interval and stores in a buffer from which the main processor can read as needed. Then, the control signal, U_{pss}, computed by the DSP processor is placed into the output channel and converted by D/A. This output channel is connected to the analog input of the PLC. Combining the PSS signal obtained from the DSP board and the AVR signal acquired from the PLC internal calculation, PLC sends the field control signal to the TCR, thus forming a closed-loop control system.

8.3.2 Software Development

With the PC accommodating the DSP board, ANF PSS program is developed in a modular manner using C and Assembly languages. Once the application program is completed, the program is downloaded to the DSP board and started on the board.

To further enhance the power and convenience of this system, a human-machine interface program has also been designed on the same PC. The operation of this program is independent of the controller. After downloading the DSP program to the board, interface program reads the parameters of the ANF PSS from the corresponding data file, sends them to the board and waits for the DSP signal to begin the main control loop. At a 50 ms sampling interval, the interface program samples the input-output signals of the controller running on the DSP board. The input-output data is plotted on-line on screen with proper scale, while the user can choose the specific time for the beginning and end of saving the data in a file.

8.4 Initial Training of ANF PSS

The structure of ANF PSS used in experimental tests is shown in Fig. 8.3. Electric power deviation, ΔP_e, and its integral, $\int \Delta P_e$, are used as the inputs to the stabilizer. Since both signals contain DC offset value, two washout filters are implemented in software program on DSP board to diminish this DC value.

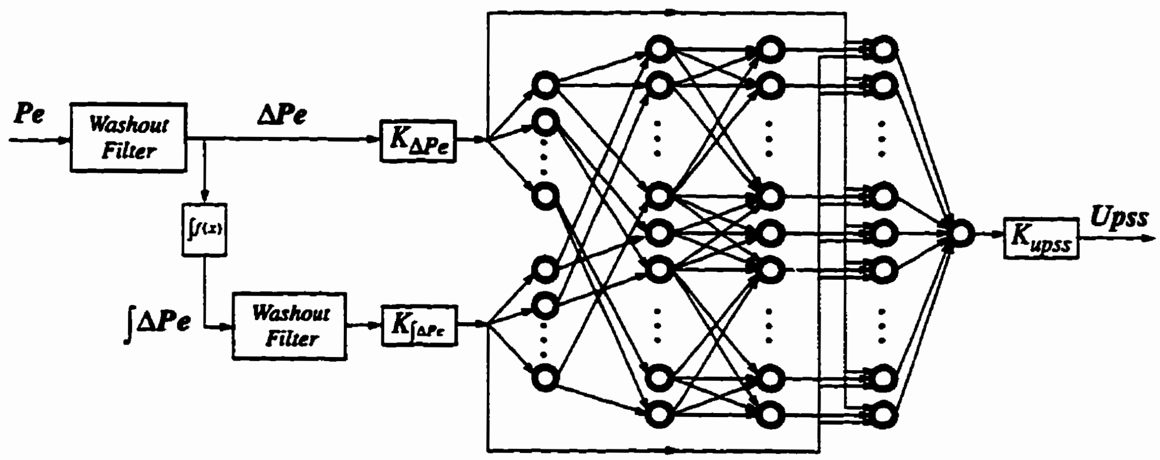

Figure 8.3: Structure of ANF PSS used in experimental tests

The ANF PSS is initially trained off-line. For this, typical disturbances under various operating conditions were applied to a power system simulation model, where the machine is modeled by a detailed 7th order model. The disturbances used were: reference voltage and infinite bus voltage disturbances in the range of -0.1 $p.u.$ to 0.1 $p.u.$, input torque variations from -0.25 $p.u.$ to 0.25 $p.u.$ and three phase fault on one of the double circuit transmission lines connected to the generating unit.

The MFs for two inputs of the controller, ΔP_e and $\int \Delta P_e$, before and after training

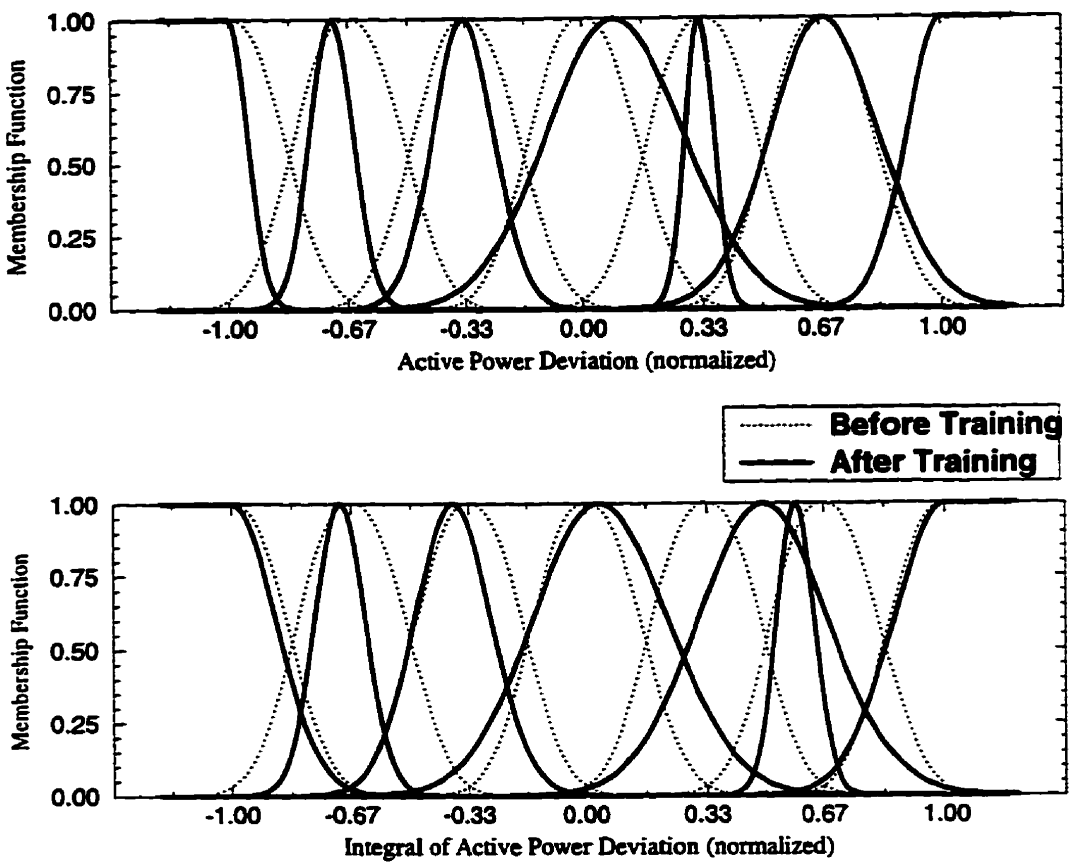

Figure 8.4: Membership functions of two inputs before and after training.

are shown in Fig. 8.4.

After a complete training procedure on a SUN Sparc Station computer, the parameters of the fuzzy controller, MFs and inference rules, were transferred back to the PC to build the ANF PSS on the DSP board.

8.5 Digital Conventional PSS

For comparison, a conventional PSS (CPSS) with the following transfer function

$$U_{pss}(s) = K_s \frac{sT_5}{1+sT_5} \frac{1+sT_1}{1+sT_2} \frac{1+sT_3}{1+sT_4} \Delta P_e(s) \tag{8.1}$$

was implemented in the same environment. Since the control setup is for the development of digital controllers, the CPSS transfer function was discretized according to the given sampling rate τ. Because the washout filter is implemented in another function block, only the lead-lag compensator needs to be discretized.

Using the bilinear transformation, $s = \frac{2}{\tau}\frac{z-1}{z+1}$, the transfer function of the CPSS in the s-domain can be transformed into the z-domain as below:

$$u_{pss}(t) = \frac{g'_0 + g'_1 z^{-1} + g'_2 z^{-2}}{f'_0 + f'_1 z^{-1} + f'_2 z^{-2}} \Delta p_e(t) \tag{8.2}$$

where the coefficients $\{g'_i\}$ and $\{f'_i\}$ are explicit functions of gain K_s and the time constants $T_1, \cdots, T_4$. The sampling rate for the digital CPSS were chosen to be $\tau = 1\ ms$.

8.6 Experimental Studies and Discussion

The active power signal P_e, computed by the ABB PLC, was sampled at 5 ms sampling interval as the input signal to the ANF PSS. The control signal was added at the sum junction after the AVR. Various disturbances under different operating conditions were applied to test the behavior of the proposed ANF PSS and the results were compared with those of the CPSS. All experimental data was collected by the human-machine interface and saved in the PC for further analysis. For easy comparison, the time axis was adjusted so that the disturbances seem to happen at the desired time point.

8.6.1 Voltage Reference Step Change

With the alternator operating at the following operating point:

$$P \;=\; 0.90\ p.u.,\ p.f. \;=\; 0.85\ lag,\ V_t \;=\; 1.10\ p.u.$$

a 10% step increase in the reference voltage was applied at 1 s and removed at 9 s. The alternator active power deviation with ANF PSS, with CPSS and without PSS are shown in Fig. 8.5. In the open loop system without any PSS, the severity of the oscillations increases as the reference voltage drops, since the system stability margin decreases as the reference voltage drops for a certain active power output. It can be noticed from Fig. 8.5 that the oscillations are effectively damped by the ANF PSS in about one cycle.

To make a reasonable comparison between the CPSS and the proposed stabilizer, the parameters of the CPSS were carefully tuned to make the CPSS produce almost the same performance as that of ANF PSS at this particular operating condition. The CPSS parameters are as follow:

$$K_s \;=\; -8.0,\ T_1 \;=\; 0.06,\ T_2 \;=\; 0.1,\ T_3 \;=\; 0.1,\ T_4 \;=\; 0.08$$

It is apparent from Fig. 8.5 that the parameters of the CPSS have been tuned properly, as it is able to enhance the performance of the system at the design operating point. Comparison of the control signals for ANF PSS and CPSS is given in Fig. 8.6.

In order to investigate the performance of the ANF PSS and CPSS, the parameters of the CPSS are kept unchanged while the operating condition is changed to:

$$P = 0.90\ p.u., \qquad p.f. = 0.95\ lead, \qquad Vt = 1.00\ p.u.$$

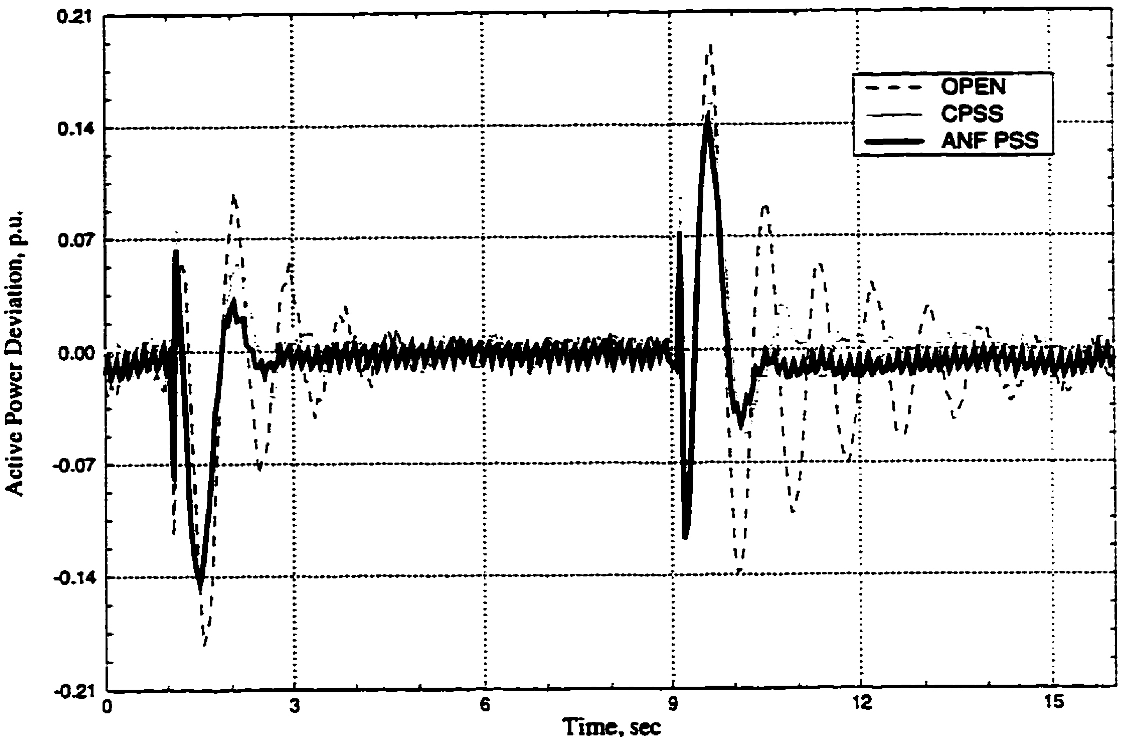

Figure 8.5: Comparison of ANF PSS and CPSS responses to a 10% step reference voltage disturbance at $P = 0.90$ $p.u.$, power factor 0.85 lag.

The same voltage reference step change of 10% is applied at 1 s and 9 s respectively. Active power deviations and the control signals for both the ANF PSS and CPSS are shown in Figs. 8.7 and 8.8 respectively. The stability margin at the leading power factor is reduced, but the ANF PSS still can yield very satisfactory results. With the ANF PSS, the system settles down within 1 s, whereas it takes longer with the CPSS. This test has shown that the ANF PSS can successfully compensate for the nonlinearity of the generating unit, i.e. the gain and phase lag changes with respect

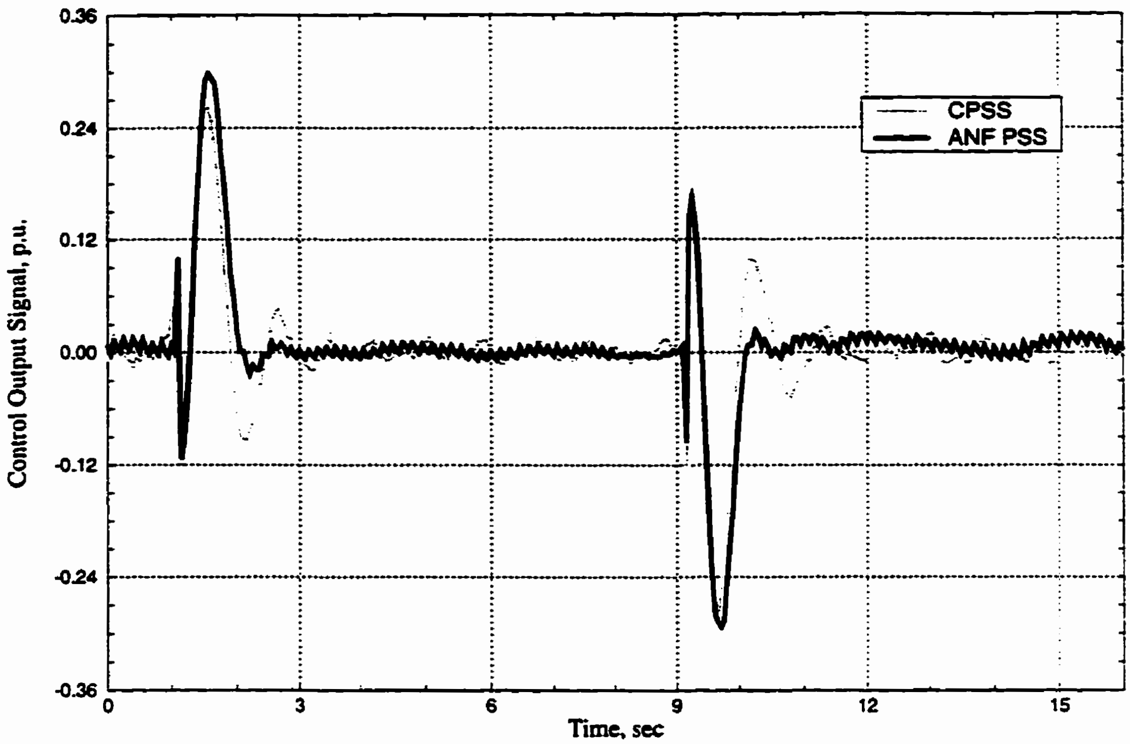

Figure 8.6: Control signals of ANF PSS and CPSS for a 10% step reference voltage disturbance at $P = 0.90$ *p.u.*, power factor 0.85 *lag*.

to the changes of the operating conditions.

8.6.2 Input Torque Reference Step Change

With the alternator operating at:

$$P = 0.90\ p.u., \qquad p.f. = 0.85\ lag, \qquad Vt = 1.10\ p.u.$$

a 0.25 *p.u.* step decrease in the input torque reference was applied at 1 *s* and removed at 9 *s*. The response is shown in Fig. 8.9. When the generator condition changes to

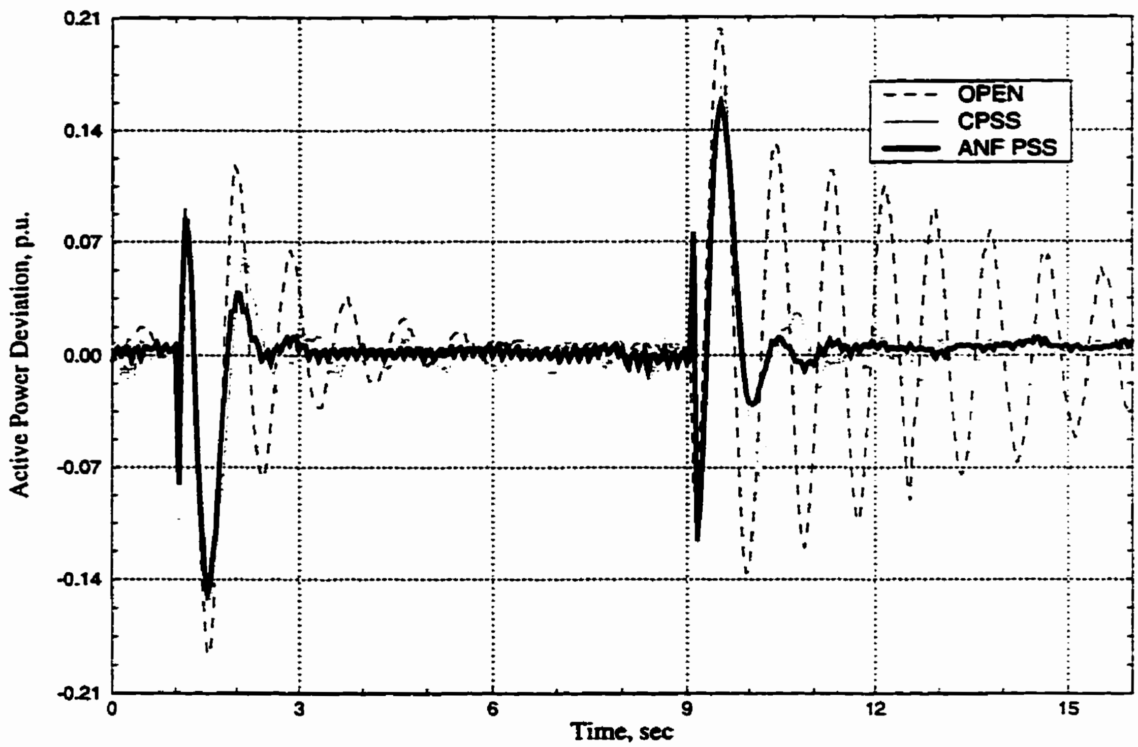

Figure 8.7: Comparison of ANF PSS and CPSS responses for a 10% step reference voltage disturbance at $P = 0.90$ $p.u.$, power factor 0.95 *lead.*

a lower operating point at 1 s, the CPSS can provide very good damping and thus there is not much difference between the ANF PSS and the CPSS. However, when a 0.25 $p.u.$ increment step change was applied to the system at 9 s, the response with the ANF PSS was consistently good.

For a leading power factor conditions, the performance with the two PSSs at operating point:

$$P = 0.90\ p.u., \qquad p.f. = 0.95\ lead, \qquad Vt = 1.00\ p.u.$$

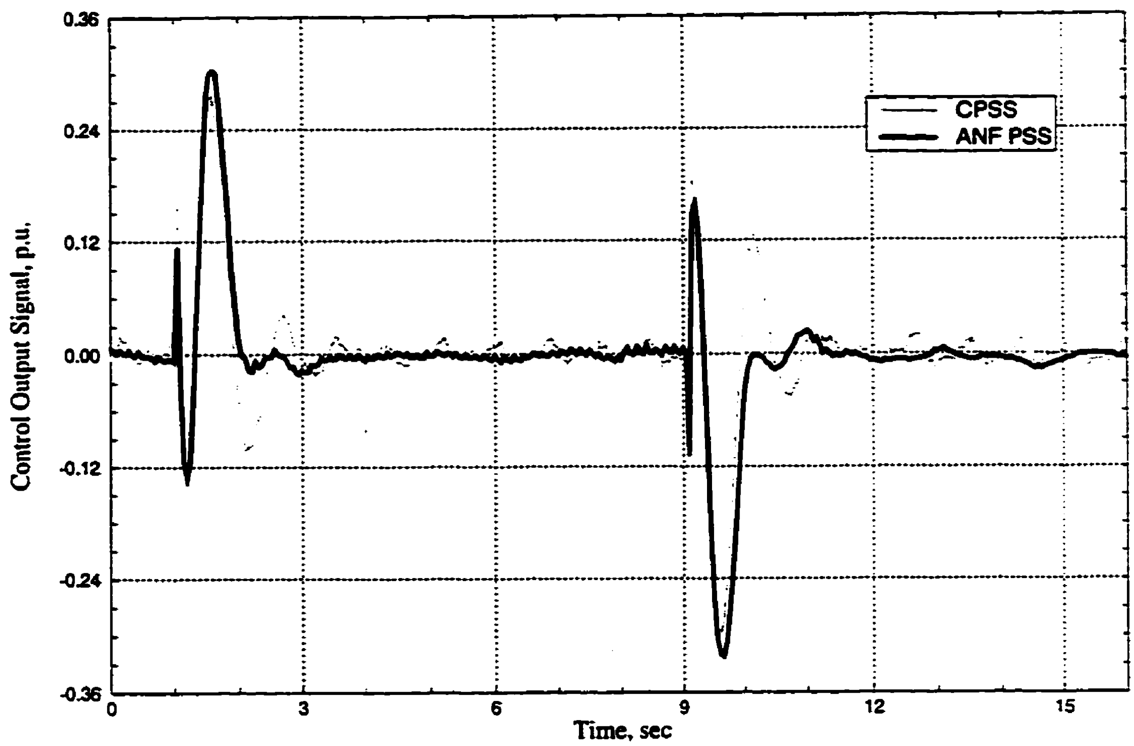

Figure 8.8: Control signals of ANF PSS and CPSS for a 10% step reference voltage disturbance at $P = 0.90$ *p.u.*, power factor 0.95 *lead*.

is shown in Fig. 8.10.

Because the ANF PSS possesses nonlinear behavior, it can provide consistent effective control signal over a wide range.

8.6.3 Three Phase to Ground Test

To investigate the performance of the ANF PSS under transient conditions, a three phase to ground fault test has been conducted at the operating point:

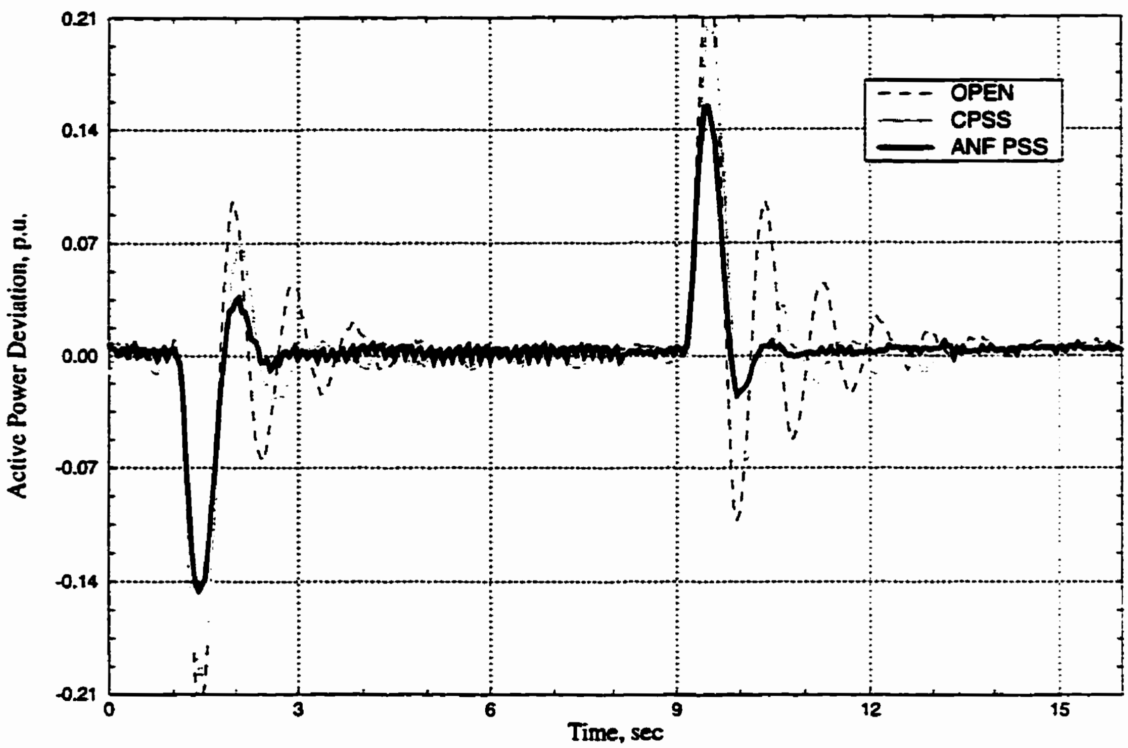

Figure 8.9: Comparison of ANF PSS and CPSS responses to a 0.25 $p.u.$ step torque disturbance at $P = 0.90$ $p.u.$, power factor 0.85 lag.

$$P = 0.50 \ p.u., \qquad p.f. = 0.9 \ lag, \qquad Vt = 1.10 \ p.u.$$

At this operating condition, with both lines in operation, a three phase to ground fault in the middle of one transmission line was applied at 3 s. The transmission line was opened, by relay action, at both ends of the line 100 ms later. An unsuccessful reclosure attempt was made after 600 ms, and the line was opened again 100 ms later due to a permanent fault.

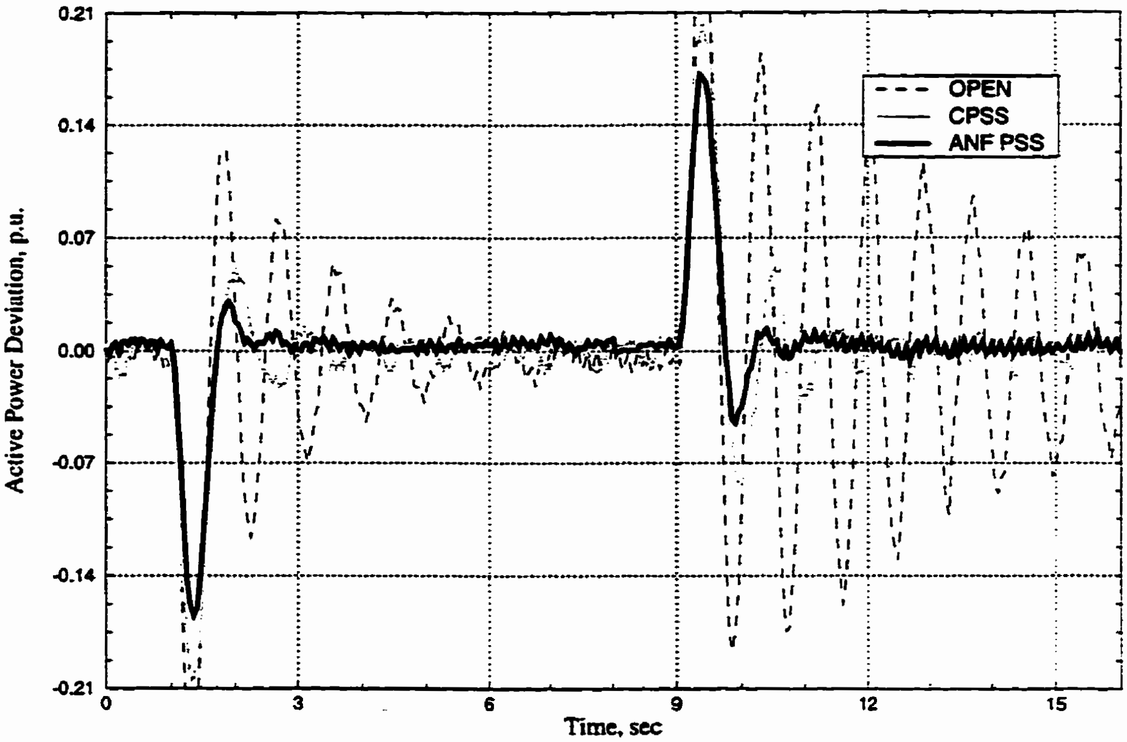

Figure 8.10: Comparison of ANF PSS and CPSS responses to a 0.25 *p.u.* step torque disturbance at $P = 0.90$ *p.u.*, power factor 0.95 *lead*.

The system response with the ANF PSS and CPSS under the above transient condition is shown in Fig. 8.11. The amplitude of the first oscillation for both controllers is the same, however, the settling time of the response with the ANF PSS is about 30% smaller than that with CPSS.

At the leading power factor operating point:

$$P = 0.90\ p.u., \qquad p.f. = 0.95\ lead, \qquad Vt = 1.00\ p.u.$$

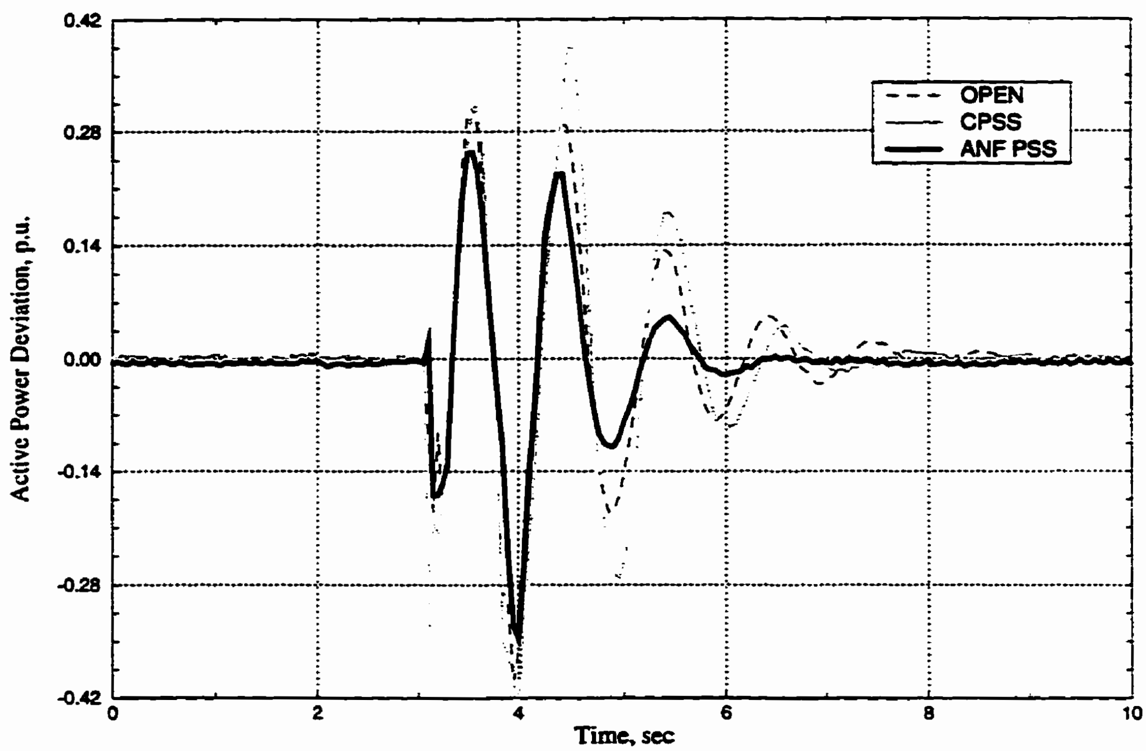

Figure 8.11: System response with ANF PSS and CPSS for three-phase short circuit test at lagging power factor.

the performance with the three phase short-circuit-fault and unsuccessful reclosure is shown in Fig. 8.12.

8.6.4 Dynamic Stability Test

The main purpose of employing power system stabilizer is to enhance the stability of the power system. With PSS in operation, the system can operate at high loads even if it is not stable without a PSS or with a poor PSS. In this test, the capability

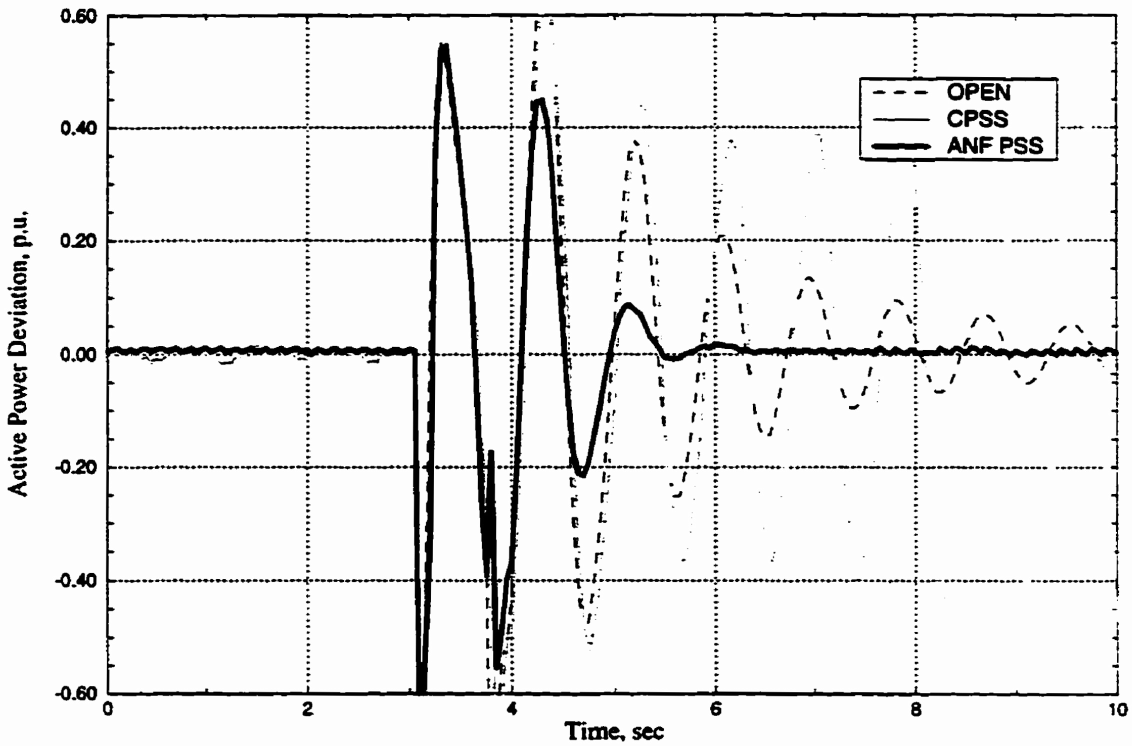

Figure 8.12: System response with ANF PSS and CPSS for three-phase short circuit test at leading power factor.

of the ANF PSS to improve the dynamic stability margin of the system is presented.

First with the ANF PSS operating, the system input torque reference was increased gradually to the level:

$$P = 1.20\ p.u., \qquad p.f. = 0.90\ lag, \qquad Vt = 1.05\ p.u.$$

at which the system was still kept stable. At 4 s, the ANF PSS was replaced by the CPSS. After replacement, the system started to oscillate without any external disturbance, which means that the CPSS is unable to maintain system stability at this load level. The ANF PSS was switched back at 17 s and the system very quickly reached the stable condition.

As shown in Fig. 8.13, the ANF PSS successfully damps the oscillations. This test demonstrates that the ANF PSS can provide a larger dynamic stability margin, thereby allowing the generating unit to operate at heavier load conditions.

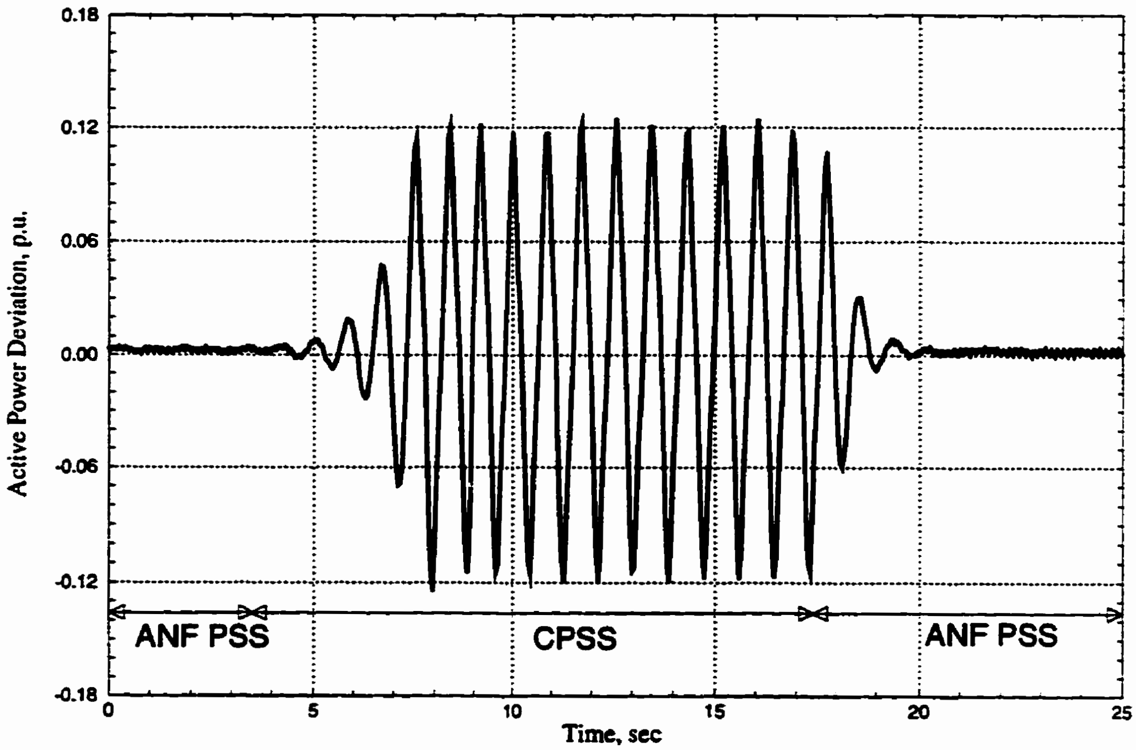

Figure 8.13: Dynamic stability improvement by ANF PSS.

8.7 Summary

Real-time implementation of the proposed ANF PSS and experimental studies on a physical model power system are presented in this Chapter. Active power deviation and its integral are employed as the inputs to the ANF PSS. Training procedure for the proposed stabilizer is based on a self-learning technique; i.e. independent of another PSS. The experimental results are discussed and compared with a digital type conventional PSS. The results demonstrate that both the CPSS and the proposed ANF PSS are effective in improving the dynamic performance of the system at design operating point. However, when operating point deviates from its design point, the performance of the CPSS deteriorates. This is due to the nonlinear characteristic of the generating unit. Also, the ANF PSS can provide a larger dynamic stability margin than the CPSS.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

As discussed in Chapter 1, Power System Stabilizers (PSSs) have proven very effective in enhancing the stability of power systems. Numerous theoretical studies and experimental tests have been conducted to better understand the behavior of the PSS and to make them more applicable in practice. Different types of PSSs have been investigated, and their advantages and disadvantages have become more and more clear. Based on these studies criteria have been developed to help the designer to choose the most suitable configuration for a particular application.

Conventional PSS (CPSS) has been successfully applied to power industry in many cases. However, because of its inherent linear characteristic, it faces many serious problems. The stabilizer should be able to catch the non-linearity of the system and produce the same performance for different operating conditions and different types of disturbances.

This dissertation is devoted to the development of an adaptive fuzzy logic power system stabilizer. It has made systematic contributions to all three stages of developing such a stabilizer –theoretical development, simulation studies and experimental tests.

After studying and comparing fuzzy logic and neural network control strategies, an adaptive fuzzy logic based power system stabilizer has been proposed [86][87].

It combines the advantages of both control strategies, avoids their drawbacks, and connects these two seemingly different algorithms together. By using the adaptive fuzzy logic PSS, the tuning problem of fuzzy controller is removed and black-box characteristic of neural network controller is significantly improved. In this way, it simplifies the tuning procedure during commissioning and thus makes it more suitable for practical applications.

The proposed adaptive fuzzy PSS requires another existing controller (desired controller) to adjust its parameters to yield the same control performance. Adaptive self-optimizing pole-shifting PSS (APSS) is selected to be the desired controller. A trained adaptive fuzzy controller can produce quick control signal and overcome the disadvantage of long computing time of the desired controller. The computing time of APSS increases as the identification and control algorithms become more complicated. By training an adaptive fuzzy PSS to simulate the function of an APSS, the new PSS combines the good control effect of APSS and quick response of ANN, and thus improves the performance of the power system.

In the next stage of designing adaptive fuzzy PSS, a self-learning approach is used to train the controller directly from the generating unit output [88][91]. This approach is independent of other PSSs. An adaptive fuzzy identifier is first trained to identify the dynamic of the unknown plant, and then this identifier is utilized to back-propagate the error at the generating unit output to the output of the controller. The parameters of the controller are updated after a certain elapsed time to minimize the difference between the plant trajectory output and the desired trajectory.

Besides the tuning problem of fuzzy logic controllers, the selection of the number of inference rules and membership functions is done by trial and error. By increasing the number of input and outputs, this problem becomes more crucial. To automate the process of finding the optimum structure for adaptive fuzzy PSS, Genetic Algorithm as a global optimization technique is employed [89]. By applying both GA and back-propagation techniques, the number and the shape of membership functions are determined by GA and the consequent part of inference rules are specified by back-propagation algorithm.

The adaptive fuzzy PSS is built and tested in the single-machine infinite-bus environment by computer simulation. In each of three mentioned approaches, the adaptive fuzzy PSS is trained in the full working range of the generating unit with a wide range of disturbances. Simulation results have demonstrated that the proposed adaptive PSS can adjust its parameters to produce a control signal that can provide enough damping to different disturbances.

The effectiveness of the adaptive fuzzy PSS to damp multi-mode oscillations in a multi-machine environment is also verified in this dissertation [92]. Test results show that each adaptive fuzzy PSS can damp the specific mode of oscillation introduced mainly by the generating unit on which it is applied. Several adaptive fuzzy PSSs working together can damp both local and inter-area mode oscillations. There are no coordination conflicts with the other types of PSSs.

These results have shown that the adaptive fuzzy PSS has many promising features that the conventional PSS lacks. This makes it a strong candidates to replace the conventional PSS in future.

Next stage in the development process is the implementation of the device. If it is considered that the simulation studies prove the proposed control algorithm theoretically, the implementation tests prove the proposed control algorithm practically.

Implementation is a critical step towards its practical application. By utilizing a PHSC2 Programmable Logic Controller as AVR and a TMS320C30 Digital Signal Processor as PSS, a real-time digital control environment has been established to implement adaptive fuzzy PSS [93]. For comparison, a digital type conventional PSS has also been implemented in this environment and tested under the same conditions. Experimental tests have produced results consistent with the simulation studies, proving the capability of the proposed adaptive fuzzy PSS.

9.2 Future Work

Research on fuzzy logic, neural network and genetic algorithm in control systems has advanced rapidly in recent years. Since the application of these techniques in power engineering is a new area, much work needs to be conducted in order to put them into practical use.

Based on the work of this dissertation, the followings are recommended as further research topics:

- For training the adaptive fuzzy controller, off-line learning method is used. The next immediate step seems to be to investigate the possibility of applying on-line learning method to track time varying stochastic power systems. However, there are many serious aspects that need to be investigated before on-line method can be put into use. Stability of the closed loop system is the major concern. Since the controller parameters are updated each sampling interval, without having an efficient criteria to limit the parameter update, on-line learning method may lead the system to unstable region.

- Adaptive fuzzy controller, in general, can be used as a multi-input multi-output controller without facing the tuning problem. Only two inputs and one output are considered for adaptive fuzzy PSS. Increasing the number of inputs and outputs could be very interesting. By using the generating unit terminal voltage, V_t, as another input to the controller, the combination of AVR loop and PSS loop will be achieved, hence both power oscillations, ΔP_e, and terminal voltage deviation, ΔV_t, can be controlled at the same time and in one control loop.

  Also, integrating both the excitation and the governor control loops and considering the interaction between them are worth looking into.

- Although many theoretical stability criterias are proposed for fuzzy and neural network based control systems, still a guaranteed reliability to handle all unpredicted situations is of greatest need for these control schemes.

- Lab implementation of the ANF PSS was based on a sequential computation method using a DSP board. To reduce the computing time further, it is suggested that the ANF PSS be implemented on a commercially available fuzzy logic chip.

Bibliography

[1] H. A. M. Moussa and Y. N. Yu, "Dynamic interaction of multi-machine power system and excitation control", *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-93, pp. 2211–2218, 1974.

[2] V. A. Venikov and V. A. Strove, "Power system stability as affected by automatic control of generators-some methods of analysis and synthesis", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-90, pp. 2483–2487, 1971.

[3] P. M. Anderson and A. A. Fouad, ", *Power System Control and Stability*, Iowa State University Press, Ames, Iowa, 1977.

[4] E. W. Kimbark, ", *Power System Stability*, vol. 1, Wiley, 1948.

[5] Y. Yu, ", *Electric Power System Dynamics*, Academic Press, 1983.

[6] M. K. El-Sherbiny and A. A. Fouad, "The dynamic interaction of closely-coupled synchronous generators", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-90, pp. 441–447, 1971.

[7] M. K. El-Sherbiny and D. M. Mehta, "Dynamic system stability, part I-investigation of the effect of different loading and excitation systems", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-92, pp. 1538–1546, 1973.

[8] F.P. de Mello and T.F. Laskowski, "Concepts of power system dynamic stability", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-94, pp. 827–833, 1975.

[9] W. A. Wittelstdat, "Four methods of power system damping", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-87, pp. 1323–1329, May 1968.

[10] O. J. M. Smith, "Power system transient control by capacitor switching", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-88, pp. 28–35, Jan. 1969.

[11] E. W. Kimbark, "Improvement of power system stability by changes in the network", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-88, pp. 773–781, May 1969.

[12] P. K. Dash, B. Puthal, O. P. Malik, and G. S. Hope, "Transient stability and optimal control of parallel ac-dc power system", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-95, pp. 811–820, March 1976.

[13] C. Concordia and F. P. de Mello, "Concepts of synchronous machine satability as affected by excitation control", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-88, pp. 316–329, April 1969.

[14] F. R. Schlief, H. D. Hunkins, G. E. Martin, and E. E Hattan, "Excitation control to improve powerline stability", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-87, pp. 1426–1434, Jun. 1968.

[15] J. P. Bayne, P. Kundur, and W. Watson, "Static excitation control to improve transient stability", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-94, pp. 1141–1146, 1975.

[16] W. Watson and G. Manchur, "Experience with supplementary damping signal for generator static excitation systems", *IEEE Trans. Power Apparatus and*

Systems, vol. PAS-92, pp. 199–211, 1973.

[17] E.W. Larsen and D.A. Swann, "Applying power system stabilizer: Parts 1-3", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-100, no. 6, pp. 3017–3046, 1981.

[18] F. P. deMello, L. N. Hannett, D. W. Parkinson, and J. S. Czuba, "A power system stabilizer design using digital control", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-101, pp. 2860–2866, 1982.

[19] P. Kundur, D. C. Lee, and H. M. Zein El-Din, "Power system stabilizers for thermal units: analytical techniques and on-site validation", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-100, pp. 81–95, Jan. 1981.

[20] J. P. Bayne, D. C. Lee, and W. Watson, "A power system stabilizer for thermal units based on deviation of accelerating power", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-96, pp. 1777–1783, 1977.

[21] R. G. Farmer, "State-of-the-art technique for system stabilizer tuning", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-102, pp. 699–709, 1983.

[22] A. Doi and S. Abe, "Coordination synthesis of power system stabilizer in multi-machine power system", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-103, pp. 1473–1479, June 1984.

[23] W. C. Chan and Y. Y. Hsu, "An optimal variable structure stabilizer for power system stabilizer", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-102, pp. 1738–1746, 1983.

[24] K. J. Astrom and B. Wittenmark, "On self-tuning regulators", *Automatica*, vol. 9, pp. 185–199, 1973.

[25] I. D. Landau, "A survey of model reference adaptive techniques: theory and application", *Automatica*, vol. 10, pp. 353–379, 1974.

[26] A. Gosh, G. Ledwich, O.P. Malik, and G.S. Hope, "Power system stabilizer based on adaptive control techniques", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-103, pp. 1983–1986, 1984.

[27] S.J. Cheng, O.P. Malik, and G.S. Hope, "Damping of multi-mode oscillations in power system using a dual-rate adaptive stabilizer", *IEEE Trans. on Power Systems*, vol. PWRS-3, no. 1, pp. 101–108, 1988.

[28] G.P. Chen, O.P. Malik, G.S. Hope, Y.H. Qin, and G.Y.Xu, "An adaptive power system stabilizer based on the self-optimizing pole shifting control strategy", *IEEE Trans. on Energy Conversion*, vol. 8, no. 1, pp. 71–77, 1993.

[29] G.P. Chen and O.P. Malik, "Tracking constrained adaptive power system stabilizer", *IEE Proceedings, Generation, Transmission and Distribution*, vol. 142, no. 2, pp. 149–156, March 1995.

[30] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representation by error propagation", *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp. 318–362, 1986.

[31] S. Haykin, "Neural network, a comprehensive foundation", *Multi layer perceptron*, pp. 138–229, 1994, IEEE Press.

[32] Y. Zhang, G.P. Chen, O.P. Malik, and G.S. Hope, "An artificial neural network based adaptive power system stabilizer", *IEEE Trans. on Energy Conversion*, vol. 8, no. 1, pp. 71–77, 1993.

[33] Y. Zhang, O.P. Malik, G.S. Hope, and G.P. Chen, "Application of an inverse input-output mapped ANN as a power system stabilizer", *IEEE Trans. on Energy Conversion*, vol. 9, no. 3, pp. 433–441, 1994.

[34] Y. Zhang, O.P. Malik, and G.P. Chen, "Artificial neural network power system stabilizer in multi-machine power system environment", *IEEE Trans. on Energy Conversion*, vol. 10, no. 1, pp. 147–153, 1995.

[35] L.A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Trans. on System, Man and Cybernetics*, vol. SMC-3, no. 1, pp. 28–44, 1973.

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

[37] L.A. Zadeh et al, "Calculus of fuzzy restriction in fuzzy sets and their application to cognitive and decision process", *Academic press*, pp. 1–40, 1975.

[38] P.J. King and E.H. Mamdani, "The application of fuzzy control systems to industrial processes", *Automatica*, vol. 13, no. 3, pp. 235–242, 1977.

[39] C.C. Lee, "Fuzzy logic in control system: fuzzy logic controller part I and II", *IEEE Trans. on Computers*, vol. 20, no. 2, pp. 404–435, 1990.

[40] T. Hiyama and T. Sameshima, "Fuzzy logic control scheme for on-line stabilization of multi-machine power system", *Fuzzy Sets and Systems*, vol. 39, pp. 181–194, 1991.

[41] M.A.M Hassan, O.P. Malik, and G.S. Hope, "A fuzzy logic based stabilizer for a synchronous machine", *IEEE Trans. on Energy Conversion*, vol. 6, no. 3, pp. 407–413, 1991.

[42] D. Mcheill and P. Freiberg, "The discovery of a revolutionary computer technology and how it is changing our world", *Simon and Schuster*, 1993.

[43] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", *International Journal ManMachine Studies*, vol. 1, no. 7, pp. 1–13, 1975.

[44] L. P. Holmblad and J. J. Ostergaard, "Control of a cement kiln by fuzzy logic", *in Fuzzy Information and Decision Processes, M. M. Gupta and E. Sanchez (eds), North-Holland, Amsterdam*, 1982.

[45] M. Sugeno and M. Nishida, "Fuzzy control of model car", *Fuzzy Sets and Systems*, , no. 16, pp. 103–113, 1985.

[46] M. Sugeno and K. Murakami, "An experimental study on fuzzy parking control using a model car", *in Industrial Applications of Fuzzy Control, M. Sugeno (ed.), Elsevier Science Publishers B.V.*, pp. 125–137, 1985.

[47] S. Yasunobu, S. Miyamoto, and H. Ihara, "Fuzzy control for automatic train operation system", *Proc. 4th IFAC/IFIP/IFORS Int. Congress on Control in*

Transportation Systems, Baden-Baden, April 1983.

[48] S. Yasunobu and S. Miyamoto, "Automatic train operation system by predictive fuzzy control", *in Industrial Applications of Fuzzy Control, M. Sugeno (ed.), Elsevier Science Publishers B.V.*, pp. 1–18, 1985.

[49] B. Kosko, "Neural networks and fuzzy systems: A dynamic approach to macine intelligence", *Prentice-Hall*, 1992.

[50] M. M. Gupt, R. K. Ragada, and R. R. Yager, "Advances in fuzzy set theory and applications", *North Holland*, 1979.

[51] M. Sugeno, "Industrial applications of fuzzy control", *Elsevier Science Publisher B.V. North-Holland*, 1985.

[52] Y. Tsukamoto, "An approach to fuzzy reasoning method", *in Advances in Fuzzy Set Theory and Applications, M. M. Gupta, R. K. Ragade and R. R. Yager. North-Holland, Amsterdam*, 1979.

[53] N. Wiener, "Cybernetics: or control and communications in the animal and the machine", *MIT Press, Cambridge, MA*, 1948.

[54] B. Widrow, "The original adaptive neural net brrom-balancer", *Proceeding, IEEE Int. Symposiom on Circuits and Systems*, pp. 351–357, 1987.

[55] H. Tolle and E. Ersu, "Neurocontrol: Learning control systems inspired by neural architectures and humann problem solving", *Lecture Notes in Control and Information SCiences, Springer-Verlag, Berlin*, , no. 172, 1992.

[56] W. T. Miller, R.S. Suttin, and P. J. Werbos, "Neural networks for control", *MIT Press, Cambridge, MA*, 1990.

[57] S. Chen and S. A. Billings, "Neural networks for non-linear dynamic system modelling and identification", *in Advances in Intelligent Control, C.J. Harris, Taylor and Francis, London, Chapter 4*, 1994.

[58] Y. H. Pao, S.M. Phillips, and D. J. Sobajic, "Neural-net computing and intelligent control of systems", *in Advances in Intelligent Control, C.J. Harris, Taylor and Francis, London, Chapter 3*, 1994.

[59] J. Moody, "Fast learning in multi-resolution hierarchies", *in Advances in Neural Information Processing System, D. S. Touretzky, Moragn Kaufmann, San Mateo, CA*, pp. 29–39, 1989.

[60] R. J. williams, "Adaptive state representation and estimation using recurrent connectionist networks", *in Neural Networtks for Control, W. T. Miller, R. S. Sutton, P. J. Werbose, MIT press, Cambridge, MA, Cahpter 4*, pp. 97–114, 1990.

[61] M. Brown and C. Harris, "Neurofuzzy adaptive modelling and control", *Prentice Hall*, 1994.

[62] B. Widrow and M. E. Hoff, "Adaptive switching circuits", *IRE WESCON Convention Record*, pp. 96–104, 1960.

[63] P. J. Werbos, "Backpropagation and neurocontrol: A review and prospectus", *International Joint Conference on Neural Networks, Washington, DC*, vol. 1,

pp. 209–216, 1989.

[64] J. M. Mendel and R. W. McLaren, "Reinforcement learning control and pattern recognition systems", *in Adaptive, learning and Pattern Recognition systems Theory and applications, New York, Academic Press*, pp. 287–318, 1970.

[65] T. Kohonen, "Self-organization and associative memory", *Berlin: Springer-Verlag*, 1989.

[66] G. A. Carpenter and S. Grossberg, "The art of adaptive pattern recognition by a self-organizing neural network", *IEEE Computer*, pp. 77–88, March 1988.

[67] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model", *Neural Networks*, vol. 1, pp. 339–356, 1988.

[68] J.H. Holland, "Outline for a logical theory of adaptive systems", *Journal of ACM*, vol. 3, pp. 297–314, 1962.

[69] J.H. Holland, "Adaptation in natural and artificial systems", *The University Michigan Press*, 1975.

[70] K. A. DeJong, "An analysis of the behavior of a class of genetic adaptive systems", *Ph.D. dissertation (CCS), Univ. Mich., Ann Arbor,MI*, 1975.

[71] J. M. Fitzpatrick, J. J. Grefenstette, and D. Van Gucht, "Image registration by genetic search", *in Proceeding IEEE Southeastcon '84*, pp. 460–464, 1984.

[72] R. Das and D. E. Goldberg, "Discrete-time parameter estimation with genetic algorithms", *preprints from the Proc. 19th annual Pittsburgh Conference on Modeling and Simulation*, 1988.

[73] D. M. Etter, M. J. Hicks, and K. H. Cho, "Recursive adaptive filter design using an adaptive genetic algorithm", *Proc. IEEE International Conference Acoustic, Speech, Signal Processing*, vol. 2, pp. 635–638, 1982.

[74] D. E. Goldberg, "Genetic algorithm in search, optimization and machine learning", *Reading, MA: Addison-Wesley*, 1989.

[75] J.G. Jang, "ANFIS: Adaptive-network-based fuzzy inference system", *IEEE Trans. on System, Man and Cybernetics*, vol. 23, no. 3, pp. 665–684, 1993.

[76] C. Lin and C.S.G. Lee, "Neural-network-based fuzzy logic control and decision system", *IEEE Trans. on Computers*, vol. 40, no. 12, pp. 1320–1336, 1991.

[77] J.G. Jang, "Self-learning fuzzy controllers based on temproal backpropagation", *IEEE Trans. on Neural Networks*, vol. 3, no. 5, pp. 714–723, 1992.

[78] H. Nomura, I. Hayashi, and N. Wakami, "A self-tuning method of fuzzy reasoning by genetic algorithm", *Proc. of the 1992 Int. Fuzzy Systems and Intelligent Control Conference*, pp. 236–245, 1992.

[79] D. Psaltis, A. Sideris, and A. Yamamura, "Neural controllers", *Proc. of 1st International Conference on Neural Networks, San Diego, USA*, vol. 4, pp. 551–558, 1987.

[80] A. Soquet M. Saerens, "A neural controller", *Proc. of 1st International Conference on Neural Networks, San Diego, USA*, vol. 4, pp. 211–215, 1981.

[81] D.H. Nguyen and B. Widrow, "Neural networks for self-learning control systems", *IEEE Control Systems Magazine*, pp. 18–23, Apr. 1990.

[82] J. Hertz, A. Krogh, and R.G. Palmer, "Introduction to the theory of neural computation", *Addison-Wesley, Redwood City*, 1991.

[83] K. Balakrishnan and V. Honavar, "Evolutionary design of neural architectures", *Artificial Intelligence Research Group, Iowa State University, Report no. CS TR 95-01*, 1995.

[84] Xin Yao, "A review of evolutionary artificial neural networks", *International Journal of Intelligent Systems*, to be appeared.

[85] IEEE working group on special stability controls Power System Engineering Committee, "Bibliography on the application of discrete supplementary controls to improve power system stability", *IEEE Trans. on Power Systems*, vol. PWRS-100, no. 9, pp. 474–485, 1987.

[86] A. Hariri and O.P. Malik, "Adaptive-network-based fuzzy logic power system stabilizer", *IEEE WESCANEX '95 Proceeding*, vol. 1, pp. 111–116, 1995.

[87] A. Hariri and O.P. Malik, "A fuzzy logic based power system stabilizer with learning ability", *IEEE Transactions on Energy Conversion*, vol. 11, no. 4, pp. 721–727, Dec. 1996.

[88] A. Hariri and O.P. Malik, "Self-learning adaptive-network-based fuzzy logic power system stabilizer", *Proceeding, IEEE Int. Conf. on Intelligent Systems Applications to Power Systems, Orlando, Fl., Jan.28 - Feb.2*, vol. 1, pp. 299–303, 1996.

[89] A. Hariri and O.P. Malik, "Fuzzy logic power system stabilizer based on genetically optimized adaptive-network", *Presented in IEEE PES Winter Meeting Feb. 1997, Paper no. 97-329*, 1997.

[90] IEEE Excitation System Model Working Group, "Excitation system models for power system stability studies", *Draft 15 for ANSI/IEEE Standard*, p. 421.5/D15, 1990.

[91] A. Hariri and O.P. Malik, "A self-learning fuzzy stabilizer for a synchronous machine", *International Journal of Electrical Power and Energy Systems, in Press.*

[92] A. Hariri and O.P. Malik, "A self-learning adaptive-network-based fuzzy logic pss in multi-machine system", *IEEE Power Engineering Society,Summer Meeting 1997, Submitted, Jan 1997.*

[93] A. Hariri and O.P. Malik, "Experimental studies with a self-learning adaptive-network-based fuzzy logic power system stabilizer", *IEEE Power Engineering Society,Summer Meeting 1997, Submitted, Jan 1997.*

Appendix A

Single-machine Power System

A.1 The generating unit is modeled by seven first order differential equations given below:

$$\dot{\delta} = \omega_0 \omega \tag{A.1}$$

$$\dot{\omega} = \frac{1}{2H}(T_m + g + K_d\dot{\delta} - T_e) \tag{A.2}$$

$$\dot{\lambda_d} = e_d + r_a i_d + \omega_0(\omega + 1)\lambda_q \tag{A.3}$$

$$\dot{\lambda_q} = e_q + r_a i_q + \omega_0(\omega + 1)\lambda_d \tag{A.4}$$

$$\dot{\lambda_f} = e_f - r_f i_f \tag{A.5}$$

$$\dot{\lambda_{kd}} = -r_{kd} i_{kd} \tag{A.6}$$

$$\dot{\lambda_{kq}} = -r_{kq} i_{kq} \tag{A.7}$$

A.2 The AVR and excitor model used in the system is from the IEEE standard P421.5,1992, Type ST1A as shown in Fig. A.1.

A.3 The governor used in the system has the transfer function:

$$g = \left[a + \frac{b}{1 + sT_g}\right]\dot{\delta}$$

A.4 The conventional PSS has the following transfer function:

$$U_{pss}(s) = K_s \frac{sT_5}{1 + sT_5} \frac{1 + sT_1}{1 + sT_2} \frac{1 + sT_3}{1 + sT_4} \frac{1}{1 + sT_6} \dot{\delta} \tag{A.8}$$

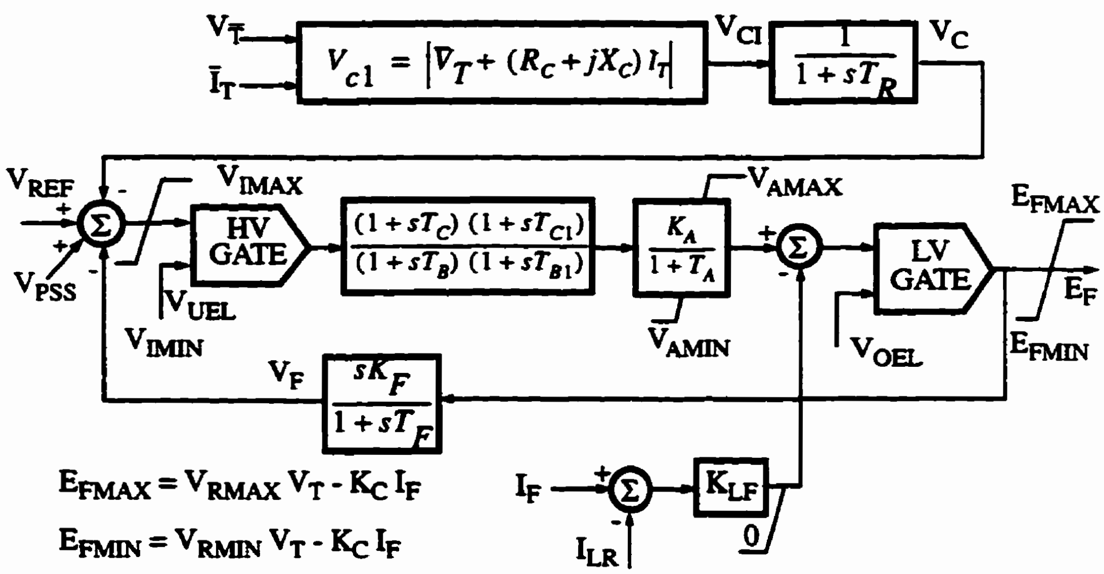

Figure A.1: AVR and excitor model Type ST1A, IEEE standard P421.5,1992.

A.5 Parameters used in the simulation study are given below:

$r_a = 0.007$	$r_{kd} = 0.023$	$r_{kq} = 0.023$
$r_f = 0.00089$	$H = 3.46$	$K_d = -\ 0.027$
$x_d = 1.24$	$x_{kd} = 1.15$	$x_{md} = 1.126$
$x_q = 0.743$	$x_{kq} = 0.652$	$x_{mq} = 0.626$
$x_f = 1.33$	$r_t = 0.05$	$x_t = 0.6$
$K_A = 200$	$T_A = 0.01$	$K_F = 0.05$
$R_C = 0.0$	$X_C = 0.0$	$T_C = 0.1$
$T_B = 0.03$	$T_{C1} = 0.0$	$T_{B1} = 0.0$
$T_F = 1.0$	$V_{IMIN} =$ -999	$V_{IMAX} = 999$
$V_{AMIN} =$ -999	$V_{AMAX} = 999$	$V_{RMIN} =$ -6.7
$V_{RMAX} = 7.8$	$V_{UEL} =$ -999	$V_{OEL} = 999$
$a = -0.001328$	$b = -\ 0.17$	$T_g = 0.25$
$T_1 = 0.1$	$T_2 = 0.02$	$T_3 = 0.1$
$T_4 = 0.02$	$T_5 = 1.65$	$T_6 = 0.005$
$K_s = 0.05$	$U_{pssMIN} =$ -0.1	$U_{pssMAX} = 0.1$

All resistances and reactances are in per-unit and time constants in seconds.

Appendix B

Multi-machine Power System

B.1 The generating unit is modeled by five first order differential equations given below:

$$\dot{\delta} = \omega_0 \omega \tag{B.1}$$

$$\dot{\omega} = \frac{1}{2H}(T_m + g + K_d\dot{\delta} - T_e) \tag{B.2}$$

$$T'_{do}\dot{e}'_q = e_f - (x_d - x'_d)i_d - e'_q \tag{B.3}$$

$$T''_{do}\dot{e}''_q = [e'_q - (x'_d - x''_d)i_d - e''_q] + T''_{do}\dot{e}'_q \tag{B.4}$$

$$T''_{qo}\dot{e}''_d = (x_q - x''_q)i_q - e''_d \tag{B.5}$$

B.2 Generator parameters:

	G_1	G_2	G_3	G_4	G_5
X_d	0.1026	0.1026	1.0260	0.1026	1.0260
X_q	0.0658	0.0658	0.0658	0.0658	0.0658
X'_d	0.0339	0.0339	0.0339	0.0339	0.0339
X''_d	0.0269	0.0269	0.0269	0.0269	0.0269
X''_q	0.0335	0.0335	0.0335	0.0335	0.0335
T'_{do}	5.6700	5.6700	5.6700	5.6700	5.6700
T''_{do}	0.6140	0.6140	0.6140	0.6140	0.6140
T''_{qo}	0.7230	0.7230	0.7230	0.7230	0.7230
H	80.000	80.000	10.000	80.000	10.000

B.3 AVR and simplified ST1A exciter parameters:

	G_1	G_2	G_3	G_4	G_5
T_r	0.0400	0.0400	0.0400	0.0400	0.0400
K_a	190.00	190.00	190.00	190.00	190.00
K_c	0.0800	0.0800	0.0800	0.0800	0.0800
T_a	10.000	10.000	10.000	10.000	10.000
T_c	1.0000	1.0000	1.0000	1.0000	1.0000

B.4 Governor parameters:

	G_1	G_2	G_3	G_4	G_5
T_g	0.2500	0.2500	0.2500	0.2500	0.2500
a	-0.0001	-0.0001	-0.0013	-0.0001	-0.0013
b	-0.0150	-0.0150	-0.1700	-0.0150	-0.1700

B.5 Transmission line parameters:

$Bus\#$	R_t	X_t	$B_t/2$
1-7	0.00435	0.01067	0.01536
2-6	0.00468	0.04680	0.00404
3-6	0.01002	0.03122	0.03204
3-6	0.01002	0.03122	0.03204
4-8	0.00524	0.01184	0.01756
5-6	0.00711	0.02331	0.02732
6-7	0.04032	0.12785	0.15858
7-8	0.01724	0.04153	0.06014

B.6 Operating point #1:

	G_1	G_2	G_3	G_4	G_5
$P, p.u.$	5.1076	8.5835	0.8055	8.5670	0.8501
$Q, p.u.$	6.8019	4.3836	0.4353	4.6686	0.2264
$V, p.u.$	1.0750	1.0500	1.0250	1.0750	1.0250
$\delta, rad.$	0.0000	0.3167	0.2975	0.1174	0.3051

Loads in admittances in *p.u.*:

$$L_1 = 7.5 \text{ -} j5.0 \ , \ L_2 = 8.5 - j5.0 \ , \ L_3 = 7.0 - j4.5$$

B.7 Operating point #2:

	G_1	G_2	G_3	G_4	G_5
$P, p.u.$	3.1558	3.8835	0.4055	4.0670	0.4501
$Q, p.u.$	2.9260	1.4638	0.4331	2.1905	0.2574
$V, p.u.$	1.0550	1.0300	1.0250	1.0500	1.0250
$\delta, rad.$	0.0000	0.1051	0.0943	0.0361	0.0907

Loads in admittances in *p.u.*:

$$L_1 = 3.755 \text{ -} j2.5 \ , \ L_2 = 4.25 - j2.5 \ , \ L_3 = 3.5 - j2.25$$

Appendix C

Physical Model Power System

C.1 The parameters of the micro-alternator are:

$r_a = 0.026$	$r_{kd} = 0.0083$	$r_{kq} = 0.0083$
$r_f = 0.000747$	$x_f = 1.27$	$H = 4.75$
$x_d = 1.2$	$x_{kd} = 1.25$	$x_{md} = 1.129$
$x_q = 1.2$	$x_{kq} = 1.25$	$x_{mq} = 1.129$

C.2 Each transmission line consists of six 50 km equivalent π-section. For each π-section, the parameters are:

$$R = 0.036 \quad X = 0.0706 \quad B = 18.779$$

C.3 The parameters of the conventional PSS are:

$$K_s = -8.0, T_1 = 0.06, T_2 = 0.1, T_3 = 0.1, T_4 = 0.08$$

All resistances and reactances are in *p.u.* and time constants in seconds.