THE UNIVERSITY OF CALGARY

ADAPTIVE POWER SYSTEM STABILIZER BASED ON RECURRENT NEURAL NETWORK

by

JIAN HE

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
OCTOBER, 1998

© JIAN HE 1998

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-38471-3

Canada

ABSTRACT

An adaptive power system stabilizer based on recurrent neural networks is developeded in this dissertation. The Real-Time Recurrent Neural Networks with the Real Time Recurrent Learning (RTRL) algorithm are applied in the design of a recurrent neural network based power system stabilizer (RNN PSS). The structure and training procedure of the proposed RNN PSS are discussed.

The architecture of the proposed RNN PSS has two recurrent neural networks. The first one functions as an identifier to learn the dynamic characteristics of power plant, the second one functions as controller to damp the oscillations of power plant caused by different disturbances. The training of these two neural networks has two stages: off-line training and on-line update. There is no reference model needed for the proposed RNN PSS. It is trained directly based on the input and output of the plant.

Simulation studies and comparison between the proposed RNN PSS and the conventional PSS are conducted on both a single-machine infinite-bus power system model and a multi-machine power system model. The results demonstrate the effectiveness of the proposed RNN PSS in damping oscillations in the power system.

The designed RNN PSS is also implemented on a TMS320C30 Digital Signal Processing Board in a real time enviroment, and applied to a physical power system which consists of micro-alternator, DC motor, ABB PHSC2 PLC. The laboratory test results show that the prototype RNN PSS can provide a good response compared to the conventional PSS.

ACKNOWLEDGEMENTS

This dissertation would have been impossible without the advice, guidance and encouragement of my supervisor Dr.O.P.Malik. I am fortunate to have such an inspiring supervisor during my program. From him, I have learned so much not only in the area of my specialty but also in the style of teaching and research.

I am very grateful to Dr. T.Chen for his valuable discussions and insightful suggestions. I also wish to thank Mr. G.Hancock for his valuable help during my lab implement procedure. I also wish to thank all professors and support staffs in the Department of Electrical and Computer Engineering, The University of Calgary, for their help during this project.

Special appreciation also goes to many people starting with: Dr.G.P.Chen, Dr.Ali Hariri, Mr. Mariano Calvo, Mr.Antol Soos and all other people around me, whose names are impossible to be list here, for their valuable help from time to time.

My Special thanks also go to my parents for their help and encouragement to me. Last, but by no means least, I am truly indebted to my wife . Her continued patience and good humor have been of immense help.

To my parents and my wife Li Xiao.

CONTENTS

CHAPTERS

APPENDIX

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND NOMENCLATURE

ACE	Adaptive Critic Element
A/D	Analog to Digital converter
ANN	Artificial Neural Network
APSS	Adaptive Power System Stabilizer
ASE	Associative Search Element
AVR	automatic voltage regulator
BP	Back Propagation algorithm
CPSS	Conventional Power System Stabilizer
COFF	Common Object File Format
CT	Current Transformer
DAS	Data Acquisition System
D/A	Digital to Analog converter
DARAM	Dual Access RAM
DSP	Digital Signal Processor
E(y)	energy function
E_p	total squared error in ANN training
FACTS	Flexible AC Transmission System
FLC	Fuzzy Logic Control
GMV	Generalized Minimum Variance
H	generator inertia constant
HVDC	High Voltage Direct Current
$J_c(k)$	neural controller performance index at time step k
$J_i(k)$	neural identifier performance index at time step k
K_s	conventional power system stabilizer gain
K_A, K_C, K_F	AVR gains

K_{LR}, I_{LR}	AVR gains
K_d	generator damping ratio coefficient
MV	Minimum Variance
P	generator active power output
PA	Pole Assignment
P_a	acceleration power
PF	Power Factor
PHSC	Programmable High Speed Controller
PLC	Programmable Logic Controller
PS	Pole Shifting
PT	Potential Transformer
PSS	Power System Stabilizer
Q	generator reactive power output
R_C, X_C	voltage transducer compensation constants
RLS	Recursive Least Squares
RNN PSS	Recurrent Neural Network based Power System Stabilizer
RTRNN	Real-Time Recurrent Neural Network
SISO	Single-Input Single-Output
SVC	Static Var Compensator
$T_1, \ldots, T_6$	conventional power system stabilizer time constants
T_A, T_R, T_F	AVR time constants
T_B, T_{B1}	AVR time constants
T_C, T_{C1}	AVR time constants
T'_{d0}	generator d-axis transient time constant
T''_{d0}	generator d-axis sub-transient time constant
T_g	governor time constant
T''_{q0}	generator q-axis sub-transient time constant

TCR	Time Constant Regulator
TCSC	Thyristor Controlled Series Capacitor
T_e	generator electrical power output
T_m	generator mechanical power input
$U_{annc}(k)$	neural controller output at time step k
V_{pss}	PSS control signal
V_{AMAX}	AVR command signal upper limit
V_{AMIN}	AVR command signal lower limit
V_{IMAX}	AVR input upper limit
V_{IMIN}	AVR input lower limit
V_{OEL}	AVR over-excitation limit
V_{RMAX}	voltage regulator upper limit
V_{RMIN}	voltage regulator lower limit
V_{STMAX}	PSS output upper limit
V_{STMIN}	PSS output lower limit
V_{UEL}	AVR under-excitation limit
V_{ref}	AVR reference voltage
a,b	governor gain constants
d_{pj}	jth desired output of pattern p
e_d	generator d-axis voltage
e''_d	generator d-axis sub-transient voltage
e_f	generator field voltage
Δe_{fd}	generator field excitation voltage deviation
$\Delta e'_q$	generator q-axis transient voltage deviation
e'_q	generator q-axis transient voltage
e''_q	generator q-axis sub-transient voltage
e_q	generator q-axis voltage

g	governor output
i_d	generator d-axis current
i_f	generator field current
i_{kd}	generator d-axis damper winding current
i_{kq}	generator q-axis damper winding current
i_q	generator q-axis current
λ_d	generator d-axis flux linkage
λ_f	generator field flux linkage
λ_{kd}	generator d-axis damper winding flux linkage
λ_{kq}	generator q-axis damper winding flux linkage
λ_q	generator q-axis flux linkage
p.u.	per unit
r_a	generator armature resistance
r_f	generator field resistance
r_{kd}	generator d-axis damper winding resistance
r_{kq}	generator q-axis damper winding resistance
r_e	transmission line resistance
u	vector of ANN external inputs
u_k	PSS output control signal to generator unit at k *th* sample
w_{ij}	weight from neuron j to neuron i
$W_i(k)$	weight matrix of the neural identifier at time step k
$W_c(k)$	weight matrix of the neural controller at time step k
x'_d	generator d-axis transient reactance
x''_d	generator d-axis sub-transient reactance
x_d	generator d-axis reactance
x_f	generator field reactance
x_{kd}	generator d-axis damper winding reactance

x_{kq}	generator q-axis damper winding reactance
x_{md}	generator d-axis mutual reactance
x_{mq}	generator q-axis mutual reactance
x_q''	generator q-axis sub-transient reactance
x_q	generator q-axis reactance
x_e	transmission line reactance
$\underset{\sim}{y}$	vector of neuron outputs
y_j	output of neuron j
y_p	output signal of the controlled plant
y_{pd}	desired response of the controlled plant
$\Delta\omega$	speed deviation of generator
$\Delta\omega(k)$	speed deviation of generator at time step k
$\Delta\hat{\omega}(k)$	predicted speed deviation of generator at time step k
$\Delta\omega_d$	desired speed deviation of generator
ΔP_e	power deviation
$\Delta P_e(k)$	power deviation of generator at time step k
$\Delta\hat{\omega}(k)$	predicted power deviation of generator at time step k
$\Delta\omega_d$	desired power deviation of generator
δ	generator power angle
$\Delta\delta$	generator power angle deviation
$\Delta\delta_{kj}$	Kronecker delta
α	momentum factor in ANN learning algorithm
η	learning rate in ANN learning algorithm
η_c	learning rate of neural controller
η_i	learning rate of neural identifier
θ	vector of neuron biases
θ_i	bias of neuron i

CHAPTER 1

INTRODUCTION

1.1 Power System and Stability

Today, the electrical power systems are no longer operated as isolated systems, but as interconnected systems which may include thousands of electric elements and be spread over vast geographical areas. The advantages of interconnected power systems are that they[1][2]:

- provide large blocks of power and increase reliability of the system.
- reduce the number of machines which are required both for operation at peak load, and required as spinning reserve to take care of a sudden change of load.
- provide economical sources of power to consumers.

On the other hand, interconnection of systems also brings about new problems. The interconnecting ties between neighboring power systems are relatively weak when compared to the connections within the system. It easily leads to low frequency interarea oscillations. Many of the early instances of oscillatory instability occured at low frequencies when interconnections were made.

The study of power system stability is an interesting topic in electrical engineering research. Power system stability may be defined as[1]:

> The property of the system that enables synchronous machines of a system to respond to a disturbance from a normal operating condition so as to return to a condition where their operation is again normal.

Depending on the nature and order of magnitude of the disturbance, stability studies are usually classified into three types, namely[1][2][3]:

Steady State Stability - It refers to the behaviour of a power system around a fixed operating point; the system is subjected to small and gradual changes in the operating conditions.

Dynamic Stability - It refers to the long time response of a power system to relatively small disturbances. It differs from the steady state stability because it assumes that the system is steady state stable, and the system is subjected to small disturbances.

Transient Stability - It is aimed at determining if a system will remain in synchronism following major disturbances such as transmission system faults, sudden load changes, loss of generating units, or line switching. Transient stability problems can be subdivided into first-swing problems where the time period under study is the first second following the disturbance , and multiswing stability problems where the period under study may be extended to over 10 seconds.

For the steady state and dynamic stability problems, a power system can be modeled by linear differential equations. For the transient stability problem, a power system must be represented by nonlinear differential equations. In all stability studies, the objective is to determine whether or not the rotors of the machines being perturbed return to constant speed operation.

1.2 Power System Damping Control Strategies

A common problem in small and large power systems is the inherent nature of oscillatory instability[4]. During the past decades, many methods have been investigated to improve the stability of power systems. Generally speaking, the damping control strategies can be divided into two broad groups:

- add damping control in the transmission path.
- add damping control at generator location.

1.2.1 Damping control in the transmission path

High-voltage direct current (HVDC) transmission plays an important role in improving system stability[5][6][7][8]. This is because there is no requirement to maintain synchronism for HVDC. On the other hand, HVDC provides positive oscillation damping because an HVDC link can change its power flow in accordance with AC system needs much faster than any power plant.

The principal disadvantage of HVDC is the cost and complexity of the rectifier-inversion equipment. Another disadvantage is that HVDC will generate harmonics in the AC system.

Flexible ac transmission system (FACTS) owes its tighter transmission control to its ability to manage the interrelated parameters that constrain power system such as series impedance, shunt impedance, phase angle etc[9][10] [11][12]. Some FACTS controllers are listed below[13]:

- Static Var Compensator (SVC) uses thyristor valves to rapidly add or remove shunt-connected reactors and/or capacitors.
- Thyristor controlled series capacitor (TCSC) can vary the impedance continuously to levels below and up to the transmission line's natural impedence.
- Static condenser (Statcon) generates reactive power and its polarity can be controlled by controlling the voltage.

- Phase angle regulator shifts voltage phase by adding or subtracting a variable voltage component that is perpendicular to the phase voltage of the line.
- Unified power controller obtains a net phase and amplitude voltage change that confers control of both active and reactive power.

Like HVDC, the main drawback of FACTS controllers is the cost.

1.2.2 Damping control at generator location

Excitation control is one type of damping control at generator location. It is preferred for the following reasons:

- the electrical system has much smaller time constants than the mechnical system,
- an electrical control system is more economical and easy to implement than a mechanical control system,
- because of small loop time constant, an electrical control system is effectively a continuously acting system; therefore, it can give smooth system response.

Generator supplementary excitation control, commonly refered to as Power System Stabilizer (PSS), is the first choice for enhancing the damping ability of an excitation system [14][15]. The basic function of a PSS is to extend stability limits by modulating generator excitation to provide damping to oscillations of synchronous machine rotors relative to one another. These oscillations of concern typically occur in the frequency range of approximately 0.2 to 2.5 Hz, and insufficient damping of these oscillations may limit the ability to transmit power. To provide damping, the stabilizer must produce a component of electrical torque on the rotor which is in phase with speed variation. The PSS input signals are usually one of the following or a combination of them:

- speed deviation
- acceleration
- bus frequency deviation
- electrical power deviation

1.3 Types of Power System Stabilizers

Power System Stabilizers(PSS) have been studied extensively in the recent decades. Various techniques have been applied for PSS design.

1.3.1 Conventional Power System Stabilizer

The earliest PSS developed is called conventional PSS (CPSS) [16] [17] [18] [19] [20] [21] [22] [23][24] . It is based on the use of a transfer function designed using the classical control theory. It uses a lead-lag compensation network to compensate for the phase shift caused by the low frequency oscillation of the system during perturbations. By appropriately tuning the parameters of a lead-lag compensation network, it is possible to make a system have desired damping ability.

However, power systems are highly non-linear systems. The linearized system models used to design conventional power system stabilizers are valid only at the operating point that is used to linearize the system. As a fixed parameter controller, CPSS can not provide optimal performance under very wide operating conditions which power systems usually have. Therefore, the following problems are presented in the design of CPSS:

- How to choose a proper transfer function for a PSS that will give satisfactory supplementary control signal, covering all frequency ranges of interest.
- How to effectively tune the PSS parameters,

- How to automatically track the variation of the system operation condition,

- How to consider the interaction between the various machines.

Lots of research has been carried out to solve these problems. One of the solutions is an adaptive controller that has the ability to automatically tune itself to a new environment.

1.3.2 Adaptive Power System Stabilizer

A large amount of work has been done on adaptive PSS (APSS) design[25][26] [27][28][29]. Most of the adaptive PSSs use self-tuning adaptive control scheme, as it is one of the most effective adaptive control schemes. The structure of a self-tuning adaptive PSS has two parts: an on-line parameter identification and a control strategy.

At each sampling period, a mathematical model is obtained by on-line identification method to track the dynamic behavior of the plant. Different identification techniques have been proposed. In self-tuning adaptive PSS design, the recursive least squares (RLS) method is widely used for its simplicity, very good numerical stability and fast convergence property.

The control strategy calculates the control signal based on on-line identified paramters. There are several control strategies that can be used in a self-tuning adaptive control such as: minimum variance (MV) control strategy, generalized minimum variance (GMV) technique, pole assignment (PA) control strategy and pole shifting (PS) control strategy.

Studies have shown that an adaptive PSS can adjust its parameters on-line according to the change in the environment, and maintain desired control ability over a wide operating range of a power system. The main limitation of adaptive control is that it takes a large amount of computing time for on-line parameter identification.

1.3.3 PSS Based on Fuzzy Logic Theory

Fuzzy logic control (FLC) is one of the new techniques that has gained increasing attention in power system control[30][31][32] [33][34][35] . It models a complex system by using a higher level of abstraction that is originated from accumulated knowledge, and expresses the knowledge with subjective concepts that are mapped to numeric ranges. The relative simplicity of FLC decreases the developement time and cost.

The basic fuzzy logic controller includes four components:

- fuzzification: transfer the crisp input variables to corresponding fuzzy variables.
- rule definition (knowledge base): contain the meaning of the linguistic values of the process state and control output variable.
- inference: obtain the firing strength of each rule according to membership values.
- defuzzification: convert the set of modified control output values into nonfuzzy control values.

The limitations of FLC are: first, sometimes it is not easy to establish a knowledge base for FLC. Second, for most FLCs, their membership functions are decided off-line, and kept unchanged during the on-line operation. From this view-point, FLC can be considered as a fixed parameter controller.

1.4 PSS Based on Artificial Neural Networks

Artificial neural networks (ANN) have become a useful tool in many engineering disciplines since 1980s, because they have the potential to treat many problems that cannot be handled by traditional analytic approaches. A neural network is a system with inputs and outputs and is composed of many simple and similar processing units called neurons. Each neuron has a number of internal parameters called weights. Changing the weights of neurons alters the behavior of these neurons and also changes the behavior of the whole system.

1.4.1 Advantages of Neural Network

The use of neural networks provides the following useful properties and capabilities:

- Nonlinearity

  A neuron is a nonlinear unit. A neural network, made up of an interconnection of neurons, is itself nonlinear. Theoretically, a neural network has the ability to approximate abitrary nonlinear mappings.

- Learning and adptation

  Neural networks have built-in ability to adapt their weights to changes in the surrounding environment. A neural network trained to operate in a specific environment can be retrained to deal with a new environment. A properly trained network also has the ability to generalize when presented with the inputs not encountered during learning procedure.

- Parallel distributed processing

  Neural networks have a highly parallel structure and the basic processing element in a neural network has a very simply structure. This makes neural network potentially fast for the computation of certain tasks and ideally suited for parallel implementation using Very-Large-Scale-Integrated technology.

- Uniformity of Analysis and Design

Basically, neural networks enjoy *universality* as information processors since the same notation is used in all the domains involving the application of neural networks.

- Multivariable systems

  Neural networks naturally process many inputs and have many outputs. This is very attractive in control area application since the neural networks are readily applicable to multi-input multi-output system.

1.4.2 Limitations of Neural Network

However, a neural network technique is not a perfect one. It also has some drawbacks:

- There is no systematic method to decide the neural network structure for different application, i.e., how many layers should be used in the network, how many neurons should be employed in the layers. It is usually decided by a trial and error method.

- There is a long training time for the neural network to be trained properly. The more complex the system, the longer the time.

- The neural network has a *black box* characteristic. It is difficult to understand the information stored in the network.

1.4.3 Neural Network Application in Power System

A power system is a complex system. It can be characterized as non-linear, large-scale, dynamic, stochastic, time-variant parameter system, etc. With the advantages of neural networks mentioned above, neural networks provide exciting alternatives for power system problem solutions. The application of neural networks in power systems is summarized as follows: [36]

(1) System Planning

- Load Forecasting
- Economic Load Dispatching
- Unit Commitment
- Forecasting of Harmonics
- Network Reconfiguration
- Generation Expansion and Maintenance Scheduling

(2) Security Assessment

- Dynamic Security Assessment
- Static Security Assessment
- Voltage Security Assessment

(3) Fault Detection and Diagnosis

- Alarm Processing
- Component Fault Detection
- Power System Fault Detection

(4) Control

- Machine and Plant Control
- Optimum Power Flow
- Voltage Reactive Power Control

(5) Analysis

- Parameter and State Estimation

- Machine, Plant and System Modeling
- Load Modeling
- Power Flow Calculation
- Harmonics Evaluation

(6) Protection

1.4.4 Neural Network in PSS design

Applications of artificial neural networks in power system stabilizer design have been reported since the early nineties.

In [37], an artificial neural network is investigated to tune the parameters of a conventional PSS based on generator loading conditions. A proportional- integral (PI) PSS is also employed in the research. A pair of on-line measurements, generator real power output (P) and power factor (PF), are chosen as the input signals. The outputs of the neural network are the desired PSS parameters: the gain setting K_p and K_I. The type of neural network in this paper is multilayer feedforward network. The neural network is trained off-line.

In [38], a neural network based regulator is proposed for non-linear, multivariable turbogenerator. The NN regulator consists of two neural networks which are used for input-output mapping and control, respectively. The multilayer feedforward network with the back-propagation training algorithm is employed. The neural network provides control signals for both excitation system and governor.

An artificial neural network based adaptive PSS is proposed in [39]. This PSS combines the advantages of self-optimizing pole shifting adaptive control strategy and the quick response of ANN. Again, the type of ANN in the design is the multilayer feedforward network trained with back-propagation algorithm. After the training is finished, the weights of the neural network are kept constant for on-line application.

Later in [40], a neural network based PSS is trained to learn the reverse input/output mapping of the generating unit and work as the inverse plant. The shaft speed deviation and past control signals are inputs to the neural network. Further, in [41], a multi-input ANN PSS is developed to function as the inverse plant. The inputs of the neural network are the generator speed deviation, electrical power deviation and previous control signal. In all three ANN PSSs, only one neural network is employed, and the output of the neural network is the supplementary control signal. A similar ANN PSS structure is also introduced in [42]; however, the ANN PSS is trained by samples obtained from power system controlled by nonlinear PSS. The back-propagation algorithm is modified with variable learning rate and momentum factor.

In [43], an adaptive neuro-control system is studied for power system stabilization. The proposed control system consists of two multi-layered feedforward neural networks that function as an identifier and a controller, respectively. The inputs for the neural network are $\Delta\delta$, $\Delta\omega$, $\Delta e'_q$, Δe_{fd} and ΔP_m. The output is also the supplementary control signal. An ANN PSS with neuro-identifier and neuro-controller is also presented in [44]. The ANN PSS is tuned by on-line training algorithm, and the inputs for ANN PSS are measurable variables from generator units.

In [45], a PSS combining fuzzy logic control and neural network is reported. A five-layer feedforward network is employed to represent the fuzzy logic system. The parameters in membership functions and inference rules are adjusted by the back propagation algorithm. To train the new controller, a self-optimizing pole-shifting APSS has been used for providing the training data. The input of network is generator speed deviation $\Delta\omega$ and its derivative $\Delta\dot{\omega}$. A similar neuro-fuzzy controller architecture is also reported in [46], where an adaptive neural network based fuzzy inference system is used to control low-head hydro power plant. The proposed controller provides control signals to exciter system and governor. In [47], a fuzzy logic

PSS based on neural network with self-learning ability is reported. The new PSS is trained directly from the input and output data of the generating unit. It does not need a desired controller to provide training data.

The successful digital simulation results in those papers show that neural networks have a considerable potential in power system stability study. By far the multi-layer feedforward network is the most common neural network employed in current research.

1.5 Dissertation Objectives

The design of an adaptive power system stabilizer based on recurrent neural network is studied in this dissertation. It is hoped that the research work will make a contribution to the development of a power system stabilizer based on artifical neural networks. The objectives of this dissertation are listed in the following aspects:

- Investigate the theory of ANN, architectures of the different ANN types, and learning algorithms for ANN. Discuss the feasibility of different ANNs for power system control. Select a specific type of ANN suitable for PSS application.

- Design a framework of new power system stabilizer based on recurrent neural networks. To obtain the proper information about the plant, besides a neural network controller, a neural network identifier is introduced in the control system.

- Develop a control algorithm for the proposed PSS. The neural network PSS has self-learning ability. It does not need another PSS to provide training data as it is trained directly from the input and output data of the controlled generator. On the other hand, power systems are not fixed. The operating conditions or the structure of the power system network change from time to time. Therefore, it is desired that the parameters of the neural network PSS can be tuned adaptively.

- Behaviour of the neural network PSS is investigated both in the single machine infinite bus environment and the multi-machine power system environment through computer simulation. The results are also compared with the results of the conventional PSS.

- In addition to the computer simulation studies, implement the proposed PSS in a physical laboratory environment, and test the performance of the RNN PSS on-line.

1.6 Contributions of Dessertation

The main contribution can be summarized as follows:

- The recurrent neural networks are employed in the design of PSS. For the recurrent neural network, its present outputs not only depend on present inputs but also depend on past inputs and outputs. This kind of input/output relationship is also true for the design object – generating unit.

- An adaptive control algorithm is developed. Therefore, the neural network PSS has self-learning ability. In the design procedure, the mathematical model of the generating unit is not linearized at any special generator operation points. Also, there is no reference model needed for the RNN PSS.

- The behavior of the neural network PSS is simulated off-line on a SUN workstation for a single machine infinite bus system and a multi-machine power system. The theoretical simulation studies verified the effectiveness of the neural network PSS.

- The laboratory implementation of the RNN PSS is tested in a real-time environment. The results illustrate that the RNN PSS can provide better damping effect than CPSS. It also shows that the proposed PSS is practical in the sense that it is feasible to install a RNN PSS by using commonly available computer hardware and obtain satisfactory performance.

1.7 Summary of Dissertation Chapters

This dissertation has seven chapters as outlined below:

- Chapter 2 gives an overview of the artifical neural network. Fundamentals of different artifical neural networks are presented, followed by a discussion on the training algorithm of different neural networks. Based on the comparison of the characters of the different types of ANNs and their suitability to power system control research, the recurrent neural network is selected for the new PSS design.

- Chapter 3 presents application of artificial neural networks in dynamic system identification and control. The structure of the adaptive PSS based on recurrent neural networks is proposed. The control algorithm of the new PSS is also discussed in this chapter.

- Computer simulation studies are given in Chapter 4. The performance of the proposed Recurrent Neural Network based PSS (RNN-PSS) is studied in a single-machine infinite bus power system with different types of disturbances under various operating conditions. The results are also compared to that of a CPSS.

- In Chapter 5 the behaviour of the proposed RNN-PSS is investigated in a multi-machine power system. The response of the RNN-PSS to different local and inter-area oscillation models is studied.

- Implementation and experimental studies of the proposed RNN PSS are given in Chapter 6. The real time test environment includes a physical model of the power system, an ABB PHSC2 PLC and a PC installed with TMS320C30 DSP board. For comparison, a CPSS is also implemented in the same environment. The results obtained from experimental tests are discussed.

- Finally in Chapter 7, conclusions and suggestions for further research in this area are presented.

CHAPTER 2

ARTIFICAL NEURAL NETWORKS

2.1 Introduction

Today, the term artificial neural network refers to any computing architecture that consists of parallel interconnections of simple neurons. The motivation is that scientists seek solutions to problems that are difficult with today's digital computing technology, problems that are easily solved by human brains.

ANN research has experienced three periods of extensive activity. In the 1940s, McCulloch and Pitts showed that the neuron can be modeled as a simple threshold device to perform logic function [48]. From their pioneering work arose the first peak of modern neural network study. The second occurred in early 1960's with Rosenblatt's Perceptron, Widrow and Hoff's Adaline, and Steinbuch's Learning Matrix. However, when Minsky and Papert pointed out serious limitations of the simple Perceptron in their book *Perceptrons*, showing that the Perceptron was only capable of performing very restricted tasks because of its linearity, the neural network research was lulled for nearly 20 years. In the early 1980s, neural networks again showed promise with the publication of Hopfield's approach of introducing non-linearity and recognizing the collective power of simple computational elements. The phase of neural network development after 1982 has led to a plethora of network paradigms and learning algorithms. However, by far the most popular network learning algorithm is Back-Propagation of Error(BP) popularized by Rumelhart et al[49]. Cutter Information Corporation once estimated that BP is used for 95% of neural network applications.

In this chapter, the neuron model and the classification of the ANNs are reviewed [50] [51] [52] [53] [54] [55] [56]. Several popular neural networks, learning algorithms and their potential applications in power system control are discussed.

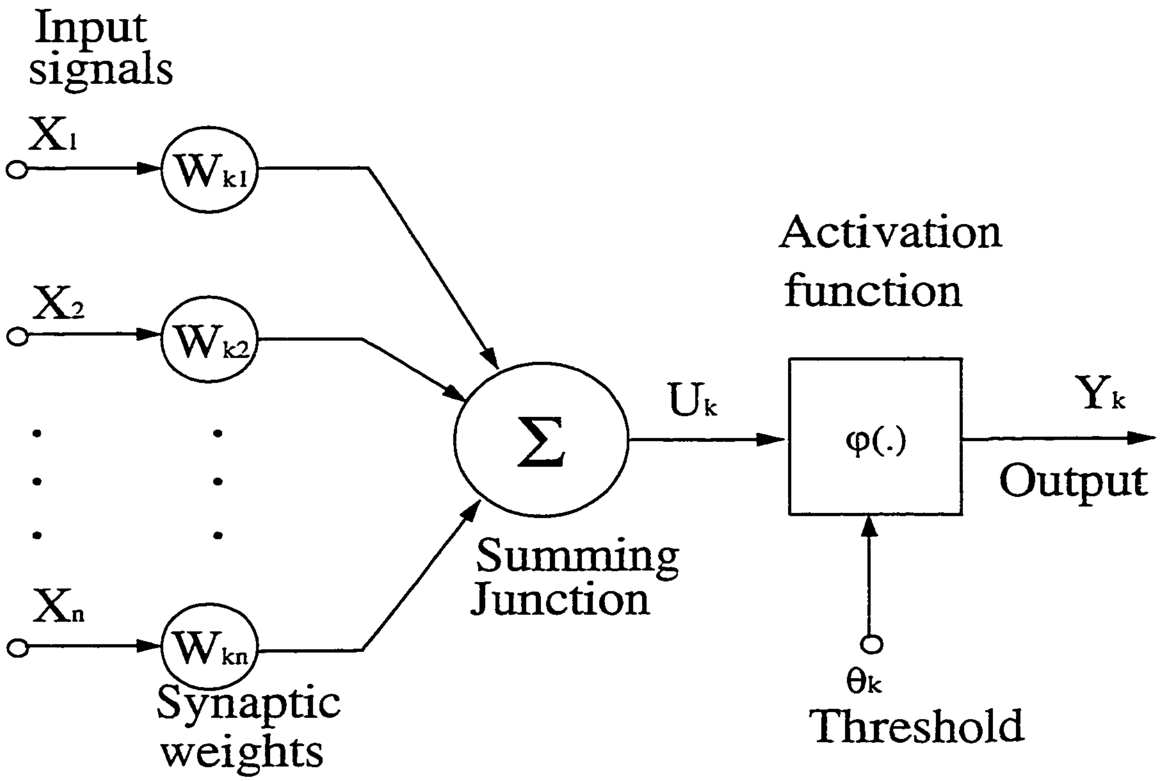


Figure 2.1. Schematic diagram of a neuron.

2.2 Neuron Model

Neuron is an information processing unit that is fundamental to the operation of a neural network. A standard model for a neuron is shown in Fig. 2.1. It has three basic elements:

- A set of synapses or connecting links.
- An adder for summing the input signals
- An activation function for limiting the amplitude of the output of a neuron.

Each of these basic elements are described as follows.

2.2.1 Synapses

The synapses or connecting links are characterized by *weights*. For a synaptic weight w_{kj}, the first subscript refers to the neuron in question and the second subscript refers to the input end of the synapse to which the weight refers. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}

2.2.2 Adder

An adder is used for summing the input signals, weighted by the respective synapses of the neuron.

In mathematical terms, the adder can be described by the following equation:

$$u_k = \sum_{j=1}^{p} w_{kj} x_j \tag{2.1}$$

where $x_1, x_2, \ldots, x_p$ are input signals; $w_{k1}, w_{k2}, \ldots, w_{kp}$ are the synaptic weights of neuron k; u_k is the *summing output.*

2.2.3 Activation Function

The activation function defines the output of a neuron in terms of the activity level of its input. Usually, the amplitude range of the output of a neuron is normalized as the closed unit interval [0, 1] or alternatively [-1, 1].

Six basic types of activation functions are shown in Table 2.1.

Among these activation functions, the sigmoid and hyperbolic tangent functions are popular. One of the advantages is that they are differentiable. This makes it possible to derive a gradient search learning algorithm for networks with multiple layers. Also, as these two functions are continuous, they are suitable for applications which require a continuous-valued output rather than the binary output.

Name	Formula	Characteristics
Threshold	+1, if x > 0 0, otherwise	Non-differentiable, Step-like, Positive
Threshold	+1, if x > 0 -1, otherwise	Non-differentiable, Step-like, Positive-Negative
Piecewise linear	+1, if $x \geq \frac{1}{2}$ x, if $-\frac{1}{2} < x < \frac{1}{2}$ 0, if $x \leq -\frac{1}{2}$	Differentiable, Positive
Sigmoid	$\frac{1}{1+e^{-ax}}$	Differentiable, Positive
Hyperbolic tangent	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	Differentiable Positive-Negative
Gausian	$e^{\frac{-x^2}{\sigma^2}}$	Differentiable Positive

Table 2.1. Types of Activation Function

2.3 Neural Network Architectures

When neurons are structured into a neural network, different connectivities yield different network behavior. Based on the architecture, neural networks can be grouped into two categories [54] [55] [56]:

feed-forward networks, in which graphs have no loops, they include:

- Single-layer perceptron;
- Multilayer perceptron;
- Radial Basis Function nets.

recurrent (or feedback) networks in which loops occur because of feedback connection. They include:

- Hopfield network;
- ART models;
- Discrete-Time Recurrent Neural Network;
- Elman Networks
- Kohonen's Self-Organized Map.

Feed-forward networks are static. They produce only one set of output values rather than a sequence of values from a given input. Feed-forward networks are memory-less in the sense that their response to an input is independent of the previous network state.

On the other hand, recurrent or feed-back networks are dynamic systems. Their outputs are a function not only of the current inputs, but also past inputs or outputs. Their node equations are typically described by differential or difference equations.

Recently, there have been a number of studies that investigate neural network architectures that are somewhat between recurrent networks and feedforward networks by incorporating local feedback in feedforward networks [57] [58] [59]. Local feedback can be introduced in one of three ways

(1) local activation feedback,

(2) local synapse feedback,

(3) local output feedback.

All those neural networks can be referred to as local recurrent globally feedforward neural networks.

2.4 Learning

Different network architectures require appropriate learning algorithms. A learning process for the neural network can be viewed as the problem of updating network architecture and connection weights so that a network can efficiently perform a specific task. Figure 2.2 shows The taxonomy of the learning process is shown in Figure 2.2.

2.4.1 Learning paradigms

A learning paradigm refers to a model of the environment in which the neural network operates. There are three broad paradigms of learning:

- Supervised learning;
- Unsupervised learning;
- Reinforcement learning.

Supervised learning implies that the system is directed by an external "teacher" to achieve the desired behaviour. A diagram of supervised learning system is shown

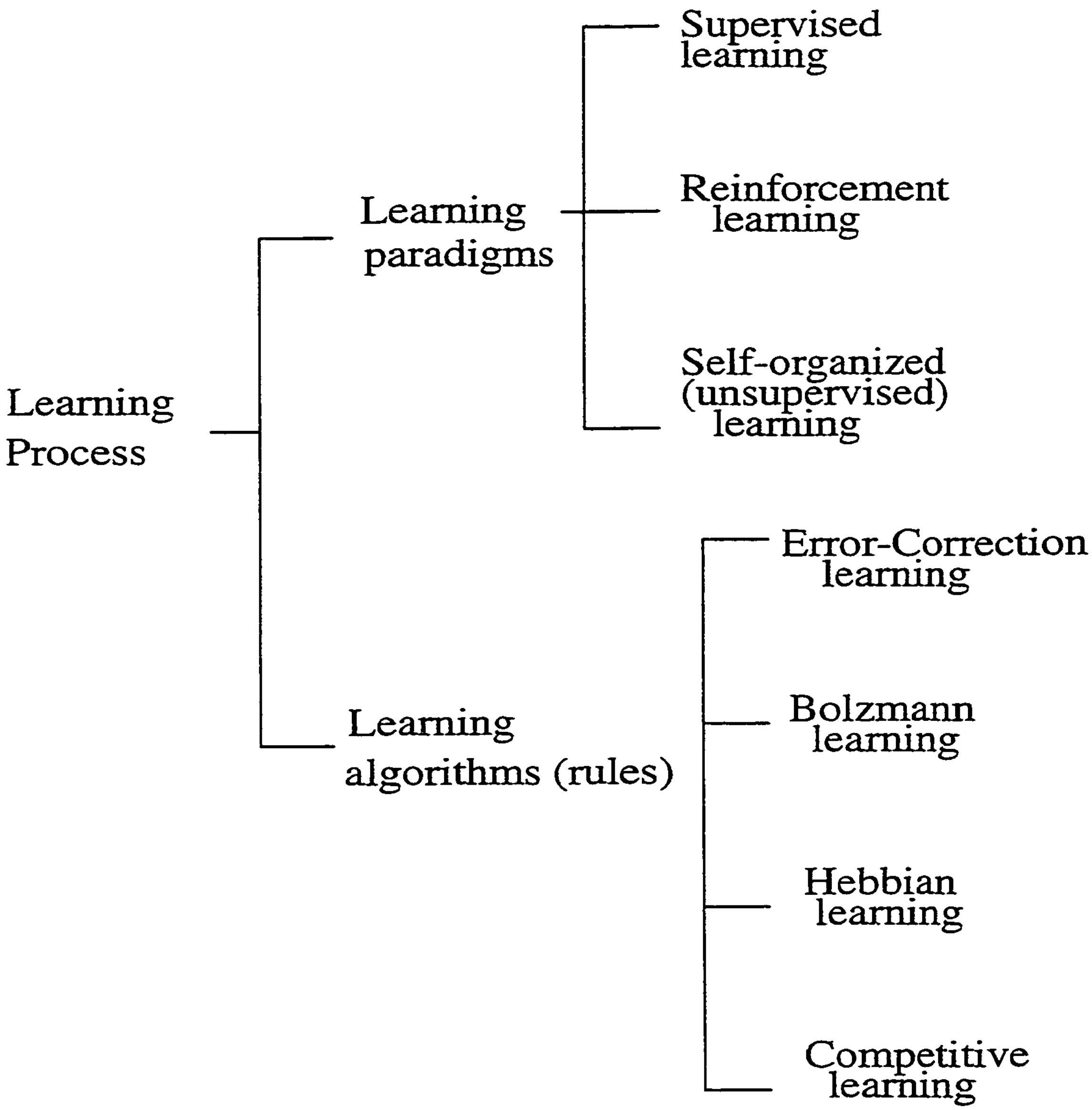


Figure 2.2. A taxonomy of the learning process

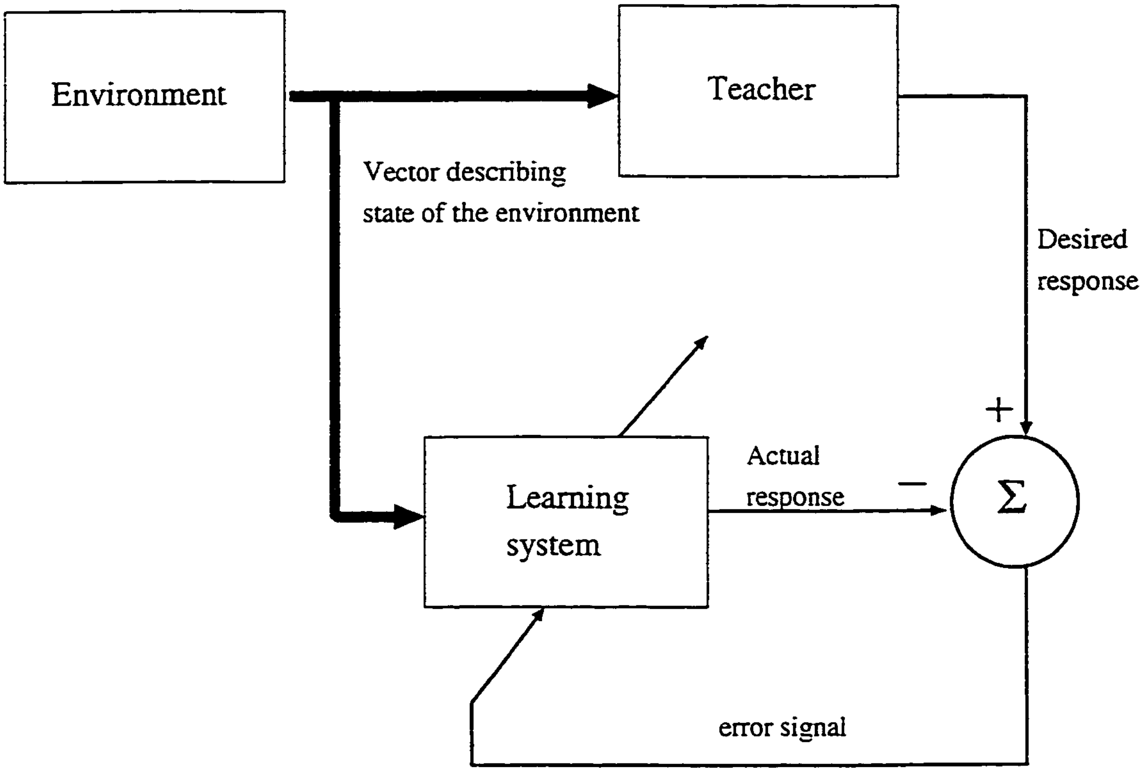


Figure 2.3. Schematic diagram of supervised learning

in Figure 2.3. In general, it implies that the system is supplied with examples of input-output samples. The input-output examples used during training are assumed to be mappings of a function that is to be learned by the neural network. It is important that the set of training examples is uniformly distributed over the input space and that it is presented to the learning network in a randomised manner. Firstly this is necessary to enable statistical independence of the data sets; and secondly, to introduce noise into the learning algorithm which aids good achievement of an optimal solution.

Unsupervised learning does not require a correct answer associated with each input pattern in the training data set, as shown in Fig. 2.4 . The basic idea is to have the processing units configurate their structure and parameters according to a probability distribution proportional to the distribution of the input vectors used to train the

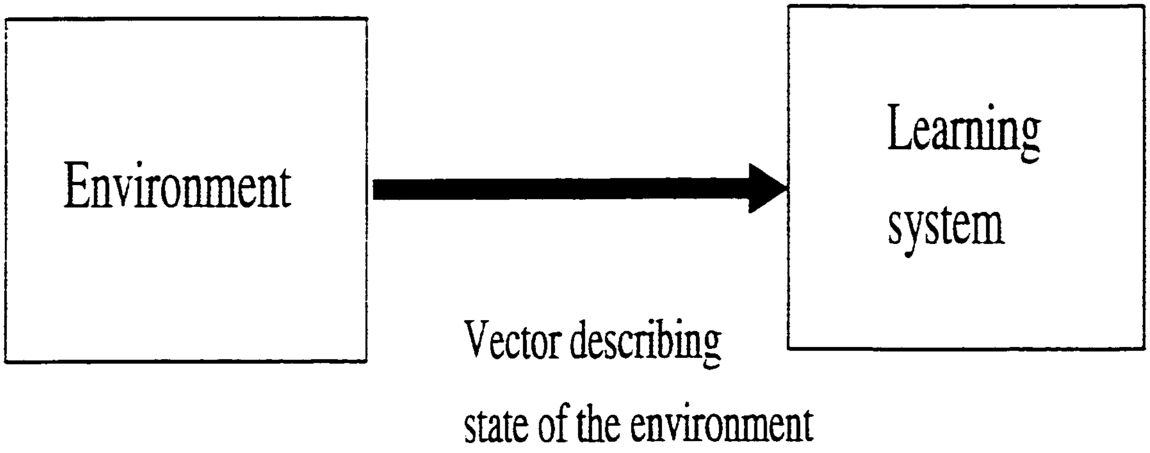


Figure 2.4. Schematic diagram of unsupervised learning

network. It explores the underlying structure in the data, or correlations between patterns in the data, and organizes patterns into categories from these correlations.

Reinforcement learning refers to the concept that if a particular action applied to a system is associated with a satisfactory response, then the trend to reproduce that action in a similar situation should be enhanced. Figure 2.5 shows the diagram of reinforcement learning. Reinforcement learning is in many ways similar to supervised learning because both require information from the interacting environment. However, Reinforcement learning only provides a scalar performance index, called the reinforcement signal, without indicating the direction in which the system could be improved. The network system just knows whether its output is correct or not. The advantage is that it is not necessary to provide the correct response to individual inputs to train the network. This is particularly useful for "on-line" application where it is usually difficult to know the desirable output resulting from a specified input.

2.4.2 Learning rules

There are four basic learning rules:

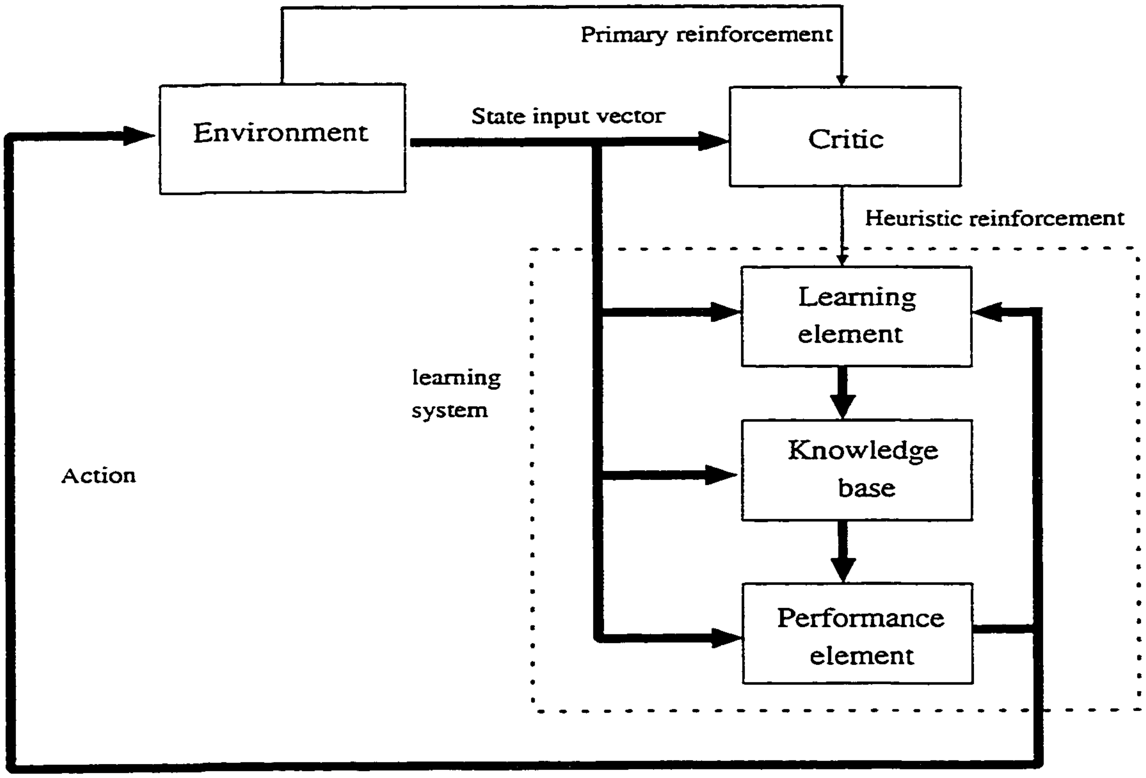


Figure 2.5. Schematic diagram of reinforcement learning

- Error-Correction learning;
- Boltzmann learning;
- Hebbian learning;
- Competitive learning.

2.4.3 Error-correction Learning

The basic principle of error-correction learning rules is to use a cost function based on the error signal to modify the connection weights to gradually reduce this error. The error signal can be defined as:

$$e_k(n) = d_k(n) - y_k(n) \tag{2.2}$$

where $d_k(n)$ is the desired response (or training output) for neuron k at time n and y_k(n) is the actual response of this neuron.

The cost function can be defined as :

$$E(n) = \frac{1}{2} \sum_k e_k^2(n) \tag{2.3}$$

The network is then optimized by minimizing E(n) with respect to the synaptic weights of the network. The adjustment made to the synaptic weight w_{kj} at time n is given by

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \tag{2.4}$$

where η is a positive constant that determines the rate of learning, x_j(n) is the neuron input. In other words, the adjustment made to a synaptic weight is proportational to the product of the error signal (measured with respect to some desired response at the output of that neuron) and the input signal of the synapse in question. The updated value of synaptic weight w_{kj}(n+1) can be writen as:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \tag{2.5}$$

The learning rate parameter η has a profound impact on the performance of the error-correction learning in that it affects not only the rate of convergence of learning but also the convergence itself. If η is small, the learning process proceeds smoothly, but it may take a long time for the system to converge to a stable solution. If, on the other hand, η is large, the rate of learning is accelerated, but now there is a danger that the learning process may diverge and the system therefore becomes unstable.

A plot of the cost function **J** versus the synaptic weights characterizing the neural network consists of a multidimensional surface refered to as an *error-performance surface*. There are two distinct suitations for the error-performance surface:

- the error-performance surface is bowl-shaped with a *unique* minimum point.

- the error-performance surface has a *global minimum* (perhaps multiple global minima) as well as *local minima.*

In both cases, the objective of the error-correction learning is to start from an arbitrary point on the error surface and then move toward a global minimum, in a step-by-step fashion. In the first case this objective is indeed attainable. In the second case, it is not always attainable, as it is possible for the algorithm to get trapped at a local minimum of the error surface and therefore never to be able to reach a global minimum.

2.4.4 Boltzmann Learning

Boltzmann learning is a stochastic learning rule derived from information-theoretic and thermodynamic principles. The Boltzmann machine is characterized by an *energy function E*, the value of which is determined by the particular states occupied by the individual neurons of the machine, as given by

$$E = -\frac{1}{2}\sum_i \sum_j w_{ji} s_j s_i; (i \neq j) \tag{2.6}$$

where s_i is the state of neuron i, and w_{ji} is the synaptic weight connecting neuron i to neuron j. The machine operates by choosing a neuron at random at some step of the learning process, and flipping the state of neuron j from s_j to -s_j at some pseudo-temperature T, with probability

$$W(s_j \rightarrow -s_i) = \frac{1}{1 + exp(-\Delta E_j/T)} \tag{2.7}$$

where ΔE_j is the *energy change* resulting from such a flip. If this rule is applied repeatedly, the machine will reach *thermal equilibrium.*

The neurons of a Boltzmann machine can be grouped into two categories: *visible* and *hidden*. The visible neurons provide an interface between the network and the environment in which it operates, whereas the hidden neurons always operate freely. There are two modes of operation to be considered:

- *Clamped condition*, in which the visible neurons are all clamped onto specific states determined by the environment.
- *Free-running condition*, in which all the neurons (visible and hidden) are allowed to operate freely.

The change in the connection weight w_{ij} is given by:

$$\Delta w_{ij} = \eta(\overline{\rho_{ij}} - \rho_{ij}), \tag{2.8}$$

where η is the learning rate, and $\overline{\rho_{ij}}$ and ρ_{ij} are the correlations between the states of units i and j when the network operates in the clamped mode and free-running mode, respectively. The values of $\overline{\rho_{ij}}$ and ρ_{ij} are usually estimated from Monte Carlo experiments, which are extremely slow.

Boltzmann learning can also be viewed as a special case of error-correction learning in which error is measured not as the direct difference between desired and actual outputs, but as the difference between the correlations among the outputs of two neurons under clamped and free-running operating conditions.

2.4.5 Hebbian Learning

Hebb's postulate of learning rule is based on the following observation from neurobiological experiments: If neurons on both sides of a synapse are activated synchronously and repeatedly, the synapse's strength is selectively increased.

Mathematically, the Hebbian rule can be described as :

$$\Delta w_{ij}(n) = \eta y_j(n) x_i(n) \tag{2.9}$$

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \tag{2.10}$$

where x_i and y_j are the output values of neurons i and j, respectively, which are connected by the synapse w_{ij}, and η is the learning rate.

However in equation. 2.9, if the input signal x_j is applied repeatedly, the synaptic weight w_{ij} will have an exponential growth and finally reach saturation. To avoid such a situation from arising, a limit needs to be imposed on the growth of synaptic weights. One method for doing this is to introduce a nonlinear forgetting factor into the formula for the synpatic adjustment $\Delta w_{ij}(n)$. It can be redefined as :

$$\Delta w_{ij}(n) = (\eta x_i(n) - \alpha w_{ij}(n))y_k(n) \tag{2.11}$$

where α is a new positive constant. If $\eta x_i(n) > \alpha w_{ij}(n)$, the synaptic weight $w_{ij}(n+1)$ will increase in proportion to y_j. When $\eta x_i(n) < \alpha w_{ij}(n)$, the the synaptic weight $w_{ij}(n+1)$ will decrease in proportion to y_j. The use of this method eliminates the problem of runaway synaptic weight instability.

An important property of this rule is that learning is done locally, that is, the change in synapse weight depends only on the activitives of the two neurons connected by it.

2.4.6 Competitive Learning

In competitive learning, the output neurons of a network compete among themselves for being the one to be active. Only a single output neuron is active at any one time, whereas in a neural network based on Hebbian learning multiple output neurons may be active simultaneously.

There are three basic elements to a competitive learning rule:

- A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.

- A limit imposed on the "strength" of each neuron.
- A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron, or only one neuron per group, is active.

For neuron j, to be the winning neuron, its net internal activity level v_j for a specified input pattern $\mathbf{x}$ must be the largest among all the neurons in the network. The output signal y_j of winning neuron j is set to one; the output signals of all the neurons that lose the competition are set equal to zero.

A neuron learns by shifting synaptic weights from its inactive to active input nodes. If a neuron does not respond to a particular input pattern, no learning takes place in that neuron. If a neuron wins the competion, its synaptic weights get updated. Let w_{ij} denote the synaptic weight connecting input node i to neuron j. The change Δw_{ij} applied to synaptic weight w_{ij} is defined by:

$$\Delta w_{ij} = \begin{cases} \eta(x_i - w_{ji}) & \text{if neuron j wins the competition} \\ 0 & \text{if neuron j loses the competition} \end{cases}$$

where η is the learning-rate parameter. The effect of this learning rule is to move the stored pattern in the winner weights a little bit closer to the input pattern. Competitive learning often clusters or categorizes the input data.

2.5 Multi-Layer Perceptron

2.5.1 Configuration

The Multi-Layer Perceptron is formed by cascading neurons in layers. A four-layer network (one input layer, two hidden layers and one output layer) is shown in Fig. 2.6. The input vector feeds into each of the first layer nodes and the outputs of this layer feed into each of the second layer neurons, and so on. The output layer of the

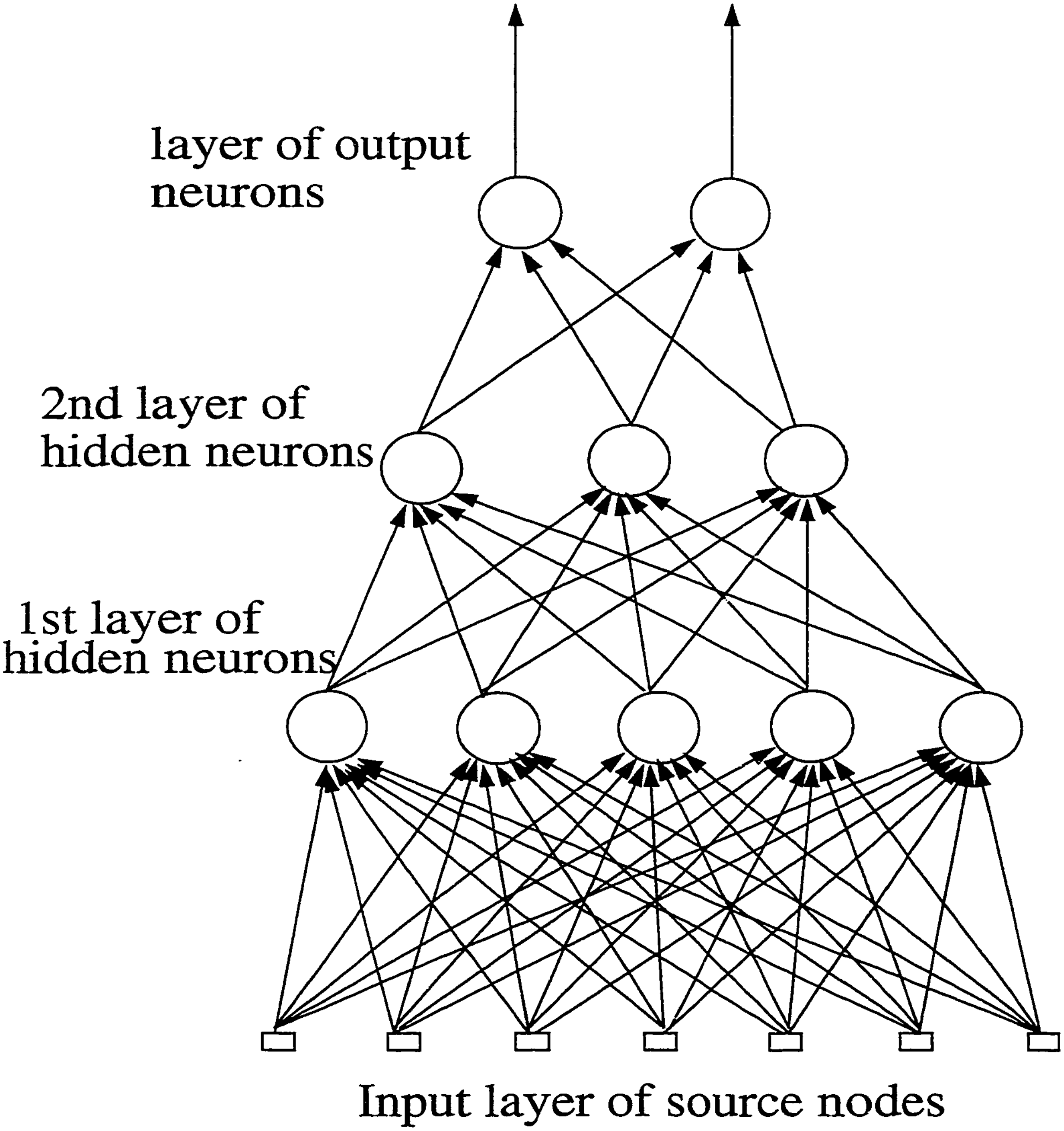

layer of output neurons
2nd layer of hidden neurons
1st layer of hidden neurons
Input layer of source nodes

Figure 2.6. Schematic diagram of a multi-layer neural network.

network generates the mapping output. Usually, the neurons are fully connected between layers, i.e., every neuron in layer l is connected to every neuron in layer l+1. However, there is no feedback connection and self-loop connection in the network. The layers that are neither inputs nor outputs are called hidden layers.

A multi-layer perceptron has three distinctive characteristics:

- The model of each neuron in the network includes a *nonlinearity* at the output end. A commonly used form is *sigmoidal nonlinearity* which is differentiable everywhere. The presence of nonlinearities is important because, otherwise, the input-output relation of the network could be reduced to that of a single-layer perceptron.
- The network contains one or more hidden layers which enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns.
- The network exhibits a high degree of connectivity, determined by the synapses of the network.

The capabilities of the Multi-Layer Perceptron can be viewed from three different perspectives.

- to implement Boolean logic functions;
- to partition space for classification problems;
- to implement nonlinear transformations for functional approximation problems.

2.5.2 Back-propagation algorithm

One of the factors that make the Multilayer perceptron network so popular is the back-propagation learning algorithm [49] for determining weights in MLP. Generally, in back-propagation there are two passes through the different layers of the network: a forward pass and a backward pass.

In the forward pass, the weights remain unchanged throughout the network, and the output signals of the network are computed on a neuron-by-neuron basis. During the backward pass, the weights are all adjusted in accordance with the error-correction rule. First an error signal is produced by subtracting the actual response of the network from a desired response. It is then propagated backward through the network.

The back-propagation algorithm can be summaried as follows:

1. Initialize all the synaptic weights and threshold of network to small random values.

2. Propagate the signal forward through the network. Let a training example in the epoch be denoted by [x(n), d(n)]. The net internal activity level $v_j^{(l)}(n)$ for neuron j in layer l is:

$$v_j^{(l)}(n) = \sum_{i=0}^{p} w_{ji}^{(l)}(n) y_i^{(l-1)}(n) \tag{2.12}$$

where $w_{ji}^{(l)}(n)$ is the weight of neuron j in layer l that is fed from neuron i in layer l-1. $y_i^{(l-1)}(n)$ is the function signal of neuron i in the previous layer l-1 at iteration n. For i = 0, $y_0^{(l-1)}$(n) = -1 and $w_{j0}^{(l)}$(n) = $\theta_j^{(l)}$(n), where $\theta_j^{(l)}$(n) is the threshold applied to neuron j in layer l. The output signal of neuron j in layer l is

$$y_j^{(l)} = \varphi(v_j(n)) \tag{2.13}$$

where $\varphi(.)$ is activation function of the neuron j in layer l. If neuron j is in the first hidden layer, set

$$y_j^{(0)} = x_j(n) \tag{2.14}$$

where x_j(n) is the jth element of the input vector X(n). If neuron j is in the output layer, set

$$y_j^{(L)} = o_j(n) \tag{2.15}$$

Hence, compute the error signal

$$e_j(n) = d_j(n) - o_j(n) \tag{2.16}$$

where d_j(n) is the jth element of the desired response vector **d**(n)

3. Compute the δ's (i.e., the local gradients) for the preceding layers by propagating the errors backward, layer by layer L. For neuron j in the output,

$$\delta_j^{(L)}(n) = e_j^{(L)}(n)\varphi_j'(v_j(n)) \tag{2.17}$$

For neuron j in hidden layer l,

$$\delta_j^{(l)}(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \tag{2.18}$$

4. Hence, adjust the weights of the neural network in layer l according to the generalized delta rule:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[w_{ji}^{(l)}(n) - w_{ji}^{(l)}(n-1)] + \eta\delta_j^{(l)}(n)y_i^{(l-1)}(n) \tag{2.19}$$

where η is the learning rate parameter and α is the momentum constant.

5. Go to step 2 and repeat for the next pattern until the error in the output layer is below a prespecified threshold or a maximum number of iterations is reached.

2.6 The Hopfield Network

The Hopfield network is one of the best known dynamic networks [60] [61]. The structure of a Hopfield network with 4 neurons is shown in Fig. 2.7. It is a single-layer network with the following characteristics:

- The output of each neuron in the network is fed back to all other neurons.

- There is no self-feedback in the network.

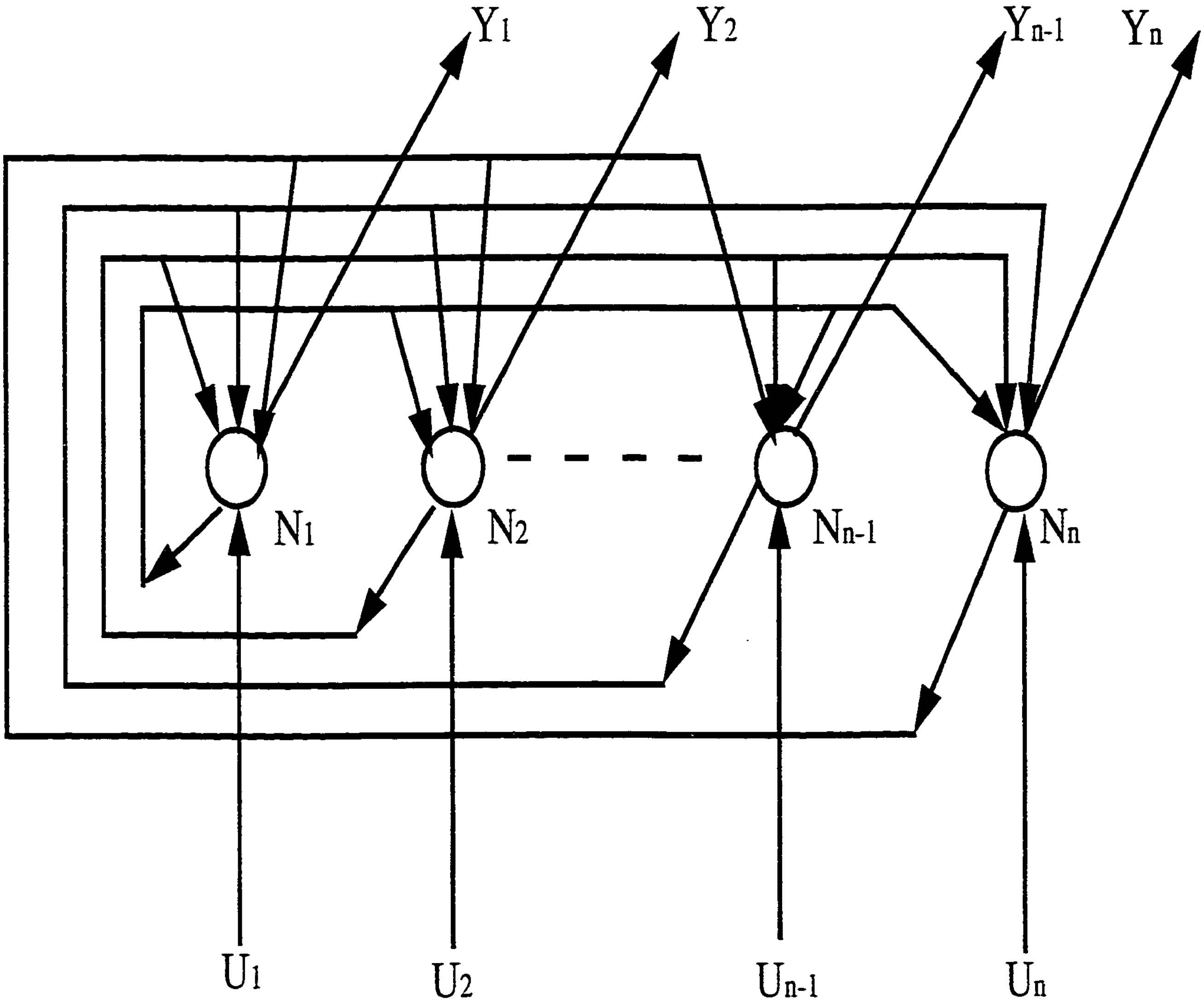


Figure 2.7. Schematic diagram of a Hopfield network.

- The weight matrix of the network is symmetric, i.e. the influence of neuron i on neuron j is equal to the influence of neuron j on neuron i.

The Hopfield network is a nonlinear dynamical system which is capable of exhibiting a wide range of complex behaviour. Depending on how the network parameters are chosen, it may function as a stable system, an oscillator, or even a chaotic system. There are two versions of the Hopfield Network: the continuous-time and discrete-time. The continous-time version will be discussed here.

The behaviour of the network can be described by the differential equation:

$$T_i\dot{x}_i = -x_i + \sum_{j=1}^{N} w_{ij}y_j + u_i \tag{2.20}$$

$$y_i = f_i(x_i), i = 1, ..., N \tag{2.21}$$

where x_i is the *internal state* of the ith neuron; y_i is the *output state* of the ith neuron; w_{ij} is the weight connecting the jth neuron to the ith neuron. Additionally, $w_{ii} = 0$ and $w_{ij} = w_{ji}$; u_i is the input to the ith neuron; f_i is the sigmoid function of ith neuron.

The Lyapunov function (or energy function) can be defined as :

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}y_iy_j + \sum_{i=1}^{N}\int_0^{y_i} f^{-1}(\xi)d\xi - \sum_{i=1}^{N} u_iy_i \tag{2.22}$$

where ρ_i are positive constants. The time derivative of the above function is given by [53] [55]:

$$\dot{E} = -\sum_{i=1}^{N} T_i(\dot{y}_i(t))^2[\frac{d}{dy_j}f^{-1}(y_j)] \tag{2.23}$$

Since T_i and $(\dot{y}_i(t))^2$ are always positive, and $f(\cdot)$ is monotonically increasing, the inverse output-input relation $f^{-1}(\cdot)$ is also monotonically increasing. $\dot{E}$(t) is always ≤ 0. Therefore, the network solution moves in the same direction as the decrease in energy. The network will eventually reach an equilibrium point.

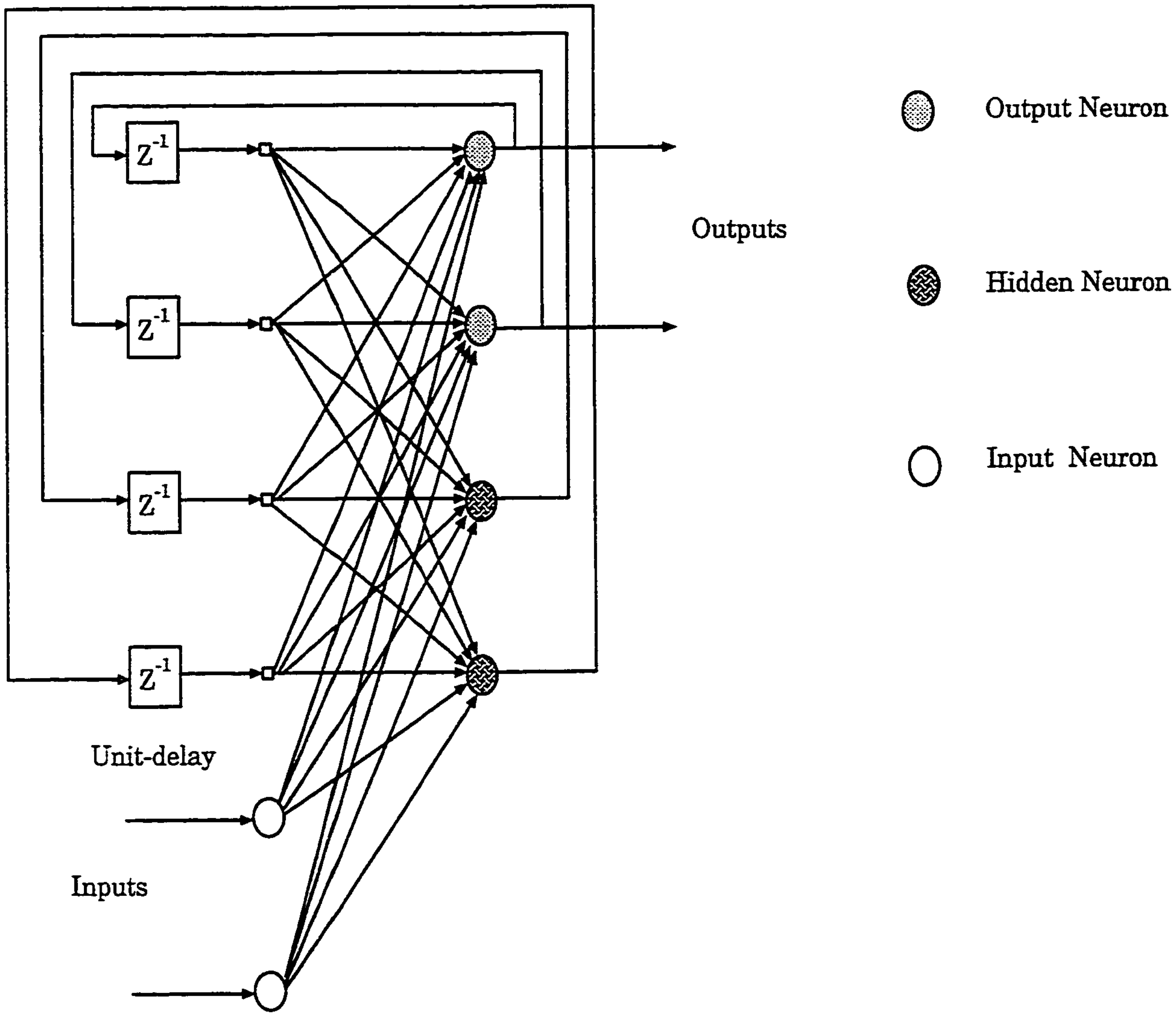


Figure 2.8. Schematic diagram of a real-time recurrent neural network.

2.7 Real-Time Recurrent Neural Network

2.7.1 Architecture

The structure of a real-time recurrent neural network is shown in Fig. 2.8. It differs from the Hopfield network in the previous section in two respects:

- The network contains hidden neurons.

- It has arbitary dynamics.

Let the network have N neurons with M external input connections. Let x(t) denote the M-by-1 external input vector applied to the network at time t, and let y(t) denote the corresponding N-by-1 vector of individual neuron outputs at time t. The input vector x(t) and output vector y(t) are concatenated to form the (M+N)-by-1 vector u(t), whose *i*th element is denoted by u_i(t). Let A denote the set of indices i for which u_i(t) is an external input and let B denote the set of indices i for which u_i(t) is the output of a neuron. Thus,

$$u_i(t) = \begin{cases} x_i(t) & \text{if i} \in \text{A} \\ y_i(t) & \text{if i} \in \text{B} \end{cases} \tag{2.24}$$

The network is fully interconnected in that there are a total of MN forward connections and N^2 feedback connections; N of the feedback connections are in self-feedback connections. Let **W** denote the N-by-(M+N) weight matrix of the network. In order to make provision for a threshold for the operation of each neuron, one input whose value is always -1 is simply included among the M input lines

The network internal activity of neuron j at time t, for j ∈ B, is given by

$$v_j(t) = \sum_{i \in A \cup B} w_{ji}(t) u_i(t) \tag{2.25}$$

where A ∪ B is the union of sets A and B. At the next time step t+1, the output of neuron j is computed by passing v_j(t) through the nonlinearity f(·), obtaining:

$$y_i(t+1) = f(v_j(t)) \tag{2.26}$$

The above two equations constitute the entire dynamics of the network. It is important to note that the external input vector **x**(t) at time t does not influence the output of any neuron in the network until time t+1.

2.7.2 Learning Algorithm

One way to train the weights in this type of recurrent neural network is to convert the network from a feedback system into a purely feedforward system by unfolding the network over time. The network can then be trained as if it is one giant feedforward network. The approach to learning is called backpropagation through time[62]. Another approach, called truncated backpropagation through time [63], tries to approximate the true gradient by only unfolding the network over the last m time steps.

The third approach is the Real-Time Recurrent Learning (RTRL) algorithm, proposed in [64]. This approach enjoys the generality of the backpropagation through time method while not suffering from its large memory requirement in arbitrarily long training sequences. The RTRL algorithm can be summarized as below:

1. Initialize the weights with a set of uniformly distributed random numbers.

2. For every time step t, starting from t = 0, use the dynamic equations of the network, i.e., equations (2.25) and (2.26), to calculate the output values of the N neurons.

3. Let $d_j(t)$ denote the desired response of neuron j at time t, and $\mathcal{C}(t)$ denote the set of neurons that are chosen to act as output neurons. Then the jth element, $e_j(t)$ of the error vector e(t) is

$$e_j(t) = \begin{cases} d_j(t) - y_j(t) & \text{if j} \in \mathcal{C}(\text{t}) \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

The instantaneous sum of squared error at time t can be computed as

$$E(t) = \frac{1}{2}\sum_{j\in\mathcal{C}} e_j^2(t) \tag{2.28}$$

4. The incremental change $\Delta w_{kl}(t)$ at time t is calculated as:

$$\Delta w_{kl}(t) = -\eta \frac{\partial E(t)}{\partial w_{kl}} \tag{2.29}$$

where η is the learning rate parameter. Considering equations (2.27) and (2.28), equation (2.29) can be written as:

$$\Delta w_{kl}(t) = \eta \sum_{j \in C} e_j(t) \frac{\partial y_j(t)}{\partial w_{kl}} \tag{2.30}$$

The partial derivative $\frac{\partial y_j(t)}{\partial w_{kl}}$ can be written as

$$\frac{\partial y_j(t+1)}{\partial w_{kl}} = f'(v_j(t))[\sum_{i \in B} w_{ji}(t) \frac{\partial y_i(t)}{\partial w_{kl}} + \delta_{kl} u_l(n)] \tag{2.31}$$

where δ_{kj} is a *Kronecker delta* equal to 1 when j = k, and zero otherwise. It is natural to assume that the initial state of the network at time t = 0 has no functional dependence on the synaptic weights; therefore;

$$\frac{\partial y_j(0)}{\partial w_{kl}} = 0 \tag{2.32}$$

Equations (2.31) and (2.32) hold for all $j \in B$, $k \in B$, and $l \in A \cup B$.

5. Update the weight w_{kl} in accordance with

$$w_{kl}(t+1) = w_{kl}(n) + \Delta w_{kl}(n) \tag{2.33}$$

and repeat the computation.

One limitation of this algorithm is computational cost at each round.

2.8 Kohonen's Self-Organizing Maps

In a Kohonen self-organizing map (SOM), the neurons are placed at the nodes of a lattice that is usually one or two dimensional. The neurons become selectively tuned to various input patterns (vectors) or classes of input patterns in the course of a competitive learning process. The locations of the neurons so tuned (i.e., the winning

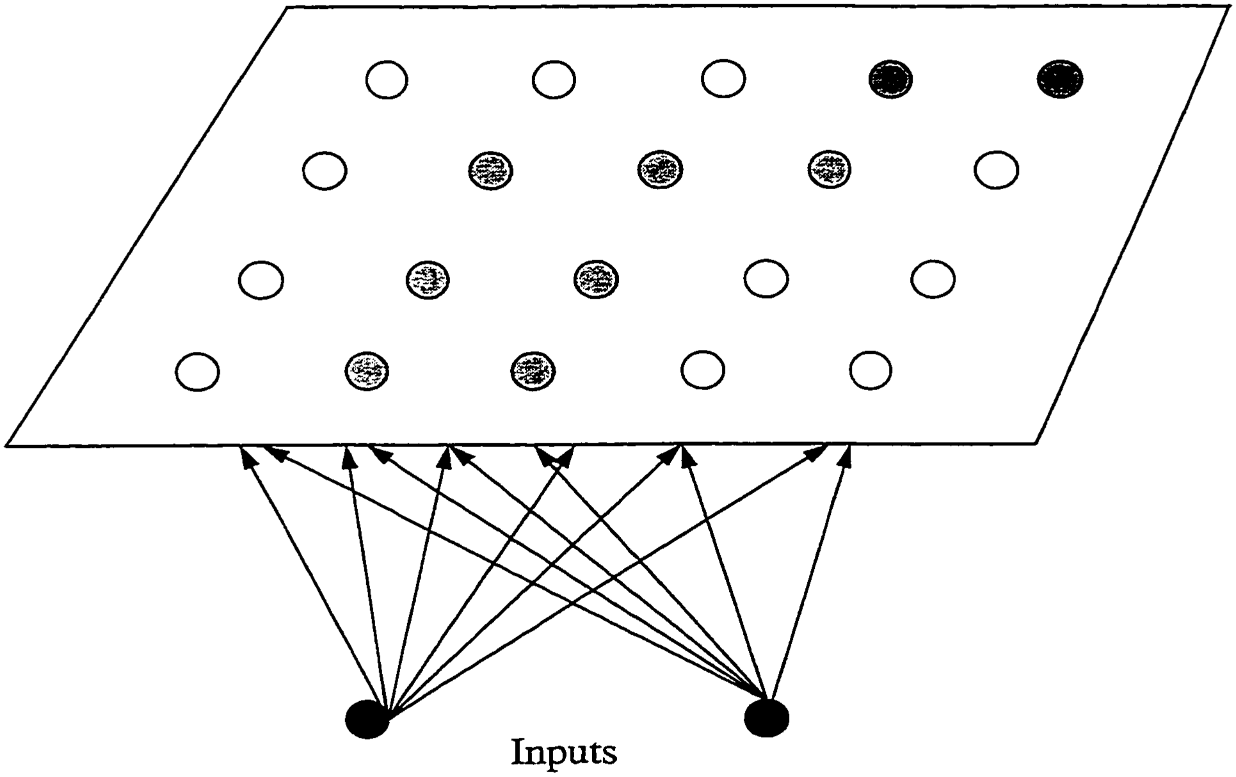


Figure 2.9. Schematic diagram of a Kohonen's Self-Organizing Map.

neurons) tend to become ordered with respect to each other in such a way that a meaningful coordinate system for different input features is created over the lattice. The Kohonen SOM is, therefore, characterized by the formation of a topographic map of the input patterns, in which the spatial locations of the neurons in the lattice correspond to intrinsic features of the input patterns. The structure of a Kohonen SOM is shown in Fig. 2.9. The different shaded cells indicate how the features of input patterns are represented by Kohonen KOM neurons.

The learning algorithm of Kohenon SOM can be summarized as follows:

1. *Initialization.* Choose random values for the initial weight vectors $\mathbf{w}_i(0)$. Select initial learning rate and neighborhood function.

2. *Sampling.* Present a pattern $\mathbf{x}$ and evaluate the network outputs.

3. *Similarity Matching.* Find the best-maching (winning) neuron i(x) at time t using the minimum-distance Euclidean criterion:

$$i(\mathbf{x}) = \min_j \|\mathbf{x}(t) - \mathbf{w}_j\|.j = 1, 2, ..., N \tag{2.34}$$

4. *Updating.* Adjust the weights vectors of all neurons according to the following learning rule: if j $\in N_i$(t), then

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t)[\mathbf{x}(t) - \mathbf{w}_j(t)] \tag{2.35}$$

   otherwise:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) \tag{2.36}$$

   where $\eta(t)$ is the learning rate parameter, and N_i(t) is the neighborhood function centered around the winning neuron i(x). Both $eta(t)$ and N_i(t) are varied dynamically during learning procedure. In [49], it turned out to be advantageous to let N_i be very wide at the beginning and shrink monotonically with time.

5. *Continuation* Repeat Step 2 through 4 until the change in weight values is less than a prespecified threshold or a maximum number of iterations is reached.

2.9 Suitability of Different Neural Networks in Control System Application

The multilayer perceptron (MLP) is the most popular neural network structure employed in control application [65] [66] [67] [68] [69] [70] [71] [72] [73] . One of the MLP capabilities is to implement nonlinear transformations for functional approximation problems. The backpropagation algorithm also provides a *computationally efficient* method for the traning of multilayer perceptrons. In control applications, the multilayer perceptrons are reported in:

- adaptive control using generic index,

- adaptive filtering and prediction
- adaptive non-linear control
- non-linear modeling
- gain scheduling

The self-organising neural network has been employed in robot control[74] [75]. In [74], the self-organising network first adaptively discretizes the input space and learns the linear mapping from input space to output space for each discretized cell, then tunes the action space weights associated with only the winning neuron. However, the neural controller is used in conjunction with another controller.

The ability of Hopfield network to approximate dynamic systems, in continuous and discrete time, has been shown respectively in [76] and [77]. The Hopfield network can also be used as a part of the adaptation mechanism [78] for a linear system. In this case, the output of the network represents the parameters of the linear model. The major limitation of hopfield network is that the number of patterns of the control outputs that can be stored and accurately recalled is severely limited. It has been shown that the number of classes M must be less than 15% of the number of nodes N[79]. For a complex nonlinear system like generating unit, a large number of nodes in a Hopefield network may be required.

In recent years, there has been a growing interest in using recurrent neural network for modeling and controling non-linear systems. The reason is that time is an important factor in non-linear system control, time needs to be represented by the effect it has on the signal processing. This means that neural networks should have dynamic properties that make it responsive to time-varying signals. However the representation and processing of temporal information is not an intrinsic capability of static neural networks as they are memory-less. For the recurrent neural networks, their outputs are a function not only of the current inputs, but also past inputs or outputs, they have important capability to deal with time varying input or output through their own natural temporal operation. The recurrent neural network can also be trained by supervised learning method.

Until now, most research of neural network application in PSS design is based on multilayer perceptrons. In this dissertation, the recurrent neural network with real time recurrent learning algorithm is employed to design an adaptive neural netwrok control system.

2.10 Summary

In this chapter, the basic concepts and theories of neural networks are introduced. The basic element of a neural network, the neuron, has three components: a set of synapses, an adder and an activation function. According to the connectivities of neurons, neural networks can be classified into two groups: feed-forward and recurrent (or feedback). Three learning paradigms: supervised learning, unsupervised learning and reinforcement learning, are introduced. Detailed discussion has been given of four basic learning rules: error-correction learning, Boltzmann learning, Hebbian learning and competitive learning.

Four types of neural networks, the multilayer perceptrons, the Hopfield network, the real-time recurrent network and Kohonen SOM, are discussed in this chapter.

The multilayer perceptron is a static network. The input signal propagates through the network in a forward direction on a layer-by-layer basis. The development of the back-propagation learning algorithm for determining the weights in a multilayer perceptron has made multilayer perceptrons very popular.

Hopfield network is a dynamic network. A network energy function is used to design and analyse its dynamic behavior. Hopfield network always evolves in the direction that leads to lower network energy. This implies that if combinatorial optimization problem can be formulated as minimizing an energy function, the Hopfield network can be used to find optimal (or suboptimal) solution.

The real-time recurrent network is a feedback network with hidden neurons. It is trained with a real-time recurrent learning algorithm. This network has a temporal processing capability through feedback built into its design.

Kohonen's self-organizing map is a special type of competitive learning network that defines a spatial neighborhood for each output unit. During the competitive learning, all the weight vectors associated with the winner and its neighboring units are updated. Kohonen's SOM can be used for projection of multivariate data, density approximation, and clustering.

By comparing the features of the above neural networks and their status in present research, the real-time recurrent network is selected to design the neural network based PSS controller in this dissertation.

CHAPTER 3

NEURAL NETWORK IDENTIFICATION AND CONTROL

3.1 Introduction

The power system is a complex system characterized with terms such as non-linear, large-scale, dynamical, discrete, stochastic, random-like, time-variant parameter system, etc. Among these terms, the non-linear and the large-scale aspects of a system make the power system problems hard to solve. In the earlier years of power system stabilizer research, the power system was treated as linear time-invariant system, then analysized by well established techniques based on linear algebra, complex variable theory, and the theory of ordinary linear differential equations. However, it is difficult to obtain satisfactory solutions.

Many efforts have been made to develope techniques for the control of systems with complex, unknown and non-linear dynamics. Several approaches have been proposed: adaptive control[80][81], Liapunov-based control[82][83] robust control[84][85] and optimal control[86][87]. Advances in the area of artificial neural networks have provided the potential of new approaches to the control of such systems. It is well known that the ability to learn, to perform nonlinear functional approximation, and to perform massive parallel processing are main advantages that make neural networks so attractive.

Neural network control represents a unique methodology by which the knowledge is acquired from sets of training examples and stored in a distributed manner in the connectionist structure of the network. The distributivity contributes to increased learning capablities of neural networks because the individual elements in the network are capable of adjusting their connections to achieve near-optimal input-output mappings. Distributed learning is also advantageous because it permits a response to a novel situation to be inferred using knowledge previously learned, in similar but not exactly the same circumstances. Therefore, the neural network is very suitable for accommodating a poorly modelled, non-linear dynamical system. It has been demonstrated in [65] [66] [72] [73] that neural networks can be used effectively for the identification and control of non-linear dynamical system.

In this chapter, a PSS based on neural networks using an indirect adaptive control method, is presented. There are two subnetworks in the controller architecture. The first one acts as a neural network identifier which tracks the dynamic properties of the plant. The second one functions as a neural network controller which provides the necessary control signal to the plant.

3.2 Plant Model

A plant model may be required for several reasons:

- to use within a larger feedback control loop which requires an estimate of the plant output;
- for predicting the performance of the plant when the true output is unavailable;
- for fault diagnosis;
- for use within an off-line controller design strategy.

In this section four models for the representation of Single-Input Single-Output (SISO) plants are described. These models can also be generalized to multivariable case.

Model I:

$$y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-i) + g[u(k), u(k-1), \ldots, u(k-m+1)] \tag{3.1}$$

Model II:

$$y_p(k+1) = f[y_p(k), y_p(k-1), \ldots, y_p(k-n+1)] + \sum_{i=0}^{m-1} \beta_i u(k-i) \tag{3.2}$$

Model III:

$$\begin{aligned} y_p(k+1) &= f[y_p(k), y_p(k-1), \ldots, y_p(k-n+1)] + \\ &\quad g[u(k), u(k-1), \ldots, u(k-m+1)] \end{aligned} \tag{3.3}$$

Model IV:

$$\begin{aligned} y_p(k+1) &= f[y_p(k), y_p(k-1), \ldots, y_p(k-n+1); \\ &\quad u(k), u(k-1), \ldots, u(k-m+1)] \end{aligned} \tag{3.4}$$

where:

- $[u(k), y_p(k)]$ are the input-output pair of the SISO plant at time k, and $m \leq n$.
- The functions f: $\Re^n \rightarrow \Re$ in Models II and III; f: $\Re^{n+m} \rightarrow \Re$ in Model IV, and g: $\Re^m \rightarrow \Re$ are assumed to be differential functions of their arguments.

In all four models, the output of the plant at time (k+1) depends both on its past n values, $y_p(k-i)$ $(i = 0, 1, \ldots, n-1)$, as well as the past m values of the input $u(k-j)$ $(j = 0, 1, \ldots, m-1)$. Dependence on the past values $y_p(k-i)$ is linear in Model I while in Model II the dependence on the past values of the input $u(k-j)$ is assumed to be linear. In Model III, the non-linear dependence of $y_p(k+1)$ on $y_p(k-i)$ and $u(k-j)$ is assumed to be seperable. Model IV is the generalized model for Models I-III. A generating unit is non-linear dynamic system. Model IV with multivariables is apparently more suited for plant model in this study.

3.3 Neural Network for Model Identification

The problem of identification includes setting up a suitably parameterized identification model and adjusting the parameters of the model to optimize a performance function based on the error between the plant and the identification model outputs.

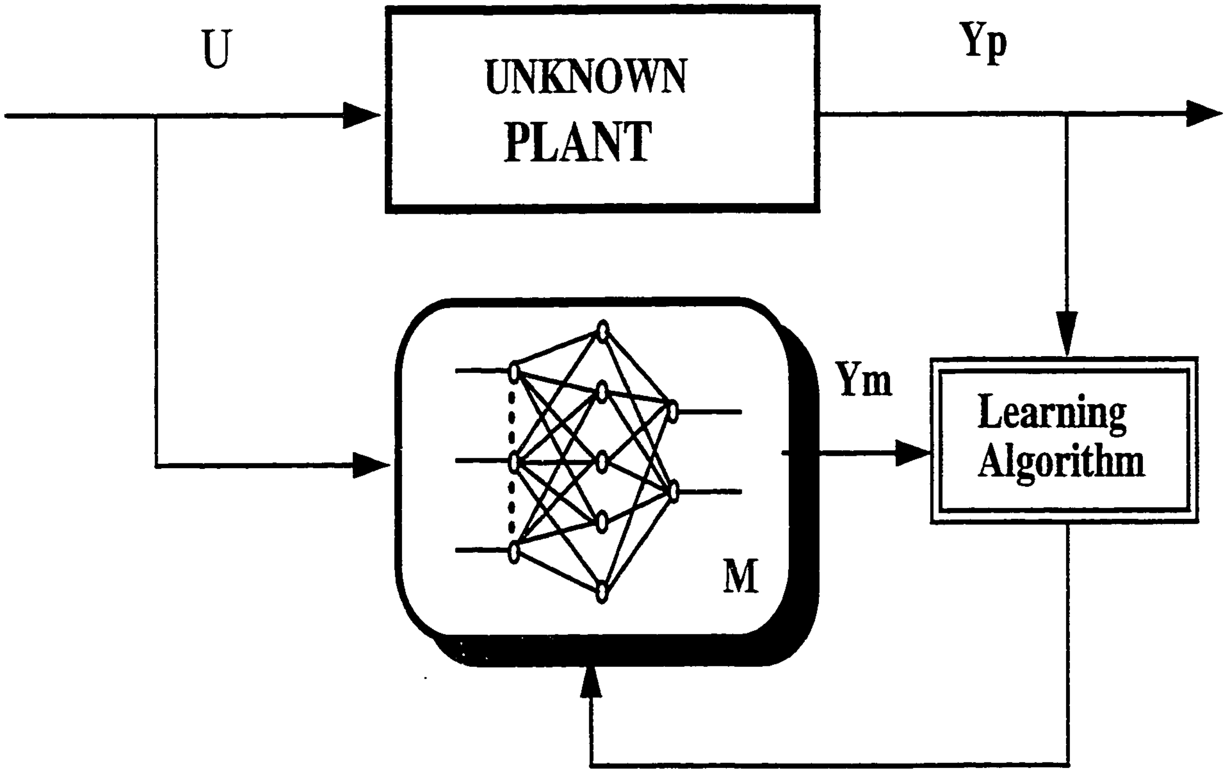


Figure 3.1. Schematic diagram of forward modelling.

3.3.1 Forward Modeling

A structure for forward modeling is shown in Fig. 3.1. The neural network model is placed in parallel with the system and the error between the system and network output is used as the neural network training signal. This learning structure is a classical supervised learning problem where the plant (acting as a teacher) provides target values (its outputs) directly in the output coordinate system of the learner (i.e.. the neural network model).

If it is assumed that the system is governed by the following equation:

$$y_p(k+1) = f[y_p(k),\ldots,y_p(k-n+1);u(k),\ldots,u(k-m+1)] \tag{3.5}$$

Then, denoting the output of the network as y_m, gives the following equation for neural model:

$$y_m(k+1) = \hat{f}[y_p(k),\ldots,y_p(k-n+1);u(k),\ldots,u(k-m+1)] \tag{3.6}$$

where $\hat{f}$ represents the non-linear input-output map of the neural network. If after a suitable training period the neural network gives a good representation of the plant (i.e., $y_m \approx y_p$), then for subsequent post-training purposes, the neural network model can be described as:

$$y_m(k+1) = \hat{f}[y_m(k),\ldots,y_m(k-n+1);u(k),\ldots,u(k-m+1)] \tag{3.7}$$

and the neural network can be used independently of the plant.

3.3.2 Inverse Modeling

The objective of the inverse modeling is to formulate a controller such that the overall controller/plant architecture has a *unity* transfer function. An inverse modeling structure is illustrated in Fig. 3.2. Here the control signal is performing as a training signal. The neural network output is compared with the training signal and this error is used to train the network. This structure will tend to force the network to represent the inverse of the plant.

The non-linear input-output relation of the network modelling the plant inverse can be described as:

$$u(k) = \hat{f}^{-1}[y_p(k+1),\ldots,y_p(k-n+1);u(k-1),\ldots,u(k-m+1)] \tag{3.8}$$

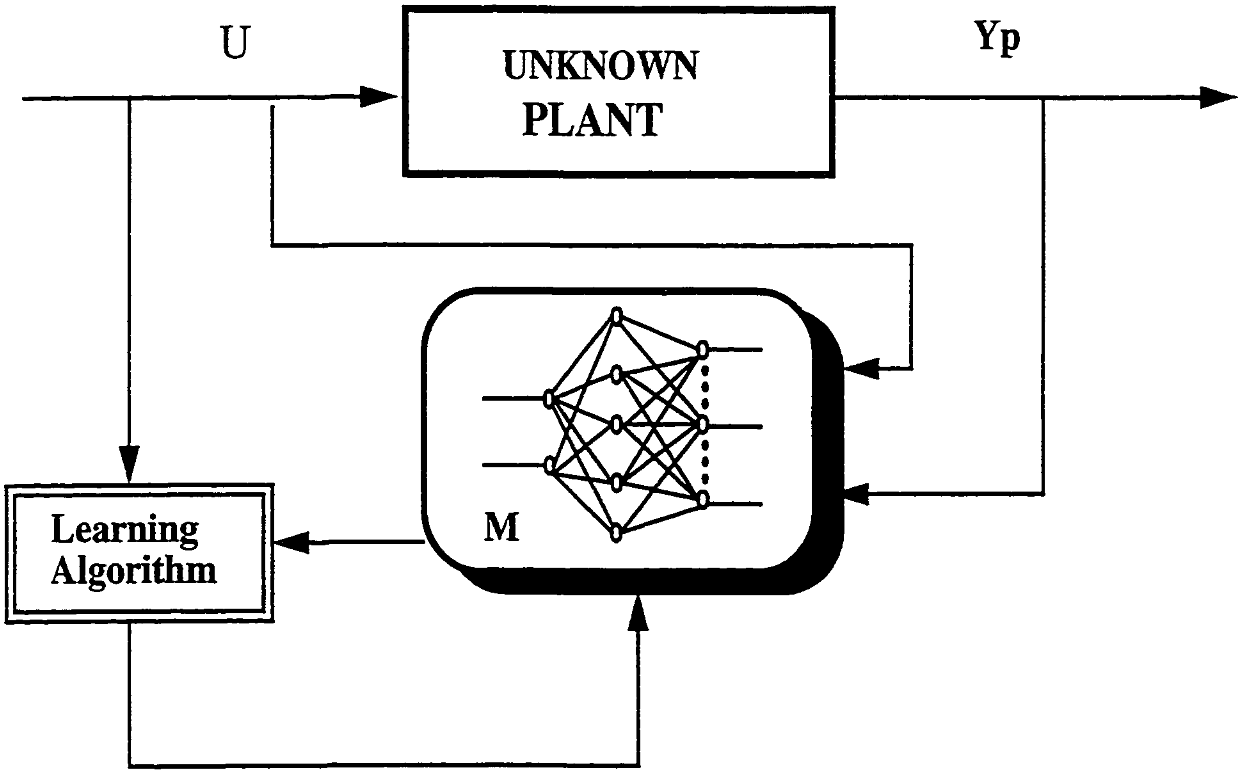


Figure 3.2. Schematic diagram of inverse modeling.

Since $y_p(t+1)$ is not available at the time to calculate $u(t)$, the solution to overcome this problem will depend on the plant modelled.

For the inverse model to be well defined, the training examples must be unique. This is satisfied when the plant is invertible or if the training data for a non-invertible plant are contained in a restricted area of the input space so that the plant is locally invertible. However, if the nonlinear system cannot be mapped one on one under any consideration, then an incorrect inverse can be obtained.

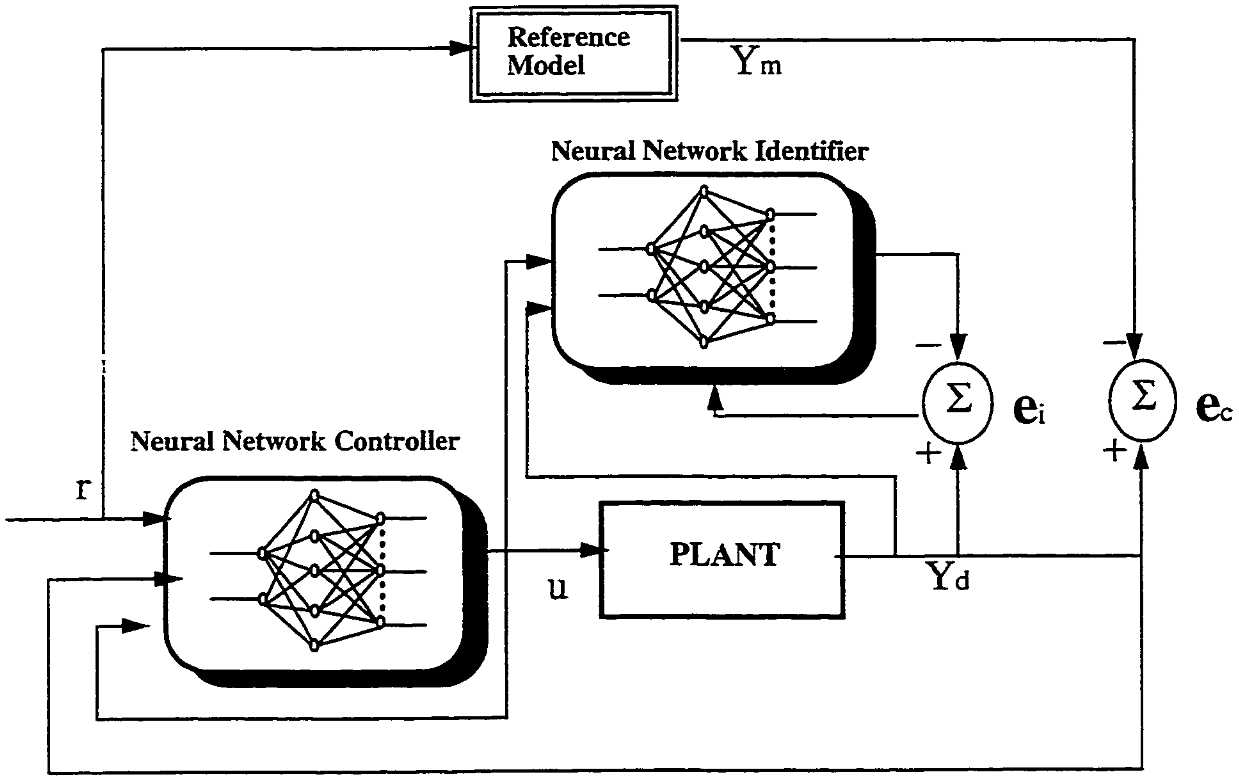


Figure 3.3. Schematic diagram of model reference control.

3.4 Neural network for Control

3.4.1 Model Reference Control

The model reference learning control architecture, shown in Fig. 3.3, has been previously widely used in the linear adaptive control field. Here, the desired performance of the closed-loop system is specified through a stable reference model which is defined by its input-output pair { r(k), y_r(k)}. The control system attempts to make the plant output y_p(k) match the reference model output asymptotically by

$$\lim_{n\to\infty} \| y_r(k) - y_p(k) \| \leq \varepsilon \tag{3.9}$$

where ε is a small positive constant. The performance of this control structure depends on the choice of a suitable reference model and the derivation of an appropriate learning machanism.

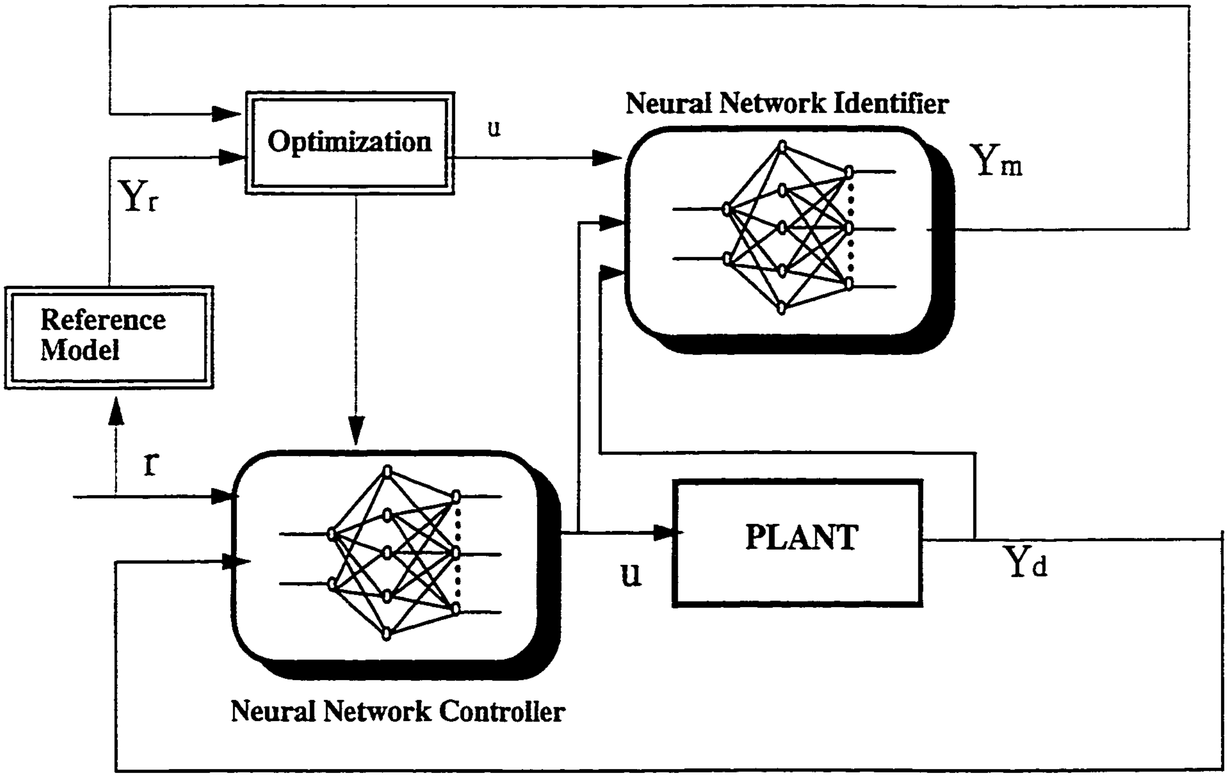


Figure 3.4. Schematic diagram of predictive learning control.

3.4.2 Predictive Learning Control

A predictive learning control architecture is shown in Fig. 3.4. In this approach a neural network model provides prediction of the future plant response over the specified horizon. The predictions supplied by the network are passed to a numerical optimization routine which attempts to minimize a specified performance criterion in the calculation of a suitable control signal.

The control signal u is chosen to minimize the quadratic performance criterion, J, subject to the constraint of the dynamical model,

$$\begin{aligned} J &= \sum_{j=N_1}^{N_2} [y_r(k+j) - y_m(k+j)]^2 \\ &\quad + \sum_{j=1}^{N_2} \lambda_j [u(k+j-1) - u(k+j-2)]^2 \end{aligned} \tag{3.10}$$

where the constants N_1 and N_2 define the horizons over which the tracking error and control increments are considered. The values of λ are the control weights.

An advantage of this approach is that the outer loop, consisting of plant model and optimization routine, is no longer needed when training is complete.

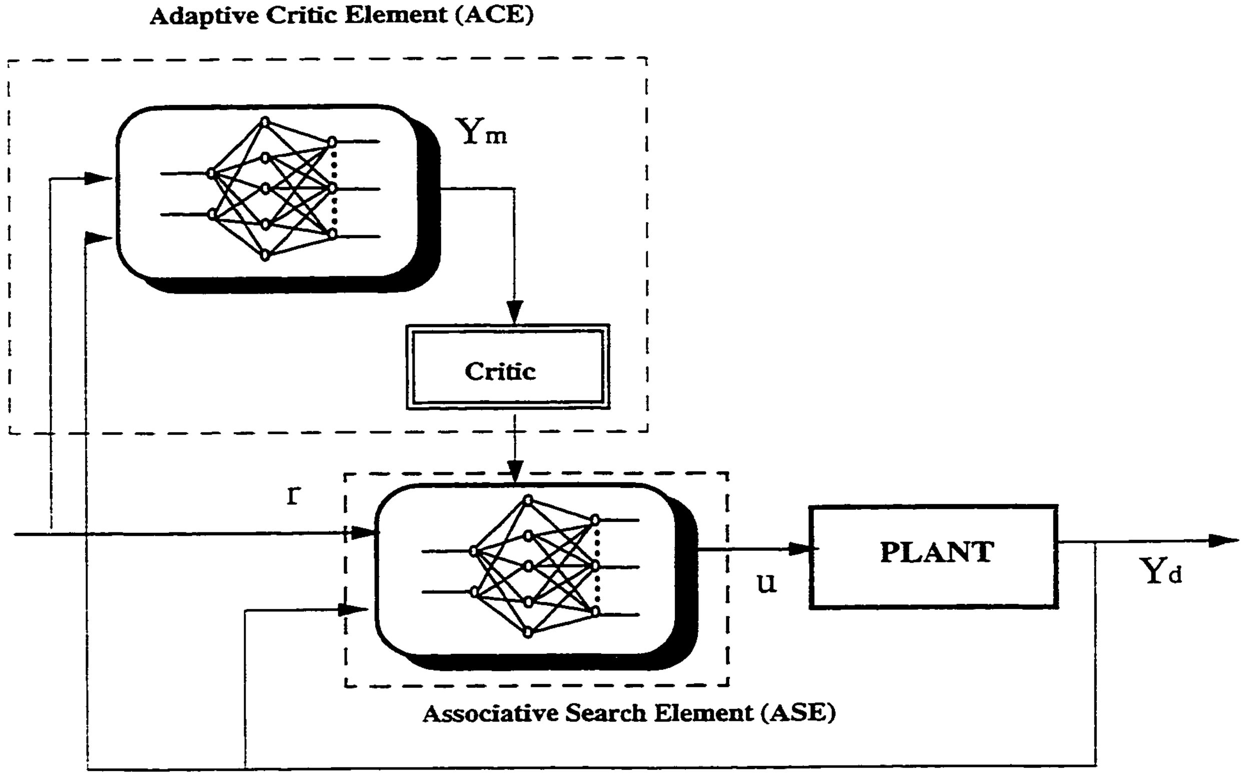

Adaptive Critic Element (ACE)
Ym
Critic
r
u
PLANT
Yd
Associative Search Element (ASE)

Figure 3.5. Schematic diagram of reinforcement learning systems.

3.4.3 Reinforcement learning Systems

Reinforcement schemes are minimally supervised learning algorithms. The only information that is made available is whether or not a particular set of control actions has been successful. A control scheme which is composed of two adaptive elements is shown in Figure 3.5: an Associative Search Element (ASE) and an Adaptive Critic Element (ACE). The ASE attempts to reproduce the optimal control signal which satisfies the given performance objectives; while the ACE attempts to monitor the performance of the controller internally and to provide an internal reinforcement signal which is used to train the ASE. The ACE is trained using the external failure/success signal. The continuous internal training of the control element has been shown to improve vastly the performance of the overall system.

3.4.4 Direct inverse Control

Direct inverse model is simply cascaded with the controlled system in order that the composed system results in an identity mapping between desired response and the controlled system output. Direct inverse control is also based on the assumption that there exits a one-to-one mapping from the input state to output state. Problems will be encountered if the mapping from control inputs to plant output is not invertible.

3.5 Proposed Adaptive Controller Based on Neural Network

Due to the nonlinear, time varying and stochastic nature of the power system, it is essential for a controller to possess the ability to change its own behaviour according to the changes of the controlled system. Using adaptive control techniques to design power system stabilizer has drawn attention from both academy and industry. In this section, an adaptive PSS based on recurrent neural network is presented.

The architecture of the proposed controller is illustrated in Fig. 3.6. It includes two subnetworks. The first one functions as a neural identifier which will track the dynamic activity of the non-linear plant and will be a channel for back propagation to train the controller network. The second one acts as a neural controller to provide proper control signal to the plant. The controller structure is very similar to that of the reference model control mentioned before. However, there is a little difference.

No reference model is needed for this control scheme since for a complex system such as power system it is difficult to obtain a proper reference model.

3.5.1 Neural Identifier

First. it is needed to select a proper model of the plant to be identified. The plant model is considered in format of Model IV mentioned in Section 3.2, i.e.

$$\begin{aligned} y_p(k+1) &= f[y_p(k), y_p(k-1), \ldots, y_p(k-n+1); \\ &\qquad u(k), u(k-1), \ldots, u(k-m+1)] \end{aligned} \tag{3.11}$$

The forward modeling method discussed in Section 3.3.1 is used, and a recurrent neural network is trained to assume the nonlinear function $f()$ which represents the dynamic relationship of the unknown plant.

For the neural-identifier. the input vector is:

$$[y_1(k), \ldots, y_1(k-m), \ldots, y_j(k), \ldots, y_j(k-n), u(k), \ldots, u(k-p)] \tag{3.12}$$

where $y_1(k)$. $y_j(k)$ are the plant output and $u(k)$ is the plant input (controller output) at the k-th time step. The output is $\hat{y_1}(k+1)$. the predicted plant output at time step (k+1). Then the performance index of the ANN identifier is :

$$J_i(k) = \frac{1}{2}[\hat{y_1}(k) - y_1(k)]^2 \tag{3.13}$$

The weights in the neural-identifier are updated as below:

$$\mathbf{W}_i(k) = \mathbf{W}_i(k-1) - \eta_i \nabla_{W_i} J_i(k) \tag{3.14}$$

where $\mathbf{W}_i(k)$ is the matrix of neural identifier weights at instant k, η_i is the learning rate for neural-identifier, and $\nabla_{W_i} J_i(k)$, the instantaneous gradient, is calculated by

$$\nabla_{W_i} J_i(k) = -[y_1(k) - \hat{y_1}(k)] \frac{\partial \hat{y_1}(k)}{\partial W_i(k)} \tag{3.15}$$

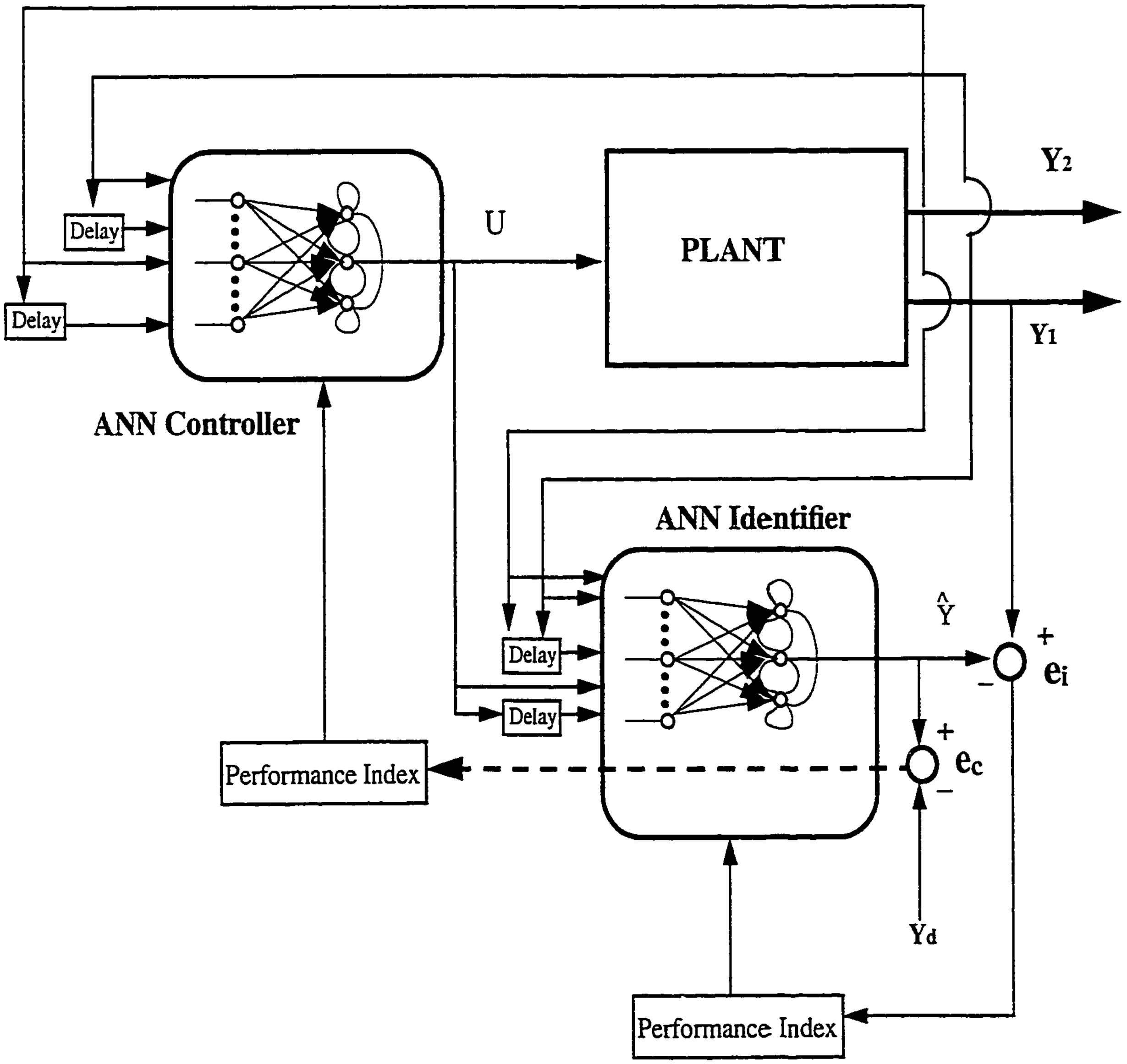


Figure 3.6. Schematic diagram of Proposed Neural Network Controller.

At each sampling instant, the inputs and outputs of the generating unit are sampled, and a mathematical model of the plant is represented by a recurrent neural network which updates its weights every sampling period. With this neural identifier, it is expected that the dynamic behaviour of the generating unit can be tracked.

3.5.2 Neural Controller

For the neural-controller, the input vector is,

$$[y_1(k), \ldots, y_1(k-m), \ldots, y_j(k), \ldots, y_j(k-n)] \tag{3.16}$$

The output is $u(k)$, the control signal at time step k. The $u(k)$ is sent to the system and the neural identifier simultaneously. The performance index of the neural controller is:

$$J_c(k) = \frac{1}{2}[\Delta\hat{y}_1 e(k) - y_{1d}(k)]^2 \tag{3.17}$$

where $y_{1d}(k)$ is the desired plant output at time step k. In this study, it is set to be zero. The weights in the neural-controller are updated as below:

$$\mathbf{W}_c(k) = \mathbf{W}_c(k-1) - \eta_c \nabla_{W_c} J_c(k) \tag{3.18}$$

where $\mathbf{W}_c(k)$ is the matrix of neural controller weights at instant k, η_c is the learning rate for neural-controller, and $\nabla_{W_c} J_c(k)$, the instantaneous gradient, is calculated by

$$\nabla_{W_c} J_c(k) = \hat{y}_1(k)[\frac{\partial \hat{y}_1(k)}{\partial u(k)}]\frac{\partial u(k)}{\partial W_c(k)} \tag{3.19}$$

For the neural controller, the desired output cannot be defined explicitly. Therefore, the neural controller has to be trained by driving the neural identifier to generate the equivalent error. The neural identifier will back-propagate the error between the desired and predicted plant output to update the weights of neural controller.

3.5.3 Tranining Methods

One of the important aspects in neural network application is to properly train the neural network. The training of the proposed adaptive neural network controller has two steps, off-line training and on-line update. In the off-line training, the structure of the neural network, i.e. the number of layers and the number of neurons will be decided. The weights of the neural network will be adjusted to proper value based on the traning patterns. The training patterns in this study will cover a wide range of operating conditions and a number of possible disturbances for the power system. After the off-line training is finished, the proposed controller will be connected to the power system for on-line update. The parameters of the neural identifier and controller will be adjusted every sampling period. This allows the controller to track the dynamic variations of the power system and provide the best control action.

The online updating algorithm of the proposed power system stabilizer can be summarized as follows:

- At k-th time step, $y_1(k), y_2(k)$ are sampled.
- $y_1(k), y_2(k)$ are used to form input vector of neural-controller at k-th time step. The output $u(k)$ is calculated. At the same time, the weights of the neural-identifier are updated to minimize the error between $y_1(k)$, and $\hat{y_1}(k)$
- $y_1(k), y_2(k)$ and $u(k)$ are used to form an input vector for the neural-identifier, and the temporal output $\hat{y_1}(k+1)$ is calculated.
- The weights of the neural-controller are updated to minimize $[\hat{y_1}(k+1)]^2$
- The output of the neural-controller $u(k)$ is calculated again with the same input vector and the new weights calculated in the previous step.

- The control signal $u(k)$ is applied to the plant, and to the neural-identifier again to calculate $\hat{y}_1(k+1)$ for (k+1)-th time step .

In every sampling period, the learning is done only once for both the neural identifier and the neural controller. This simplifies the training algorithm in terms of computation time.

3.6 Summary

In this chapter, various plant models are introduced first. Then two schemes for using neural networks for identification are discussed and various neural network architectures for the control of non-linear system are presented. Based on the discussion, an adaptive controller based on neural networks is presented. The proposed controller consists of two subnetworks; a neural network identifier and a neural network controller. The two neural networks are first trained off-line and then their parameters are updated during on-line operation. The on-line update algorithm allows the proposed controller to track the dynamic behaviour of the controlled plant and provide the proper control action. Finally, no reference model is needed for the proposed controller.

CHAPTER 4

SIMULATION STUDY OF ADAPTIVE RNN PSS IN A SINGLE MACHINE INFINITE BUS SYSTEM

4.1 Introduction

Studies in the past four decades have shown that a power system stabilizer (PSS) is a very effective tool to damp the low frequency oscillations in the power system. Since power system is a highly non-linear dynamic system, to design a PSS which can keep desired performance under different operating conditions is still a topic that needs more investigation. Recently new techniques such as artificial neural networks and fuzzy logic theory have been introduced to the design of PSS.

The conventional PSS is usually designed under a particular operating condition, and its parameters are fixed at the end of the design procedure. It works well within a certain range around the design point. When there is a drastic change in the operating conditions, the conventional PSS may no longer provide fully satisfactory damping effects.

Power systems are nonlinear and operate over a wide range. They are subject to random load changes and the network structure changes from time to time. Therefore, it is desired to develop a stabilizer which has the ability to adjust its own parameters on-line according to the environment in which it works to provide satisfactory control performance. This idea leads to the research and development of power system stabilizer with adaptivity.

With their various advantages such as: *nonlinearity, input-output mapping, adaptivity, fault tolerance*, the artificial neural networks have become a useful tool in the control area since the late 1980s . Especially, since in recurrent neural networks their neuron outputs have relation not only with present inputs but also with past inputs or past outputs, i.e. they have dynamic behaviour, there has been a growing interest in using recurrent neural networks for modeling and control of nonlinear dynamic systems in the past few years [88] [89] [90] [91] [92] [93] [94] [95] [96].

Artificial neural networks have also been introduced in the design of PSS. Several papers that applied the artificial neural network in the PSS design have been published [37] [38] [39] [40] [41] [42] [43] [44] [45] [47] .Successful digital simulation results in these papers show that neural networks have a considerable potential in power system stability study. However, the feedforward Multi-layer Perceptron (MLP) is the most common neural network employed in current research.

Application of the adaptive control structure proposed in Section 3.5 as an adaptive PSS based on Real-Time Recurrent Neural Network(RTRNN) is presented in this chapter. In the proposed control system architecture, there are two neural networks which act as an identifier and a controller, respectively. The training of ANN PSS has two stages: off-line training and on-line update. So the recurrent neural network PSS (RNN PSS) is not designed for a specific operating point. It can track the variations of operating condition of the power system. The performance of the proposed ANN PSS under different perturbations and operating conditions is investigated in the system of one machine connected to an infinite bus through a double circuit transmission lines.

4.2 Design of the RNN PSS

The structure of the proposed PSS is shown in Fig. 4.1. It consists of two components: a control neural network to output the suitable control signal and an identifier neural network to emulate the system behaviour.

For the neural identifier, the input vector is: $(\Delta\vec{\omega}(k), \Delta\vec{P}e(k), \vec{U}_{annc}(k))$. The output is $\Delta\hat{\omega}(k+1)$. Then the performance index of the RNN identifier is :

$$J_i(k) = \frac{1}{2}[\Delta\hat{\omega}(k) - \Delta\omega(k)]^2 \tag{4.1}$$

The weights in the RNN identifier are updated as below:

$$\mathbf{W}_i(k) = \mathbf{W}_i(k-1) - \eta_i \nabla_{W_i} J_i(k) \tag{4.2}$$

where

- $\Delta\omega(k)$ is the generator speed deviation, and $\Delta\hat{\omega}$ is estimated value at instant k,
- $\Delta Pe(k)$ is the change of electrical power,
- $U_{annc}(k)$ is the control signal output,
- $\Delta\vec{\omega}(k) = [\Delta\omega(k), \Delta\omega(k-1), \ldots, \Delta\omega(k-m)]$
- $\Delta\vec{P}e(k) = [\Delta Pe(k)\Delta Pe(k-1), \ldots \Delta Pe(k-n)]$
- $\vec{U}_{annc}(k) = [U_{annc}(k), U_{annc}(k-1), \ldots, U_{annc}(k-p)]$,
- $\mathbf{W}_i(k)$ is the matrix of the RNN identifier weights at instant k,
- $\nabla_{W_i} J_i(k)$ is the instantaneous gradient,
- η_i is the learning rate for the RNN identifier.

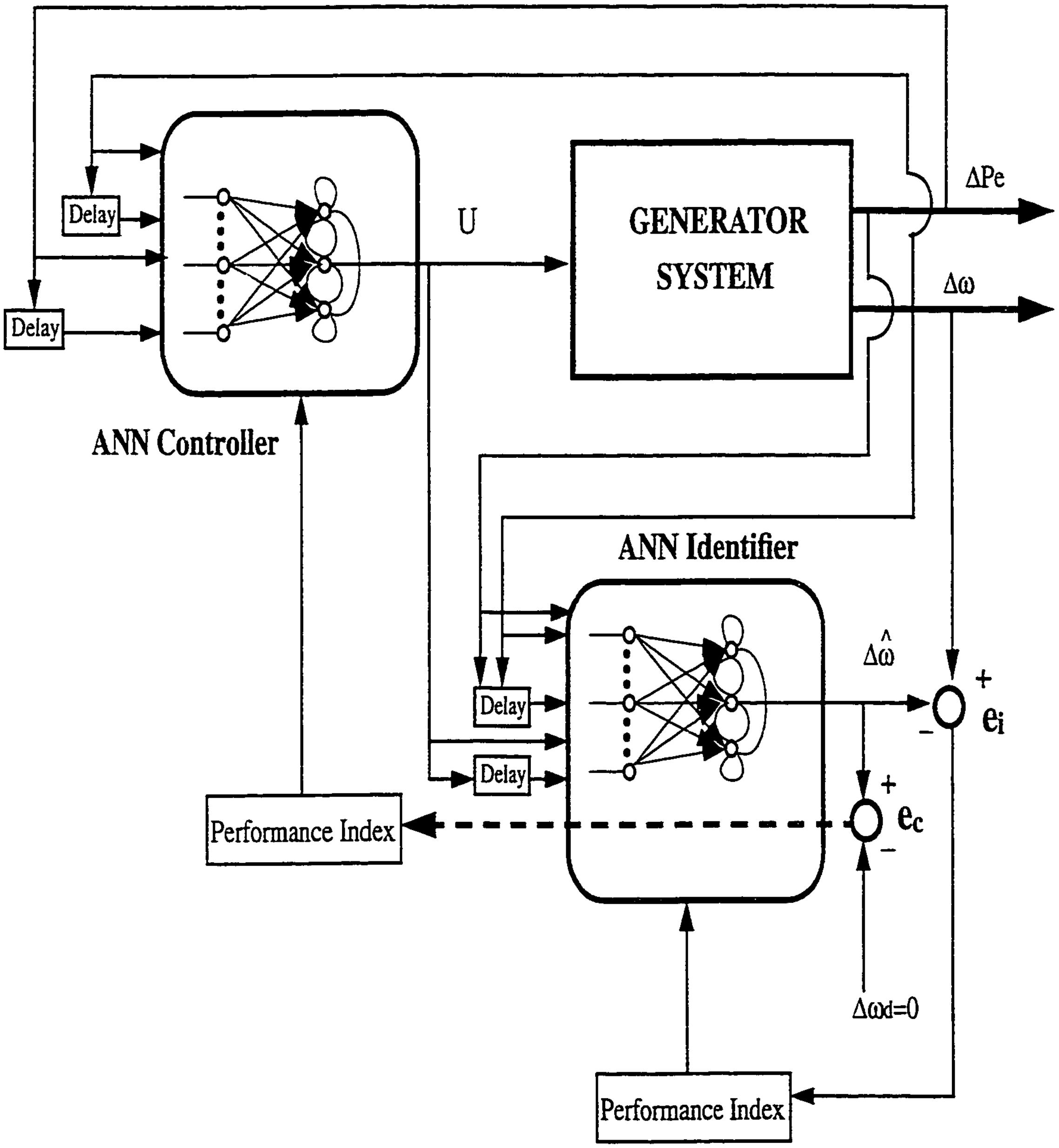


Figure 4.1. Structure of the control system

For the neural-controller, the input vector is $(\Delta\bar{\omega}(k), \Delta\vec{P}e(k))$. The output is $U_{annc}(k)$. The performance index of the RNN controller is:

$$J_c(k) = \frac{1}{2}[\Delta\hat{\omega}(k+1)]^2 \tag{4.3}$$

The weights in the RNN identifier are updated as follows:

$$\mathbf{W}_c(k) = \mathbf{W}_c(k-1) - \eta_c \nabla_{W_c} J_c(k) \tag{4.4}$$

where

- $\mathbf{W}_c(k)$ is the matrix of RNN controller weights at instant k,
- $\nabla_{W_c} J_c(k)$ is the instantaneous gradient,
- η_c is the learning rate for RNN controller.

Since the output of control neural network has no direct connections to $\Delta\hat{\omega}(k+1)$, the identifier neural network is acting as a bridge to back-propagate $\Delta\hat{\omega}(k+1)$ to the control neural network. In this process, the weights of the identifier neural network are treated as constants.

The algorithm of the proposed power system stabilizer can be summarized as follows:

- At k-th time step, $\Delta\omega(k), \Delta Pe(k)$ are sampled.
- $\Delta\omega(k), \Delta Pe(k)$ are used to form input vector of neural-controller at k-th time step. The output $U_{annc}(k)$ is calculated. At the same time, the weights of the neural-identifier are updated to minimize the error between $\omega(k)$ and $\hat{\omega}(k)$
- $\Delta\omega(k), \Delta Pe(k)$ and $U_{annc}(k)$ are used to form an input vector for the neural-identifier, and the output $\Delta\hat{\omega}(k+1)$ is calculated.
- The weights of the neural-controller are updated to minimize $[\Delta\hat{\omega}(k+1)]^2$

- The output of the neural-controller $U_{annc}(k)$ is calculated again with the same input vector and the new weights calculated in the previous step.

- The control signal $U_{annc}(k)$ is applied to the plant, and to the neural-identifier again to calculate $\Delta\hat{\omega}(k+1)$ for (k+1)-th time step .

4.3 Neural Network Training

The weights of the RNN controller and identifier are first trained off-line. During the off-line training procedure, the following parameters are decided:

(1) m,n,p. These were selected as 4 based on off-line simulation studies. This means that , for $\Delta\omega(k)$, $\Delta Pe(k)$ and $U_{annc}(k)$, three orders of delay are used as inputs. Higher order of delays have been tested, however, it does not improved the performance. Therefore, the ANN controller has 8 and RNN identifier has 12 input neurons, respectively.

(2) The hidden neuron numbers, three, are assigned to networks at beginning, then the numbers are increased gradually until 12. The neural networks performance after training are compared. It is found that 5 and 7 hidden neurons in RNN controller and RNN identifier respectively provide best performance. Further increasing the numbers of hidden neurons provides almost the same results but needs computation.

(4) The weights of the RNN controller and the RNN identifier are set randomly at ±0.03 and ±0.04, respectively.

The training data for off-line studies was collected from a wide generator operating range: the generator output ranging from 0.1 p.u. to 1.0 p.u., the power factor ranging from 0.8 lead to 0.85 lag. Disturbances, such as input torque step change (± 0.1 p.u.), exciter voltage step change (± 0.03 p.u.) and 3 phase short circuit fault on one transmission line, are also employed in the training. The sampling time is 30 ms.

After the off-line training is finished, the weights of the RNN controller and the RNN identifier will be further updated in the simulation studies described in the following sections.

4.4 System Identification

In the proposed RNN PSS, the neural network controller will calculate the control signal based on the results of the identifier neural network. Therefore, it is very important to properly track the system behaviour.

In this section, the proposed RNN identifier is studied in a single machine infinite bus power system shown in Fig. 4.2. The mathematical models and the parameters of the generator, AVR, exciter and governor are given in Appendix A. The response of the RNN identifier after training is compared to the output of the plant. The sampling period used in the studies is 30 ms.

4.4.1 Normal load condition

With the generator operating at $P_e = 0.7$ p.u. and $pf = 0.85$ lag, a 0.10 step increase at the input torque reference is applied at 1 second. The actual generator speed deviation and the RNN identifier output are shown in Fig. 4.3. It is clearly demonstrated that the RNN identifier can closely follow the response of the generator.

4.4.2 Light load condition

In this test, with the generator operating at $P_e = 0.3$ p.u., $pf = 0.85$ lag, a 0.15 p.u. step increase in the input torque reference is applied at 1 second. The response of the generator speed deviation and identifier output under the new operating condition is shown in Fig. 4.4.

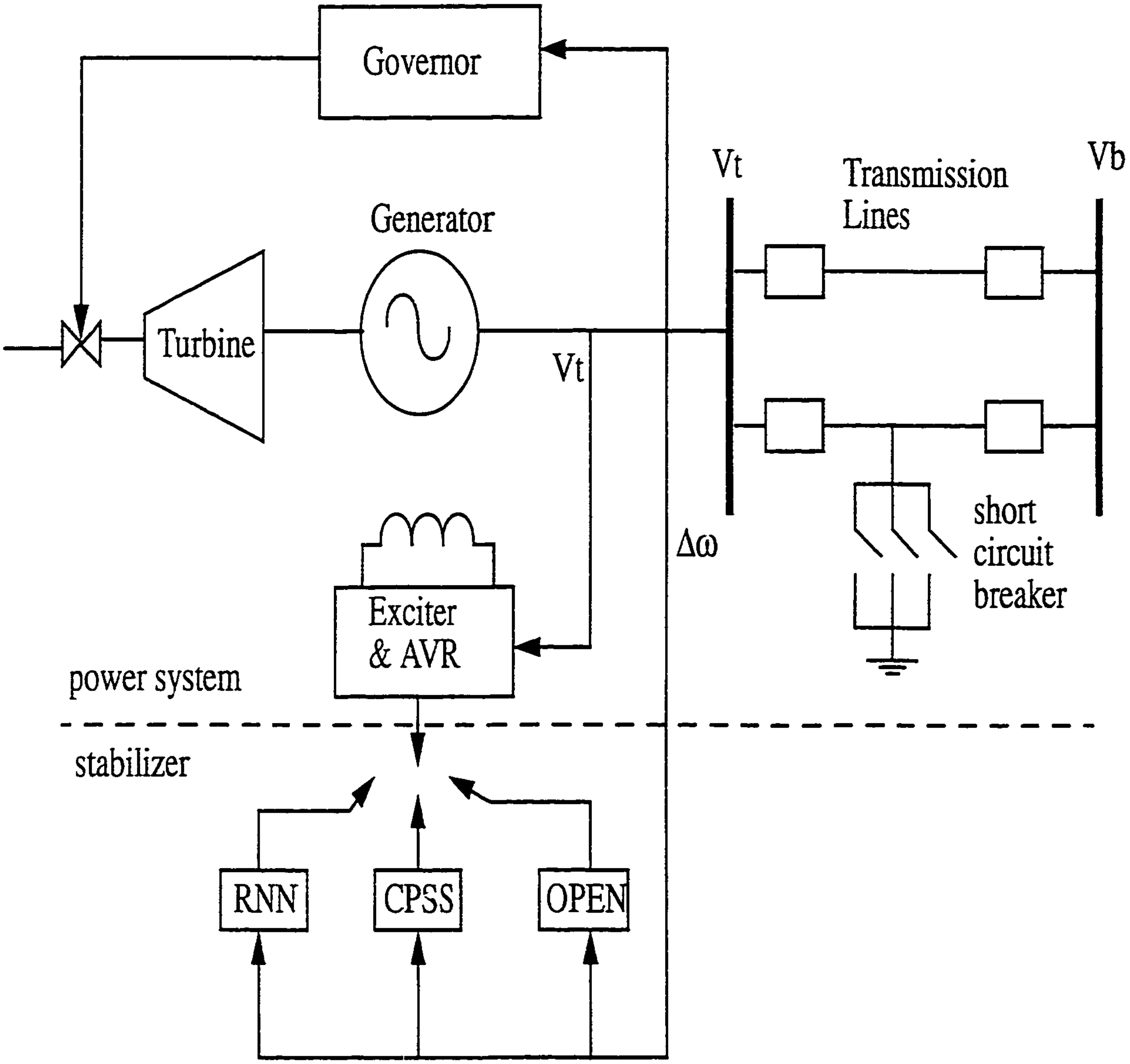


Figure 4.2. Structure of the power system

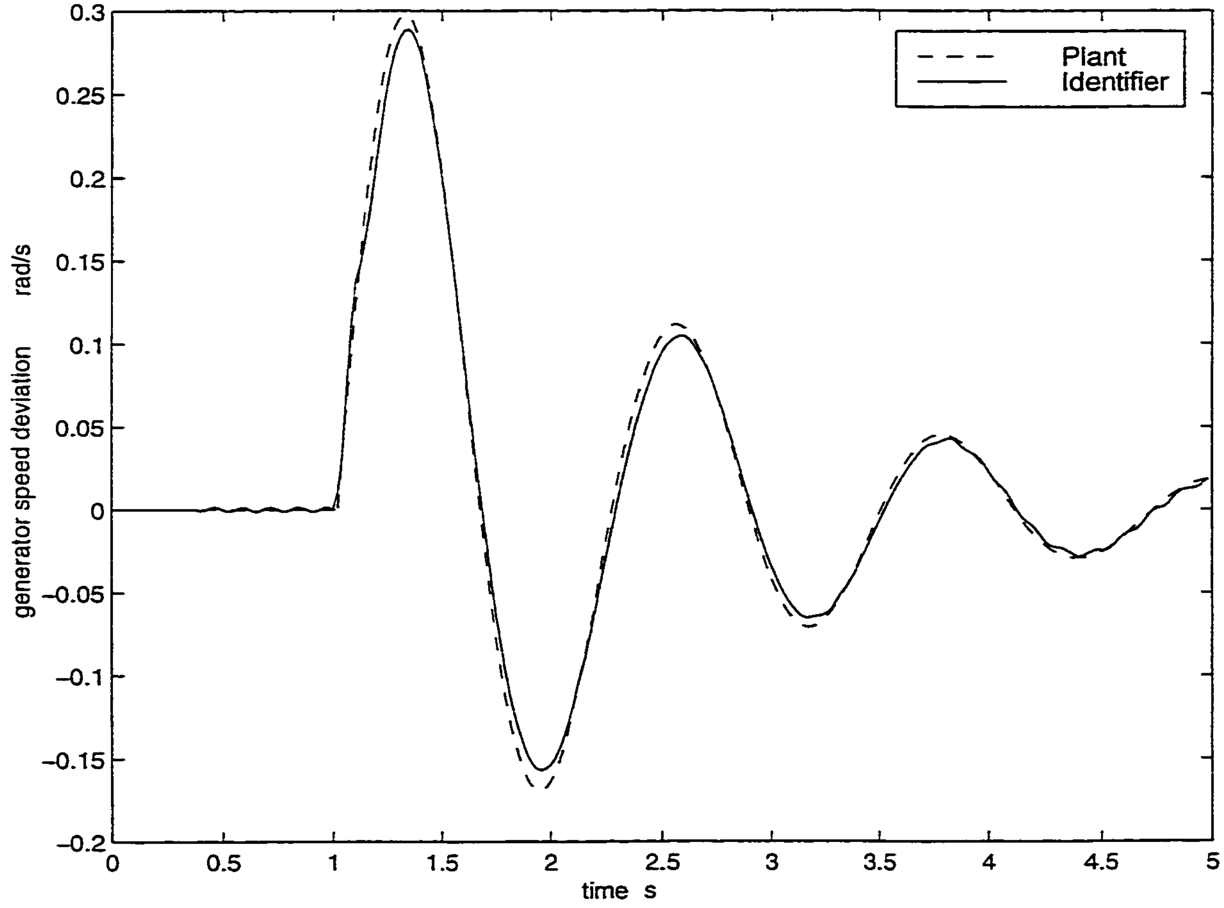


Figure 4.3. Identification result for a 0.10 pu step change disturbance in the input torque

4.4.3 Voltage Reference Change

With the generator operating at $P_e = 0.8$ p.u. and $pf = 0.85$ lag, a 0.03 step increase in the exciter reference voltage is applied at 1 second. The actual generator speed deviation and the RNN identifier output are shown in Fig. 4.5. The operating condition and the type of disturbance are different from previous studies. Again the RNN identifier can satisfactorily track the variation of the generator.

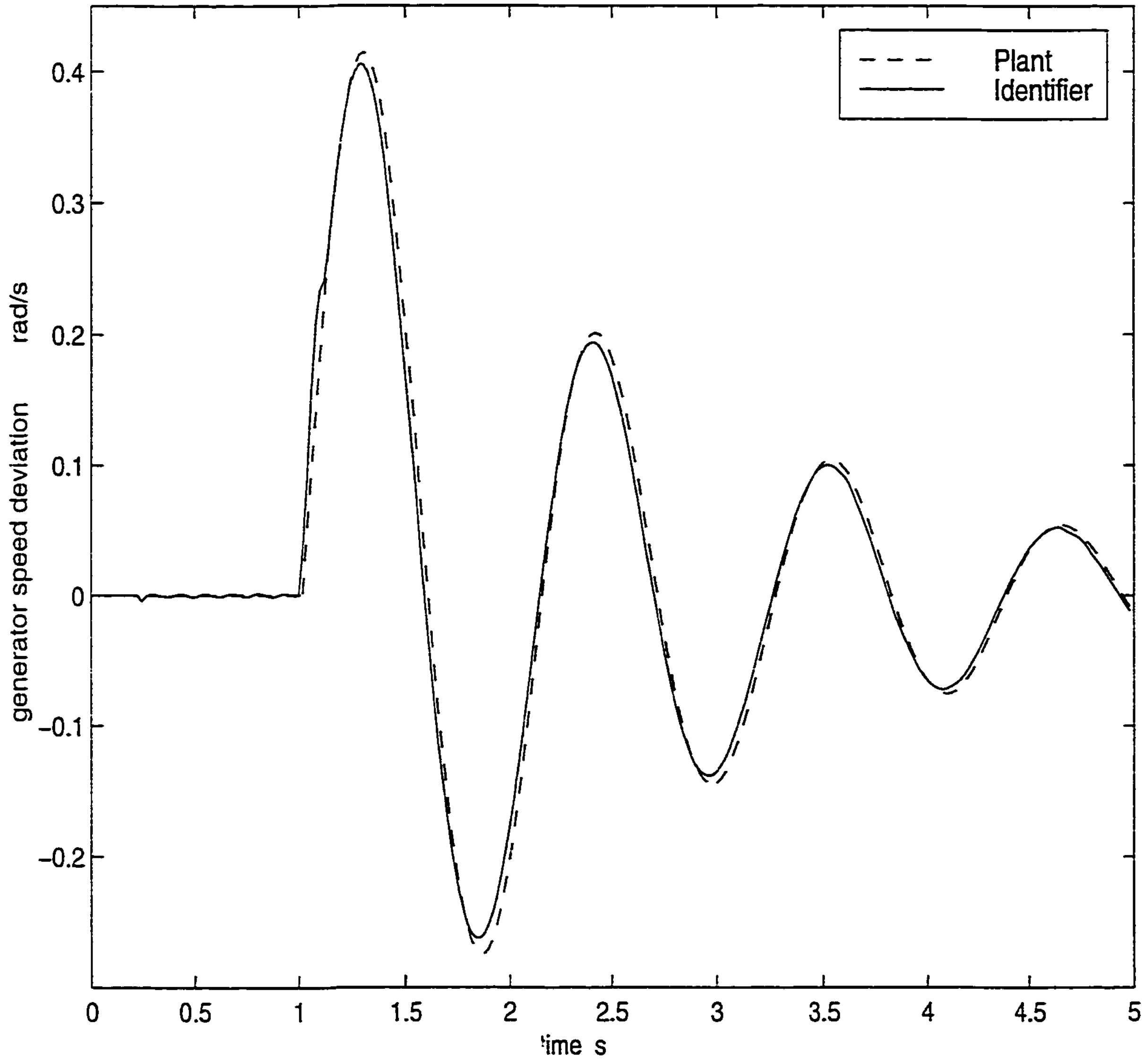


Figure 4.4. Response for 0.15 pu step change disturbance in input torque under light load

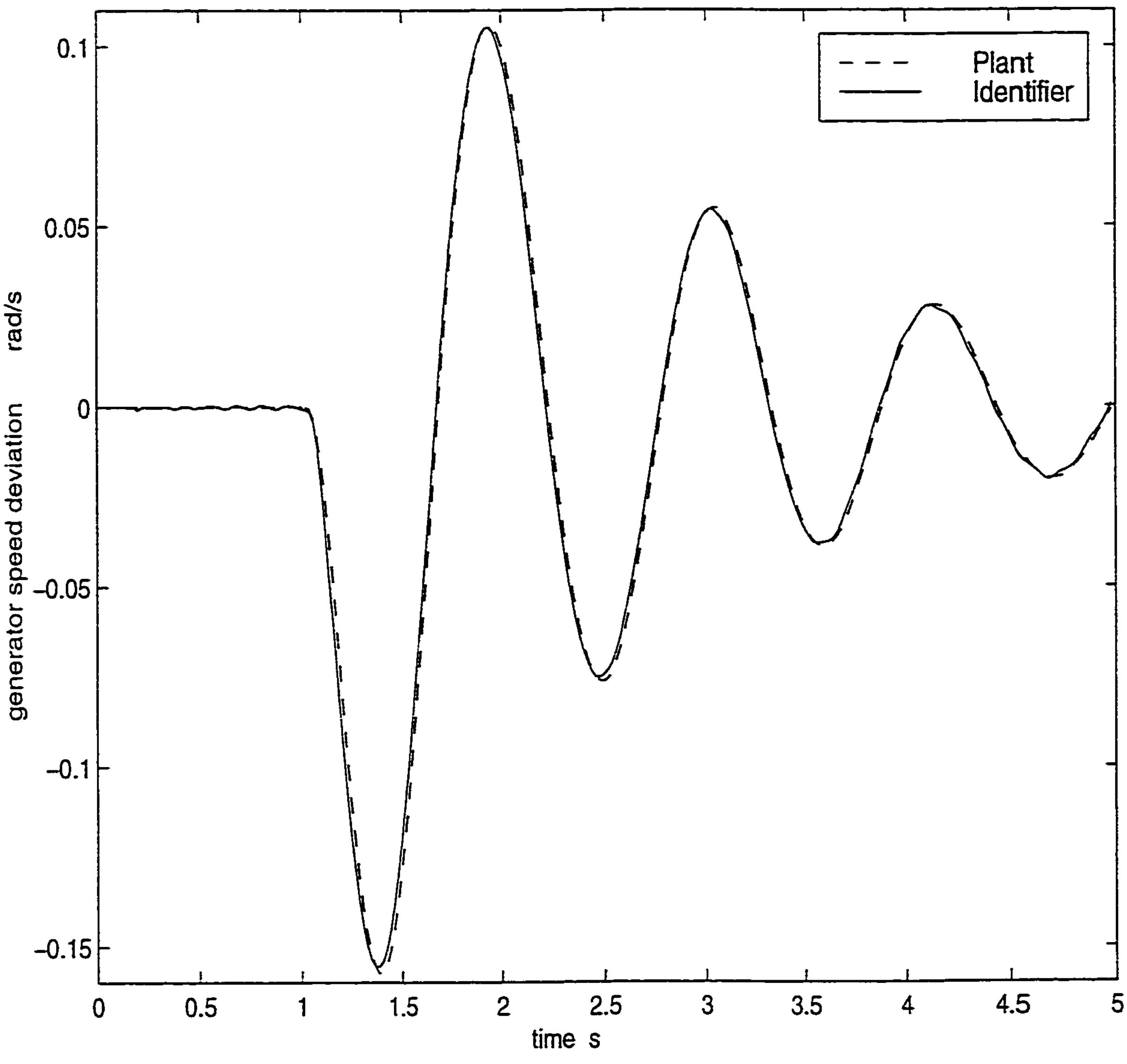


Figure 4.5. Identification result for 0.03 pu step change in exciter reference voltage

4.4.4 Leading power factor

In this test, with the generator operating at $P_e = 0.3$ p.u., $pf = 0.85$ lead, a 0.20 p.u. step increase at the input torque reference is applied at 1 second. The response of generator speed deviation and identifier output under the leading power factor operating condition is shown in Fig. 4.6. The RNN identifier can provide good tracking performance.

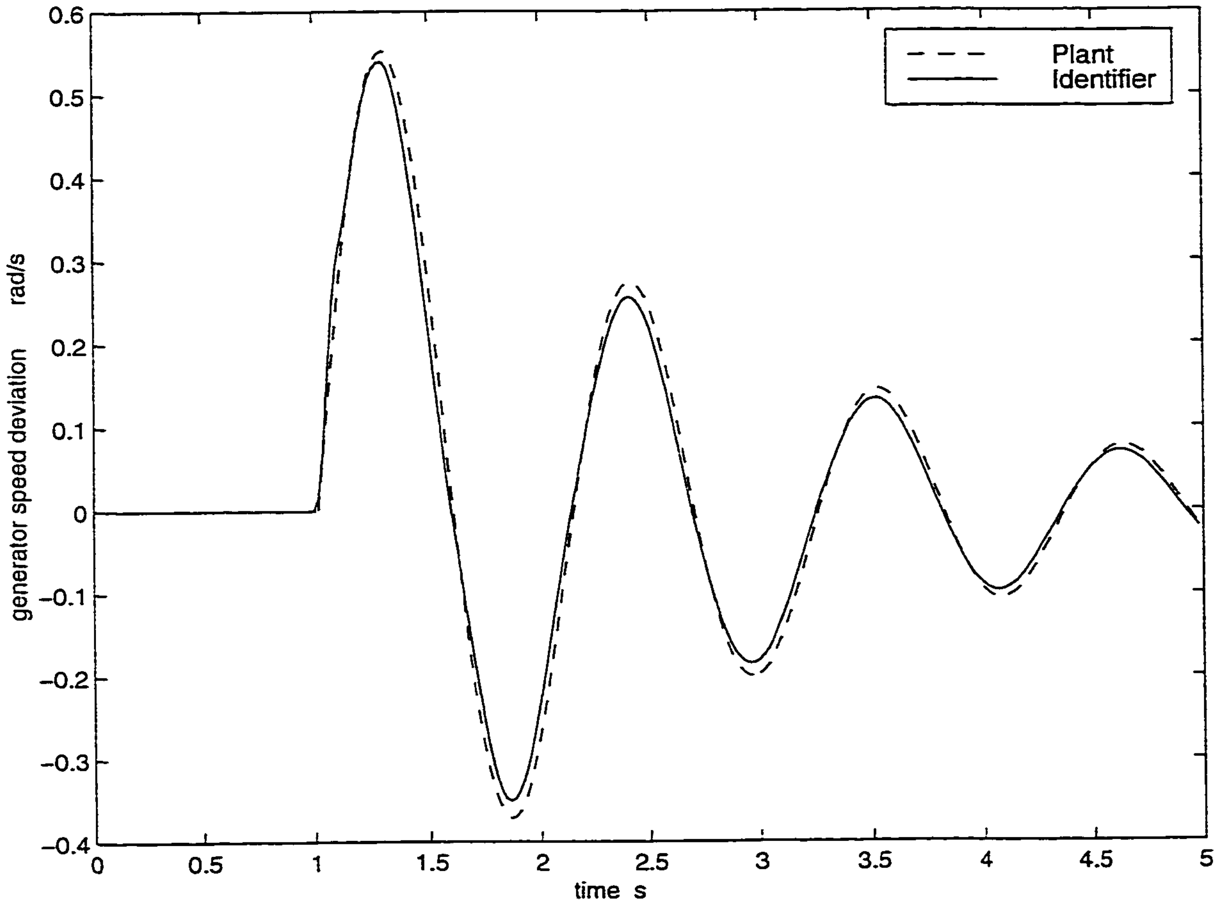


Figure 4.6. Identification result for 0.20 pu step change in input torque

4.5 Control Simulation Studies

Performance of the proposed RNN PSS is next investigated on a synchronous generator connected to a constant voltage bus through two parallel transmission lines. The structure of the power system is shown in Fig. 4.2 . The response of the system under different control schemes (i.e. RNN PSS, CPSS and no stabilizer (Open)) is compared. The mathematical model and parameters of the conventional PSS are listed in Appendix A. The sampling period for the RNN PSS is 30 ms.

4.5.1 CPSS parameter tuning

With the generator operating at a power of 0.7 *p.u.*, 0.85 power factor lag, a 0.06 *p.u.* step increase in input torque reference is applied at 1 second, and removed at 5 second, and the system returns to its original operating condition.

The CPSS is carefully tuned under the above conditions to yield the best performance. the parameter of the CPSS are then kept the same for all the rest of simulation studies. Results of the study with RNN PSS and CPSS for a 0.06 *p.u.* step change in input torque are shown in Fig. 4.7. It can be seen both RNN PSS and CPSS can effectively damp the low frequency oscillations .

4.5.2 Light Load Test

A light load test with the initial condition of 0.3 p.u. power and 0.85 pf lag is conducted. The disturbances are a 0.1 p.u. step increase in torque reference applied at 1.02s and then removed after 5s. The disturbance is large enough to cause the system to operate in a nonlinear region. Power angle response without PSS, with CPSS and with RNN PSS is shown in Fig. 4.8 and voltage response with RNN PSS and CPSS is shown in Fig. 4.9. Since RNN PSS can adapt its response to the change of the environment, it provides better damping effect than the fixed parameter CPSS.

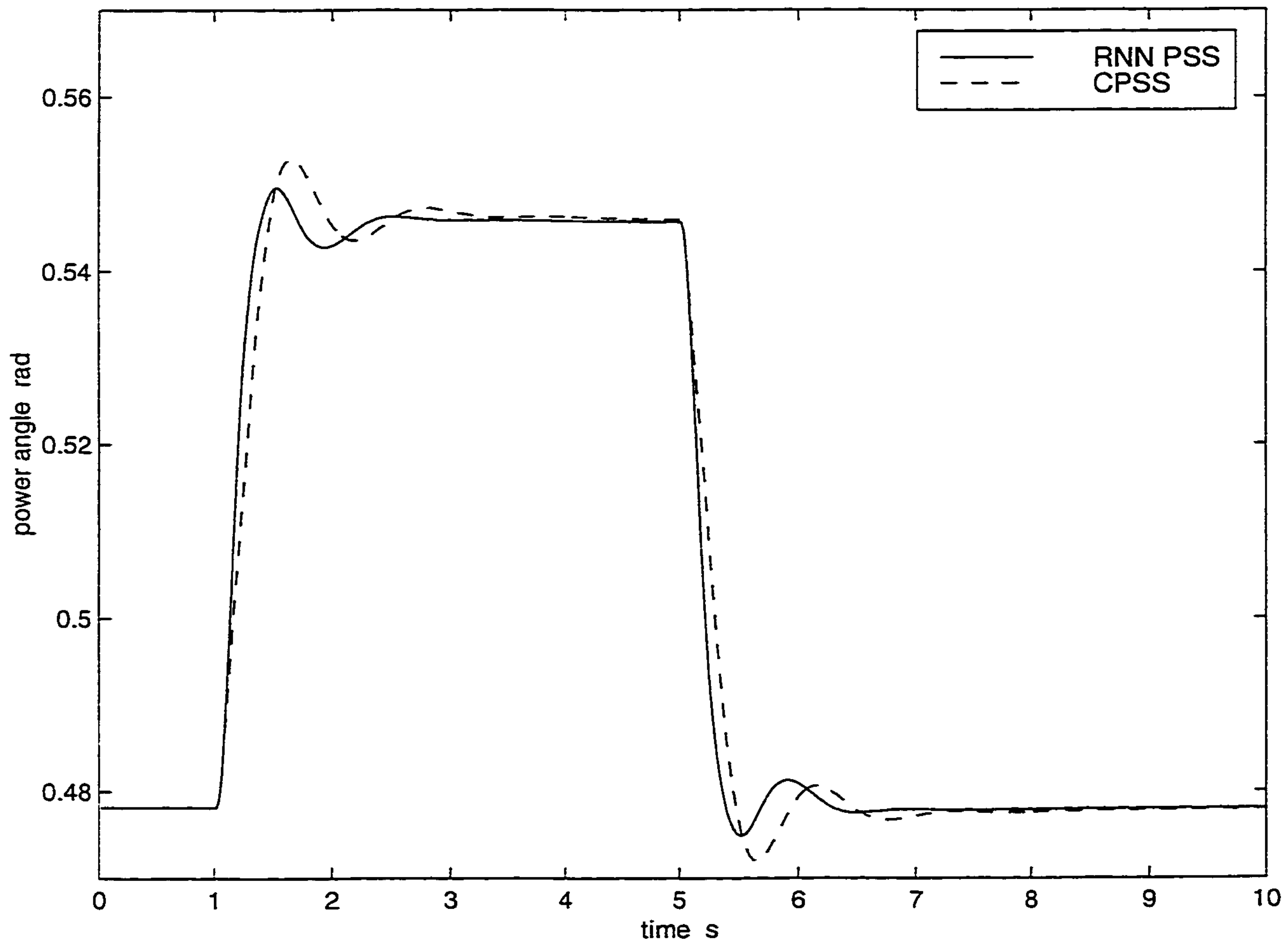


Figure 4.7. Response to a 0.06p.u. step increase in torque and return to initial condition. $P_e = 0.70pu$, p.f.=0.85 *lag*

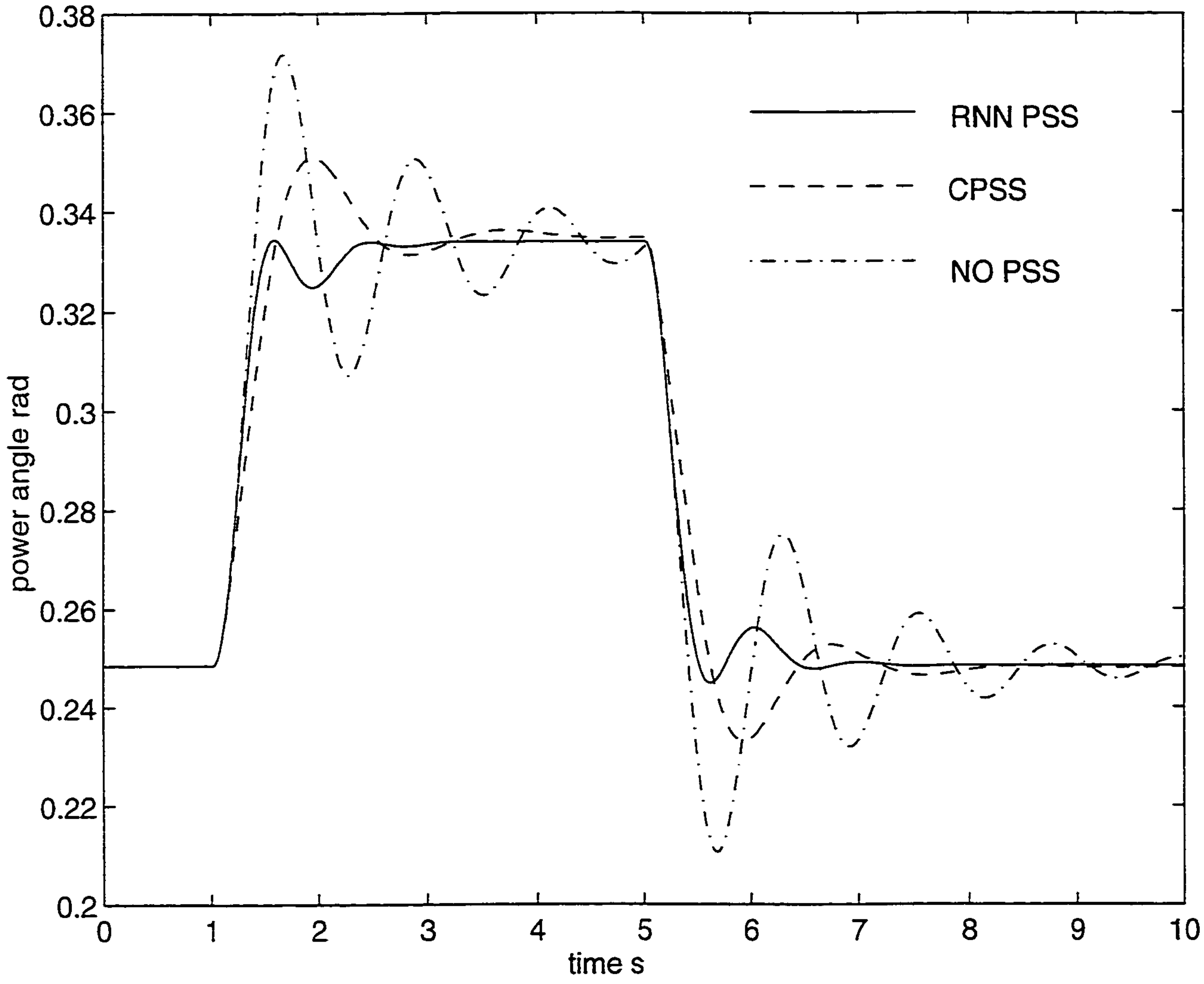


Figure 4.8. Response to a 0.10p.u. step increase in torque and return to initial condition. $P_e = 0.30pu$, p.f.=0.85 *lag*

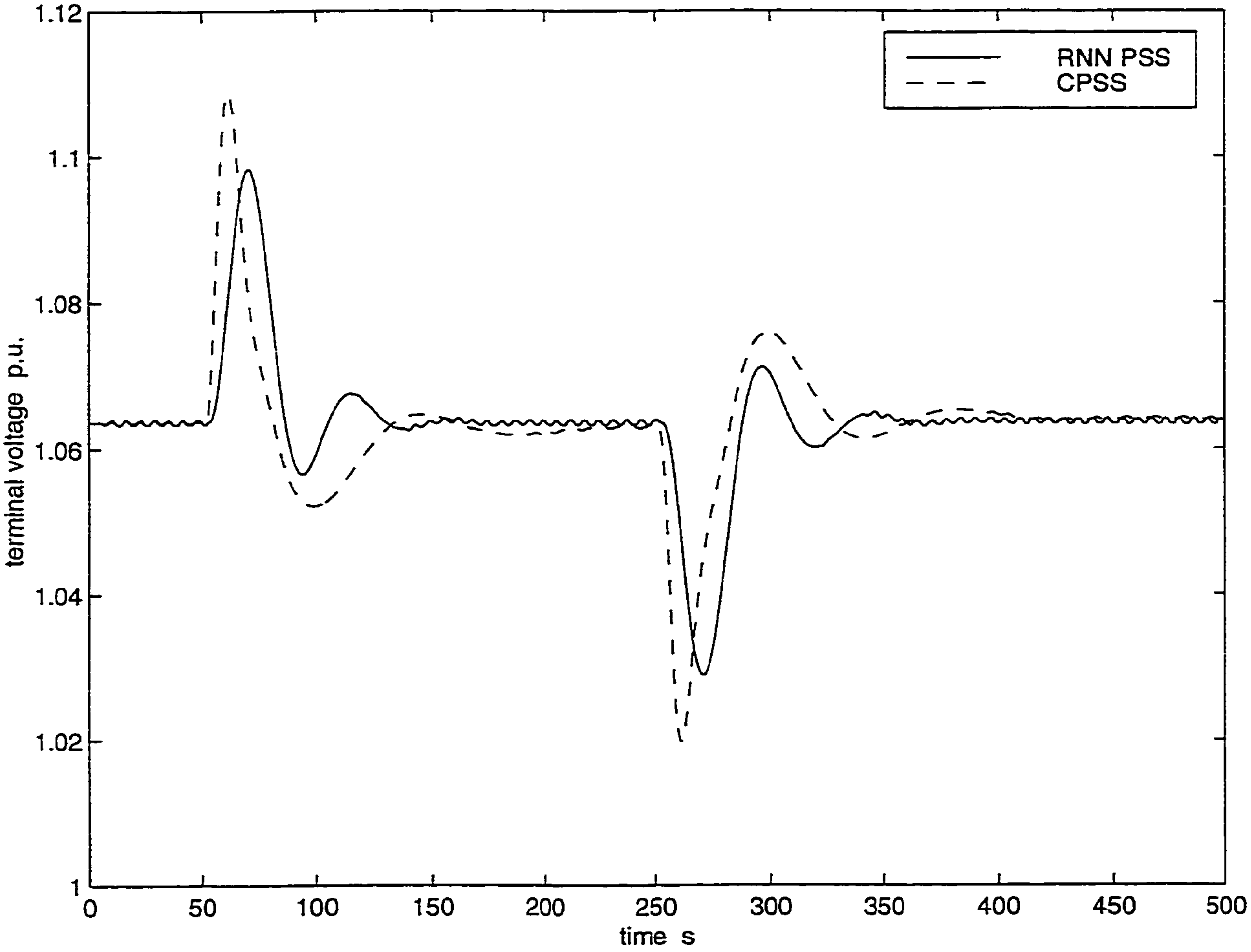


Figure 4.9. Voltage response to a 0.10p.u. step increase in torque and return to initial condition. $P_e = 0.30pu$, p.f.=0.85 *lag*

4.5.3 Voltage Reference Change Test

In this test, the initial operating condition is 0.70 p.u. power and 0.85 pf lag. A 0.03 p.u. step increase in exciter reference voltage is applied at t = 1.02s and removed after 5s. The power angle response is shown in Fig. 4.10. It is clear that ANN PSS has a better damping effect than CPSS.

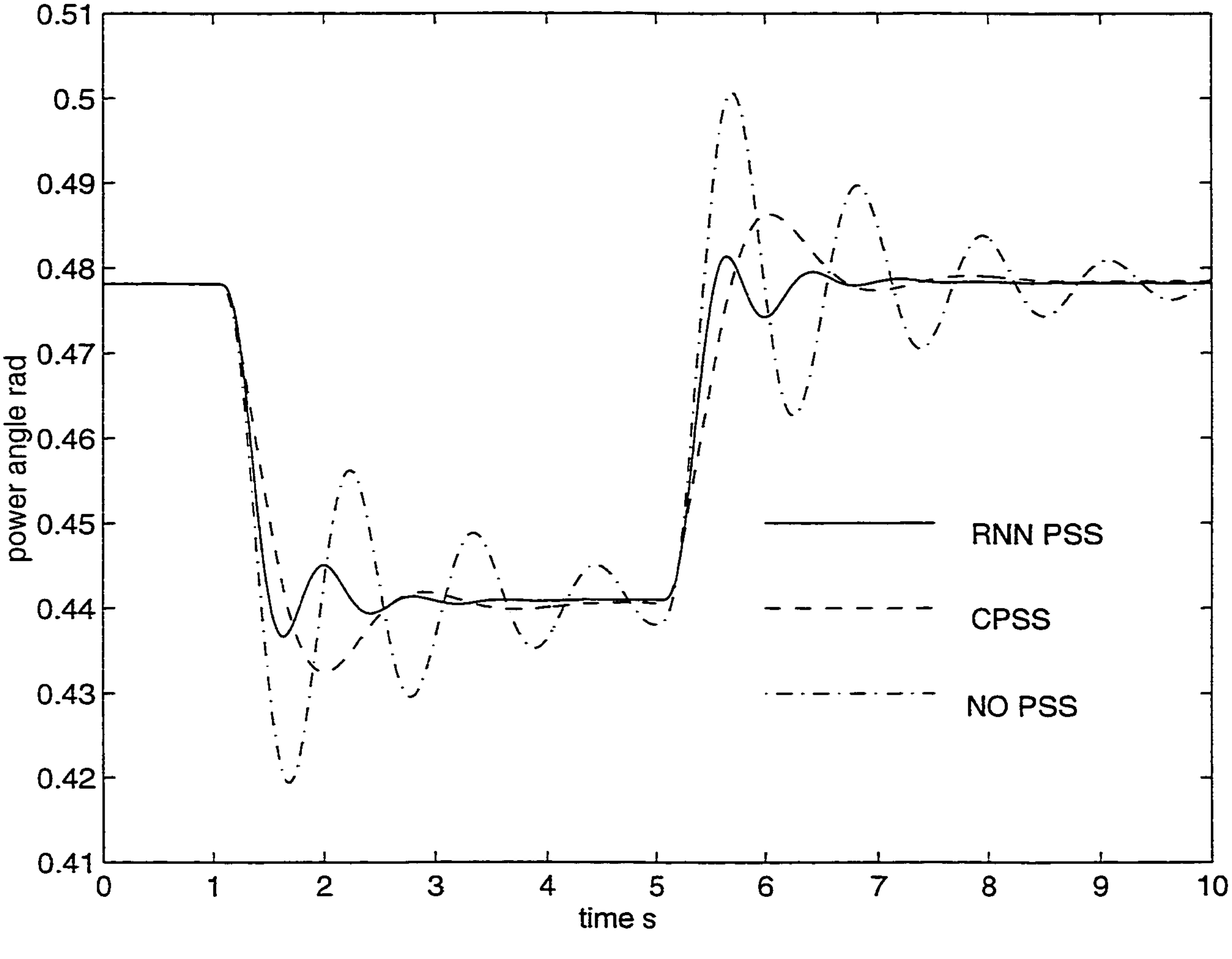


Figure 4.10. Response to a 0.03p.u. step increase in exciter reference voltage and return to initial condition. $P_e = 0.7pu$, pf= 0.85 *lag*

4.5.4 Leading Power Factor Test

In order to absorb the capacitive charging current in a high voltage power system, a generator may be required to operate at leading power factor. Under this circumstance, the stability margin of the generator is reduced compared to the normal operating condition. Therefore, it is difficult for the controller to keep the generator stable when disturbance happens.

The behavior of the ANN PSS is also studied under leading power factor condition with 0.25 p.u. power and 0.90 pf lead. A 0.10 p.u. change in input torque reference is applied at 1.02s and removed after 5s. The results given in Fig. 4.11 show that with the RNN PSS, the system goes to the new operating point quickly.

4.5.5 Transient Test

With the system operating at a power of 0.90 p.u. and 0.85 pf lag, a transient test was conducted to study the performance of the proposed ANN PSS. In the test, a three phase to ground short circuit fault was applied at the middle of one transmission line, the faulted line was cleared 50ms later, and then after 3.93s successfully reclosed. The power angle response and controller output are shown in Figs. 4.12 and 4.13, respectively. The terminal voltage response with RNN PSS and CPSS is shown in Fig. 4.14. The output of RNN PSS is smoother than CPSS, and the RNN PSS suppressed the oscillations more efficiently than the CPSS. The system with the RNN PSS goes to the new stable state quickly. This is very helpful in the enhancement of the disturbance tolerance ability of the system.

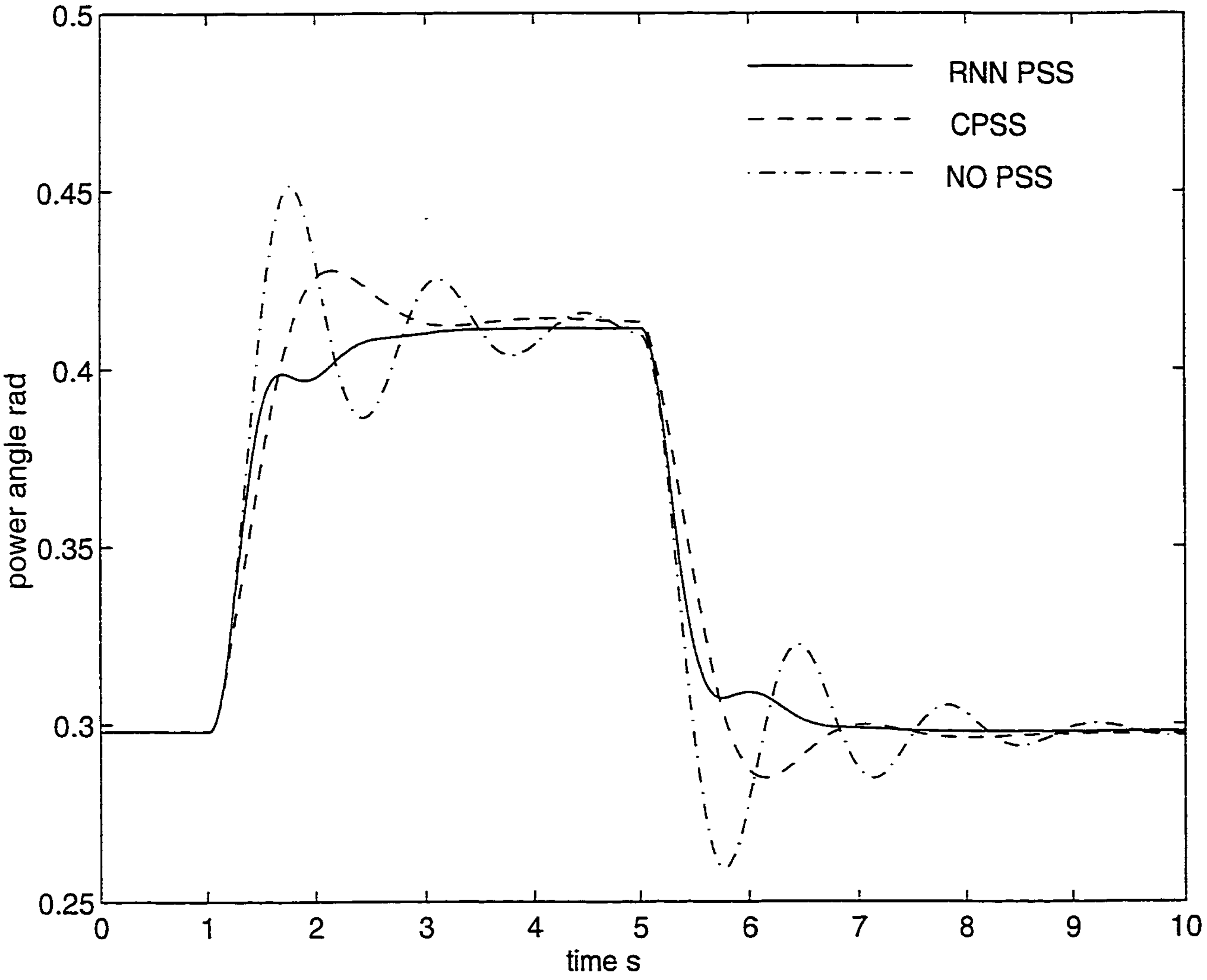


Figure 4.11. Response to a 0.10 p.u. step increase in torque and return to initial condition. $P_e = 0.25 p.u.$, pf=0.90 *lead*

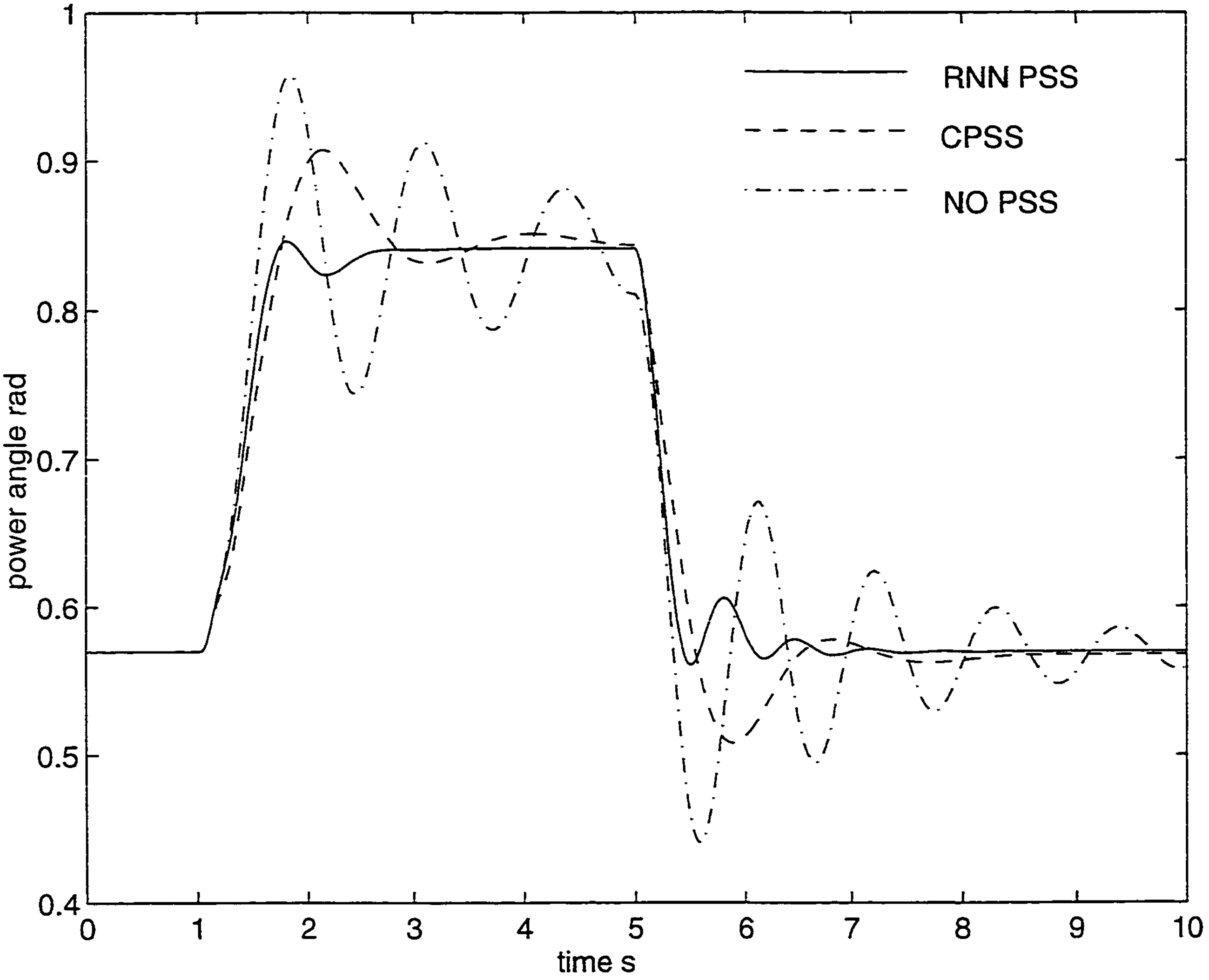


Figure 4.12. Response to a 3 phase short circuit at the middle of one transmission line, disconnected after 0.05s and successful reclosure 3.93s later. $P_e = 0.90pu$, p.f.=0.85 *lag*

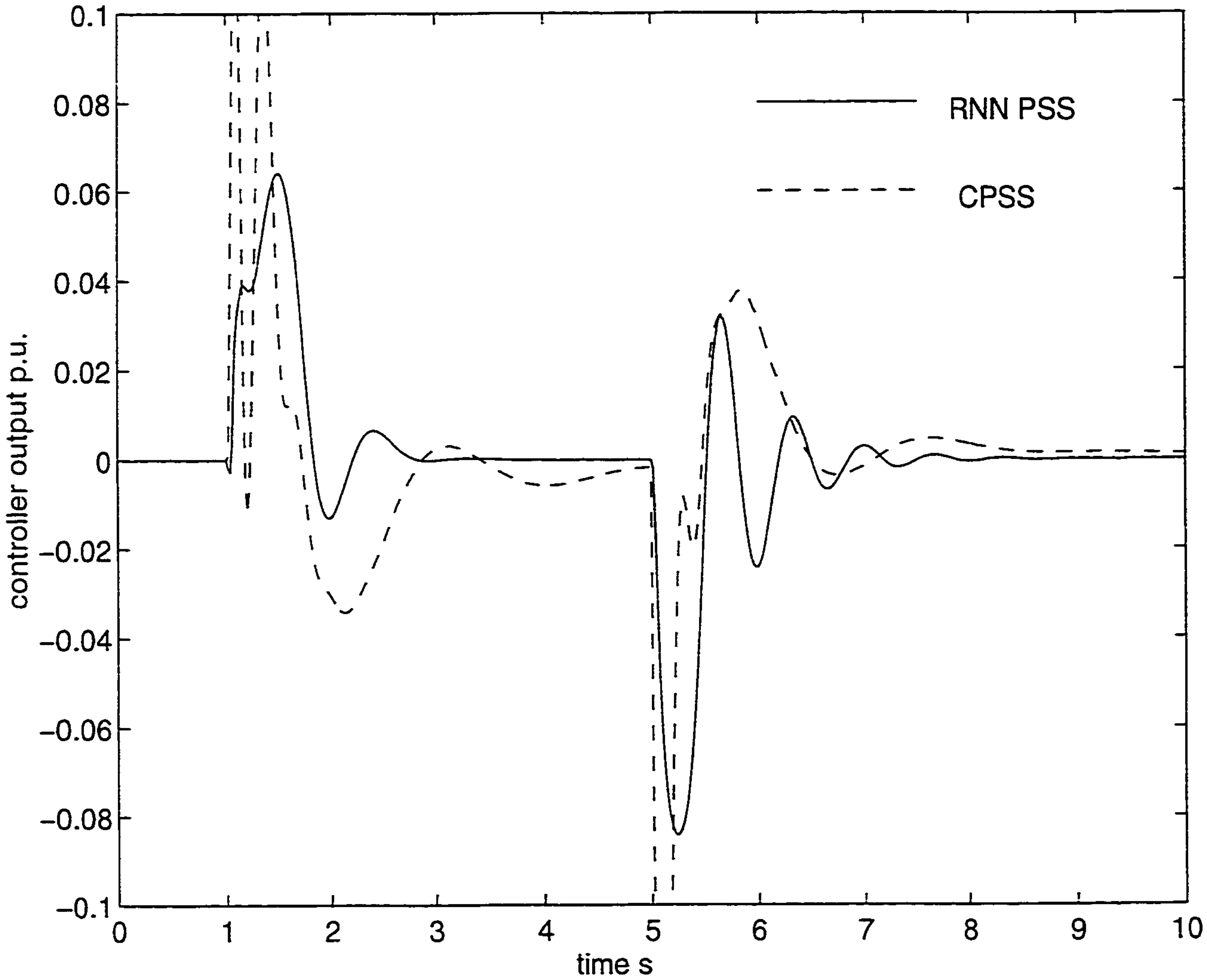


Figure 4.13. Controller output in response to a 3 phase short circuit at the middle of one transmission line, disconnected after 0.05s and successful reclosure 3.93s later. $P_e = 0.90pu$, p.f.=0.85 *lag*

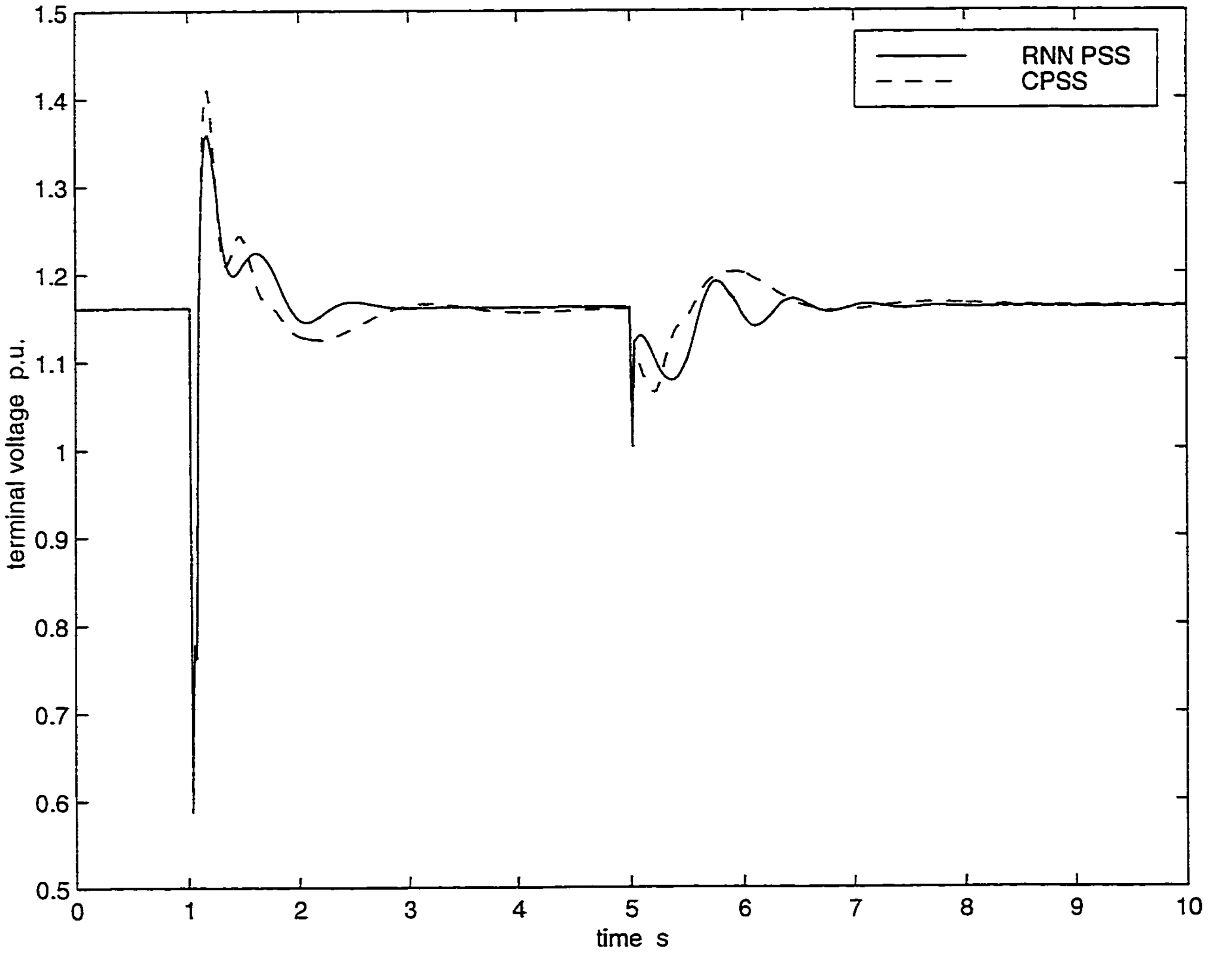


Figure 4.14. Terminal voltage in response to a 3 phase short circuit at the middle of one transmission line, disconnected after 0.05s and successful reclosure 3.93s later. $P_e = 0.90pu$, p.f.=0.85 *lag*

Table 4.1. Dynamic Stability Limit

	Open (without PSS)	with CPSS	with RNN PSS
Max. Power(p.u.)	1.787	3.050	3.227
Max. Rotor Angle(rad)	1.108	2.004	2.096

4.5.6 Different Sampling Period

With the system operating at a power of 0.3 p.u. and 0.9 pf lead, a 0.20 p.u. change in input torque reference is applied at 2s. The RNN PSS is operated under different sampling periods: 25ms, 30ms and 35ms. The results given in Fig. 4.15 show that under different sampling periods, RNN PSS still can provide satisfactory result.

4.5.7 Dynamic Stability Margin Test

The ANN PSS not only improves the generator's transient behavior, but also enhances its dynamic stability properties. In this section, a test was conducted to demonstrate the effect of the proposed RNN PSS on the dynamic stability margin.

With the system initially operating at 0.95 p.u. power and 0.90 pf lag, the input torque reference was increased gradually from the initial value. The dynamic stability margin is described as the maximum power output at which the system losses synchronism. The results for the system without stabilizer (Open), with CPSS, and with RNN PSS are listed in Table 4.1. The results show that the dynamic stability margin of the generator is increased with the RNN PSS.

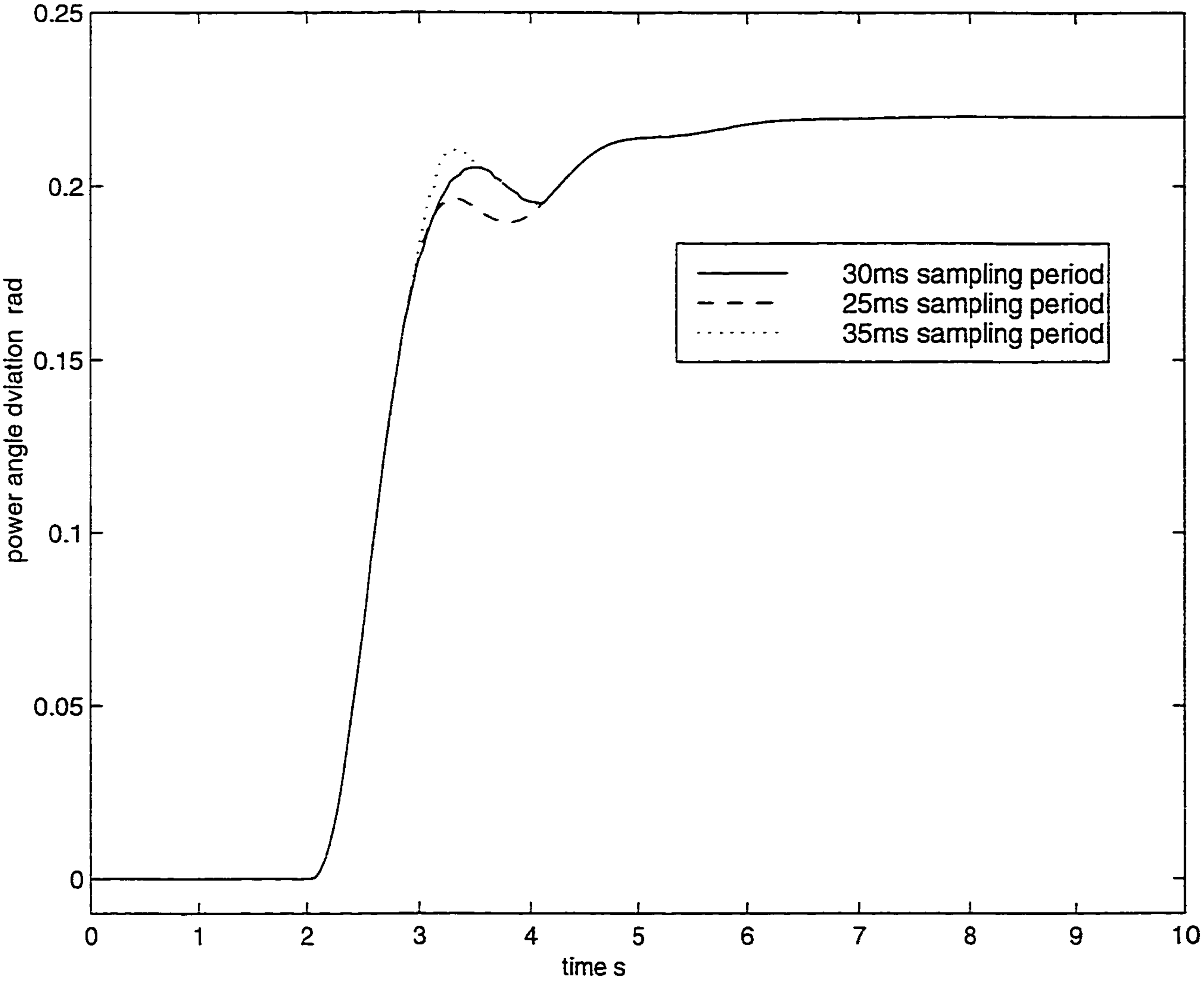


Figure 4.15. Responses to a 0.20 p.u. change in input torque reference with different sampling periods, $P_e = 0.90pu$, p.f.=0.85 *lag*, with RNN PSS

4.5.8 Weight Variation

For the proposed RNN PSS, the weights of both the RNN identifier and the RNN controller are updated on-line. The variation of the sum of the RNN identifier weights squared and the variation of the sum of the RNN controller weights squared is shown in Figs. 4.16 and 4.17, respectively. The test case is the same as described in Section 4.5.5.

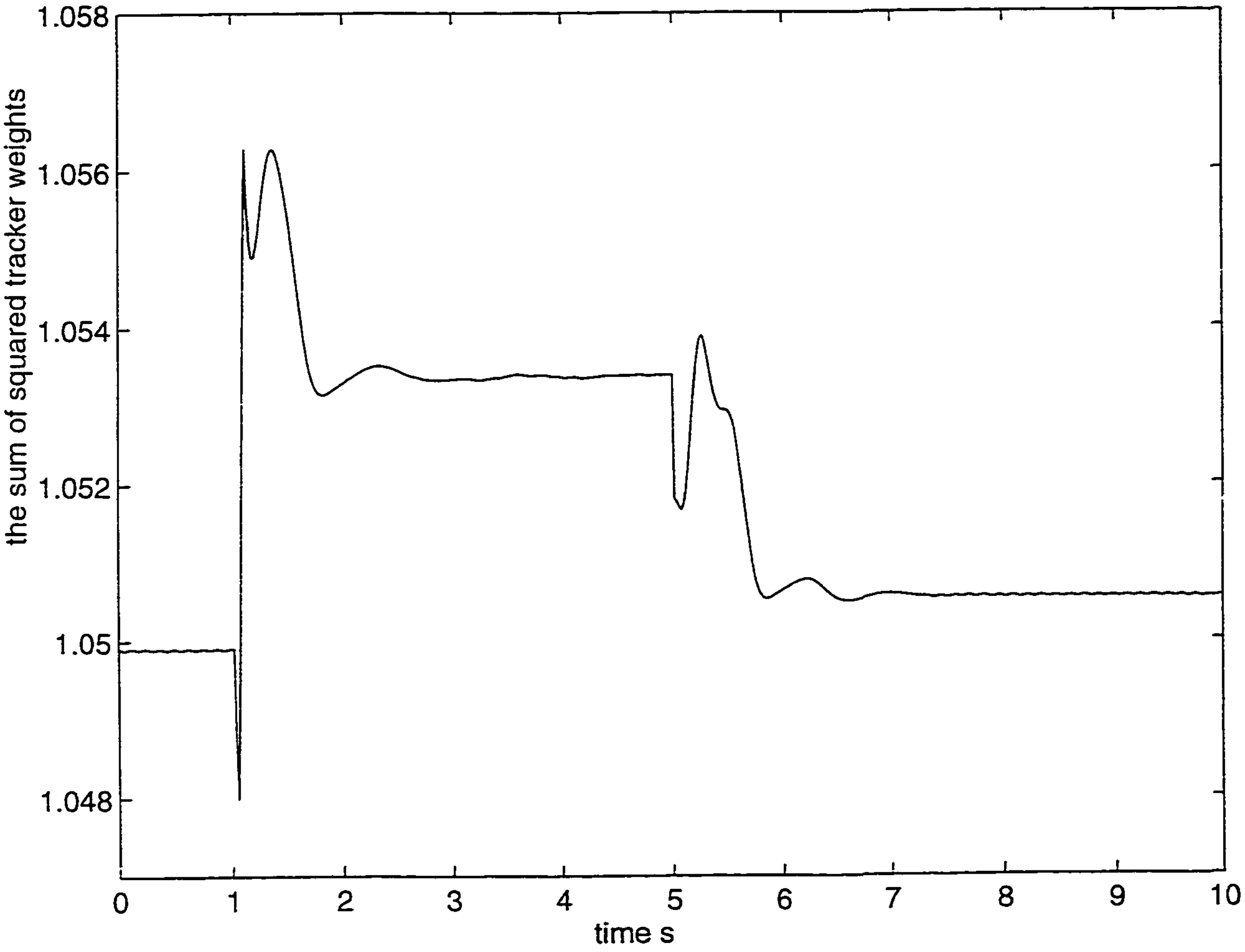


Figure 4.16. Variation of identifier weights in response to a 3 phase short circuit at the middle of one transmission line. $P_e = 0.90pu$, p.f.=0.85 *lag*

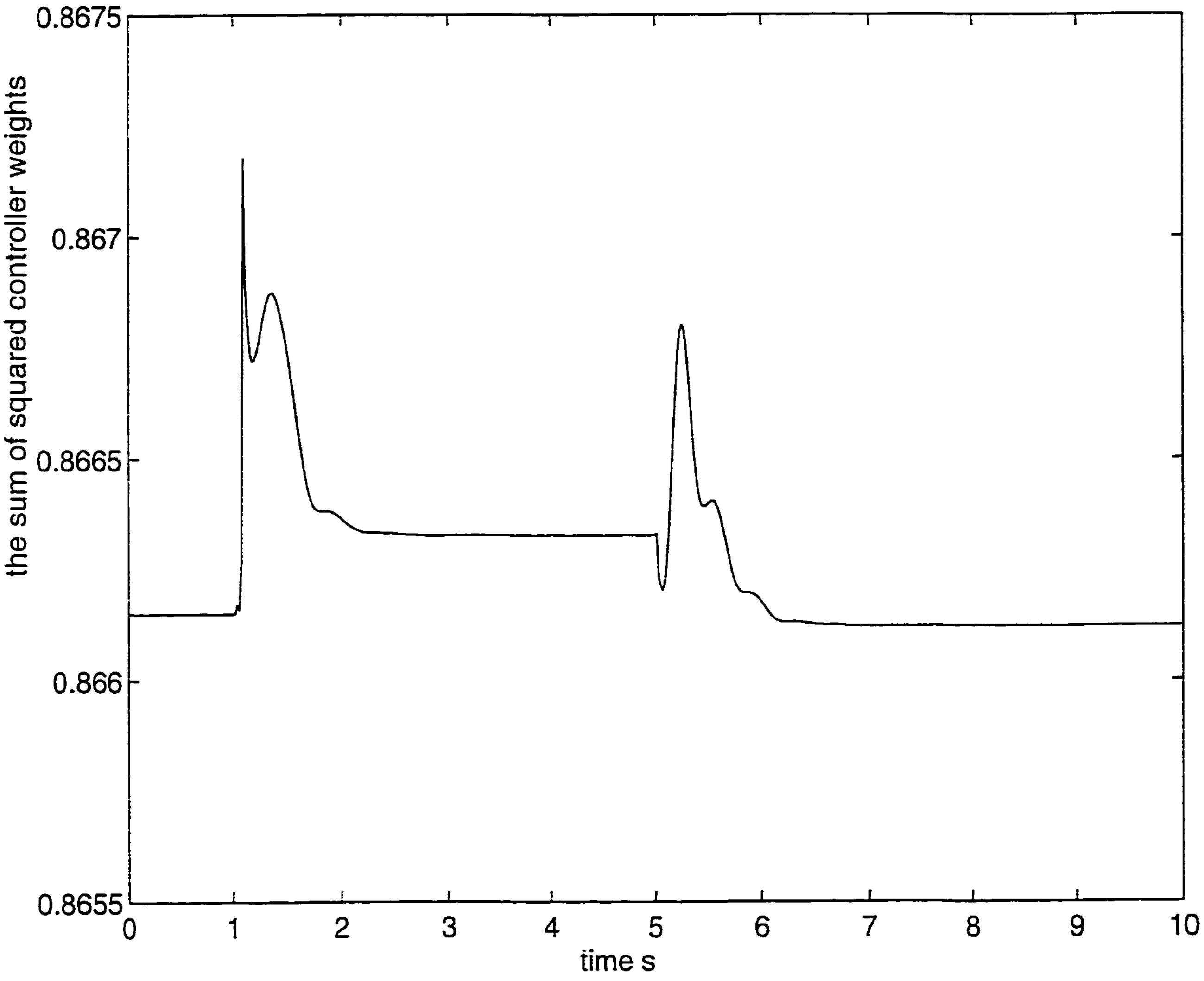


Figure 4.17. Variation of controller weights in response to a 3 phase short circuit at the middle of one transmission line. $P_e = 0.90pu$, p.f.=0.85 *lag*

4.6 Summary

In this chapter, the performance of an adaptive PSS based on recurrent neural networks is investigated in a single machine-infinite bus power system.

The proposed stabilizer has the following advantages:

- The RNN PSS is not designed for a fixed operating point. The ANN PSS is trained on a wide range of generator operating conditions. With the learning ability of neural networks, the ANN PSS can performance well for a wide range of operating condition.
- During the design procedure, the generator model is not linearized.
- The parameters (i.e., the weights of the neural networks) of the controller are updated on line. Therefore, the controller can track changes in operating conditions.
- In this control architecture , no reference model is needed. Since for the neural network tracker, the inputs are the outputs of neural network controller and plant, its weights are tuned based on the derivative of the tracker output and plant output. For the neural network controller, the inputs are the outputs of the plant. Its weights are tuned based on the derivative of plant output and desired output.

The simulation results show that compared to the conventional PSS, the proposed PSS has a better damping ability for different operating conditions and disturbances. It also improves the dynamic stability properties of the generator.

CHAPTER 5

RNN PSS SIMULATION STUDIES IN MULTIMACHINE SYSTEM

5.1 Introduction

Today, the large-scale power system interconnections make electric energy generation and transmission more economical and reliable. However, at the same time, the multiple interconnections also make the power system much more vulnerable to instability. One of the instability problems is the multi-mode oscillations at very low frequencies. Once started, they could continue for a while and then disappear, or continue to grow, causing interconnected system separation.

The simulation studies in Chapter 4 demonstrate that properly trained RNN PSS can provide an effective damping of the power system [97]. These studies are conducted in the single-machine infinite-bus environment. In this environment, there are no multi-mode oscillations. To verify the damping ability of the new PSS, the performance of the new PSS should be investigated in a multimachine power system. [98] [99] [100] [101] [102] [103]

The effectiveness of the RNN PSS to damp multi-mode oscillations in a multimachine power system is investigated in this chapter. A five machine power system is used in these studies. Speed deviation $\Delta\omega$ and accelerating power ΔP_e are selected as input signals to the RNN PSS. Simulation results of the RNN PSS are compared with those of the CPSS, and the coordination ability of the RNN PSS with other RNN PSSs and CPSSs is also investigated.

5.2 Multimode oscillations in Multimachine System

In a multimachine power system, there are three major modes of oscillations:

Local Mode this refers to oscillations occuring 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. 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 voltage bus. Local mode oscillations are in the range of 0.8 to 2 *Hz*.

Inter-Machine Mode 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 inertias of the closely coupled generator groups and are in the range of 0.3 to 1.0 *Hz*

Inter-Area Mode 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 definitions.

5.3 Multimachine System Model

5.3.1 Configuration of a Five Machine Power System

A five-machine power system without infinite bus, shown in Fig. 5.1 , is used for simulation studies in this chapter. In this system, Generators #1, #2 and #4 have much larger generating capacities than Generators #3 and #5. All five generators

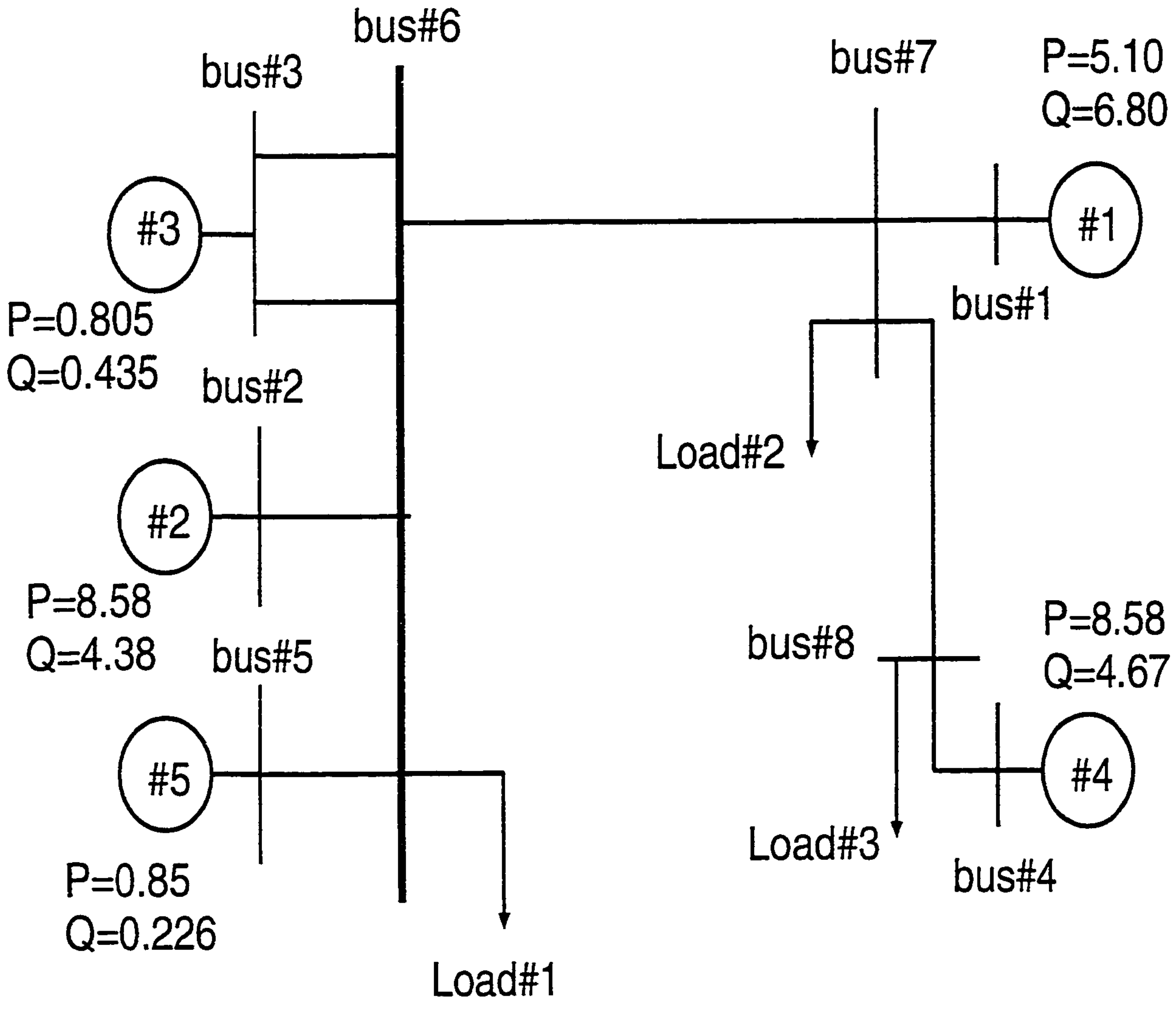


Figure 5.1. Schematic diagram of a five machine power system.

are equipped with governors, AVRs and exciters. The whole system can be viewed as a combination of two areas connected through a tie transimission line between bus #6 and bus #7. Generators #1 and #4 form one area, and generators #2, #3 and #5 form another one. Parameters of all generators, transmission lines, loads and operating conditions are given in Appendix B. Under normal operating condition, each area serves its local load and is almost fully loaded with a small load flow over the tie line.

5.3.2 Multi-Mode Oscillation in the System

When a disturbance happens in this system, the multi-mode oscillations will arise because of the different inertias of the generators and the topology of the system.

A 0.1 *p.u.* step decrease in the mechanical input torque reference of generator #3 is applied at 1 *s*, and returns to the original level at 10 *s*. Without any PSS installed in the system, the following obeservation can be made from the system responses to this disturbance in Fig. 5.2:

- The speed difference between generator #2 and generator #3 exhibits the local mode oscillation at a frequency of about 1.3 *Hz*.
- The speed difference between generator #1 and generator #2 shows mainly the inter-area mode oscillation with a frequency about 0.65 *Hz*.
- The speed difference between generator #1 and generator #3 displays the combination of local mode and inter-area mode.

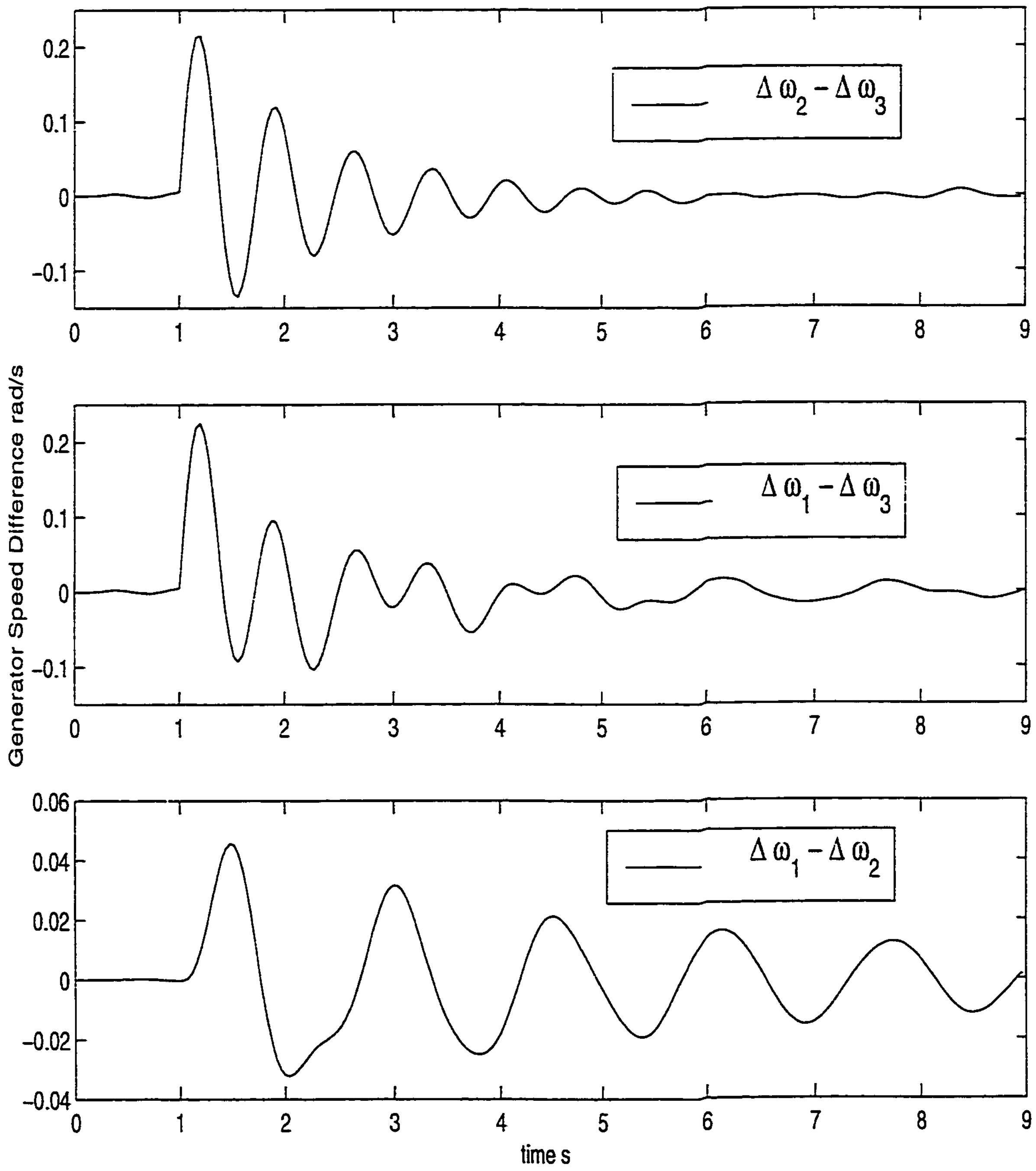


Figure 5.2. Multi-mode oscillations of the five-machine power system

5.4 RNN PSS Design in Multimachine System

5.4.1 RNN PSS Structure

The structure of the RNN PSS for this study is shown in Fig. 5.3. For the neural-identifier, the input vector is:

$$[\Delta\omega(k), \ldots, \Delta\omega(k-m), \Delta Pe(k), \ldots, \Delta Pe(k-n), U_{annc}(k), \ldots, U_{annc}(k-p)] \quad (5.1)$$

where $\Delta\omega(k)$, $\Delta Pe(k)$ and $U_{annc}(k)$ are the generator speed deviation, accelerating power and the PSS signal, respectively, at the k-th time step. The output is $\Delta\hat{P}e(k+1)$, the predicted accelerating power, at time step (k+1). Then the performance index of the RNN identifier is :

$$J_i(k) = \frac{1}{2}[\Delta\hat{P}e(k) - \Delta Pe(k)]^2 \quad (5.2)$$

The weights in the RNN identifier are updated as below:

$$\mathbf{W}_i(k) = \mathbf{W}_i(k-1) - \eta_i \nabla_{W_i} J_i(k) \quad (5.3)$$

where $\mathbf{W}_i(k)$ is the matrix of neural identifier weights at instant k, $\nabla_{W_i} J_i(k)$ is the instantaneous gradient, η_i is the learning rate for the neural-identifier.

For the neural-controller, the input vector is,

$$[\Delta\omega(k), \ldots, \Delta\omega(k-m), \Delta Pe(k), \ldots, \Delta Pe(k-n)] \quad (5.4)$$

The output is $U_{annc}(k)$, the PSS control signal at time step k. The performance index of the neural controller is:

$$J_c(k) = \frac{1}{2}[\Delta\hat{P}e(k) - \Delta Pe_d(k)]^2 \quad (5.5)$$

where $\Delta Pe_d(k)$ is the desired accelerating power at time step k. In this study, it is set to be zero. The weights in the neural-controller are updated as below:

$$\mathbf{W}_c(k) = \mathbf{W}_c(k-1) - \eta_c \nabla_{W_c} J_c(k) \quad (5.6)$$

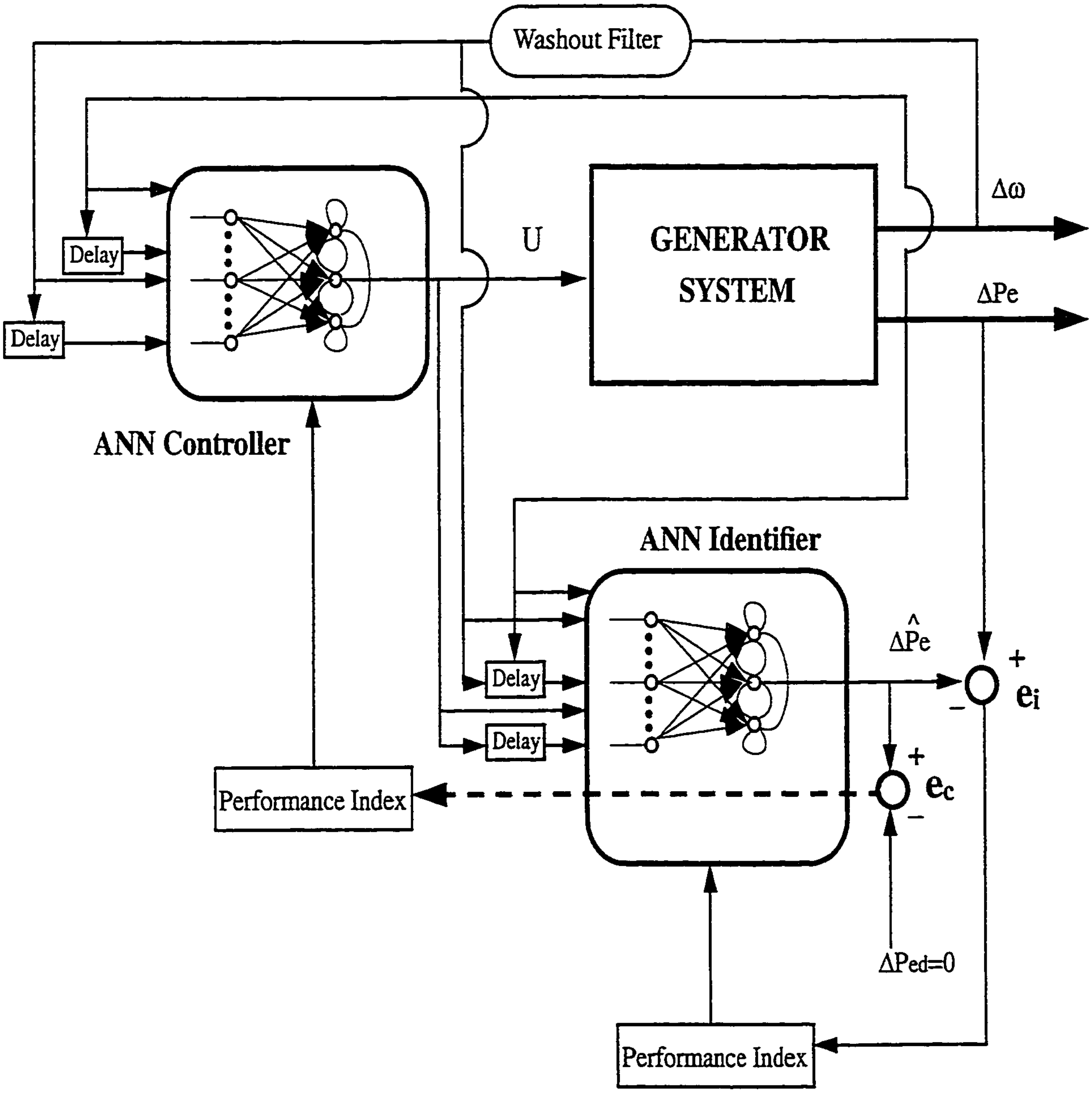


Figure 5.3. Schematic diagram of Neural Network Controller.

where $\mathbf{W}_c(k)$ is the matrix of neural controller weights at instant k, $\nabla_{W_c} J_c(k)$ is the instantaneous gradient, η_c is the learning rate for neural-controller.

For both the neural identifier and controller, n = 3 and m = 3. For the neural identifier, p equals 3. The hidden neurons are 7 and 5 for the identifier and controller, respectively. All inputs to the identifier and the controller are scaled in the range of [-1, +1].

5.4.2 Training of RNN PSS

The training procedure is the same as in the Chapter 4. There are two stages for training. At first, both the neural identifier and the controller are trained off-line. The training data covers a wide range of the generating unit operating condition, the active power output ranging from 0.1 *pu* to 1.0 *pu*; the power factor ranging from 0.8 lead to 0.8 lag. The disturbances to the system include voltage reference step change, input torque step change and three phase to ground fault. After the off-line training is finished, the weights of the identifier and the controller will be updated on-line using the algorithm described in Chapter 4.

5.5 Simulation Studies

5.5.1 RNN PSS installed on one generator

For five generator units in the system, only generator unit #3 is installed with the proposed RNN PSS. The system operating condition is the same as in the previous section with the same disturbance. The accelerating power and the speed deviation of generator #3 are sampled at the sampling period of 30 ms. The system responses are shown in Fig. 5.4 and Fig. 5.5. The RNN PSS on generator #3 can provide satisfactory damping to the local oscillation mode in $\Delta\omega_2 - \Delta\omega_3$. However, it has little influence on the inter-area oscillation between generator #1 and #2. This is because the rated capacity of generator #3 is much smaller than that of generators

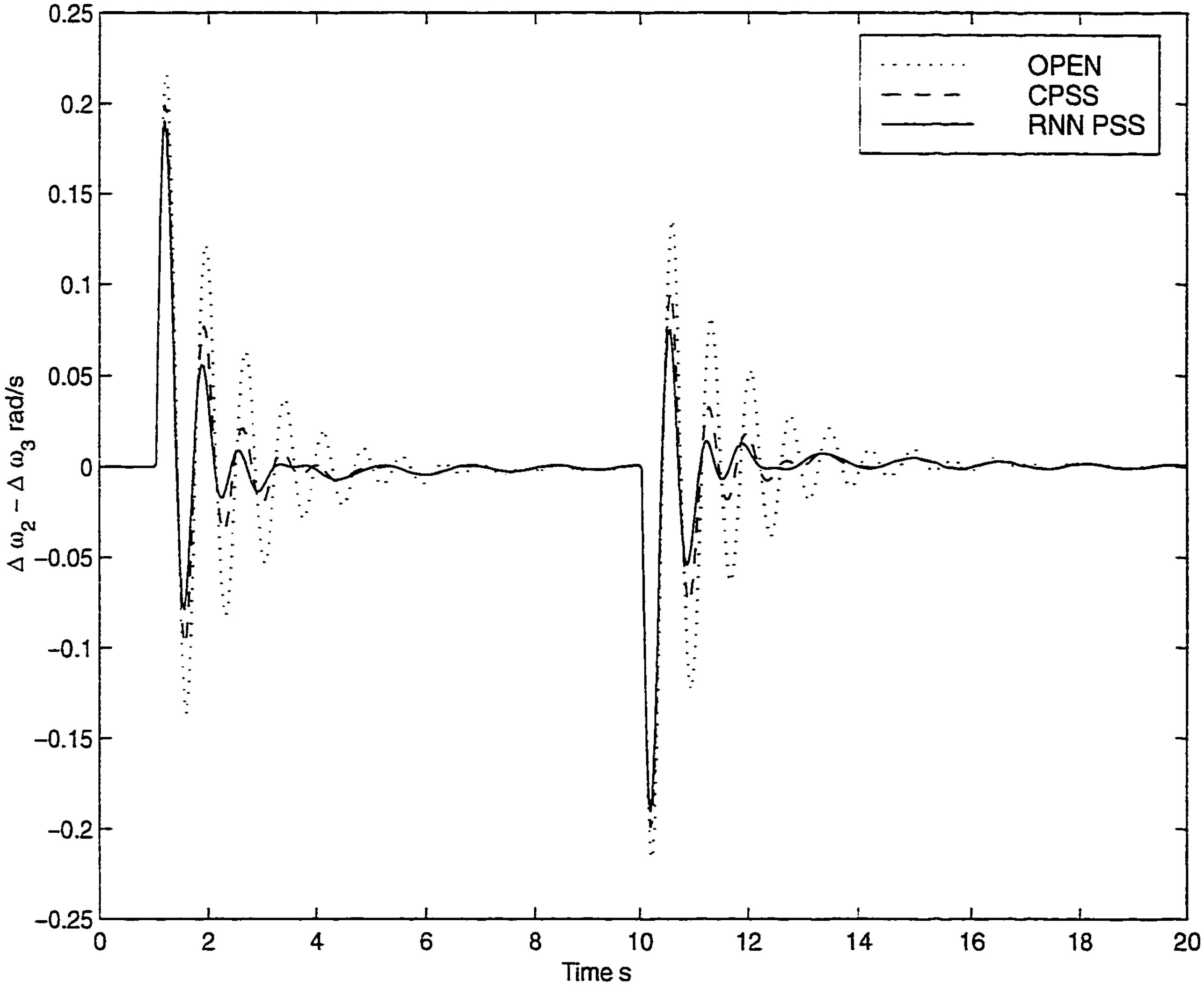


Figure 5.4. System response ($\Delta\omega_2 - \Delta\omega_3$) to input torque step change ±0.10 p.u. at generator 3, PSS installed on generator 3

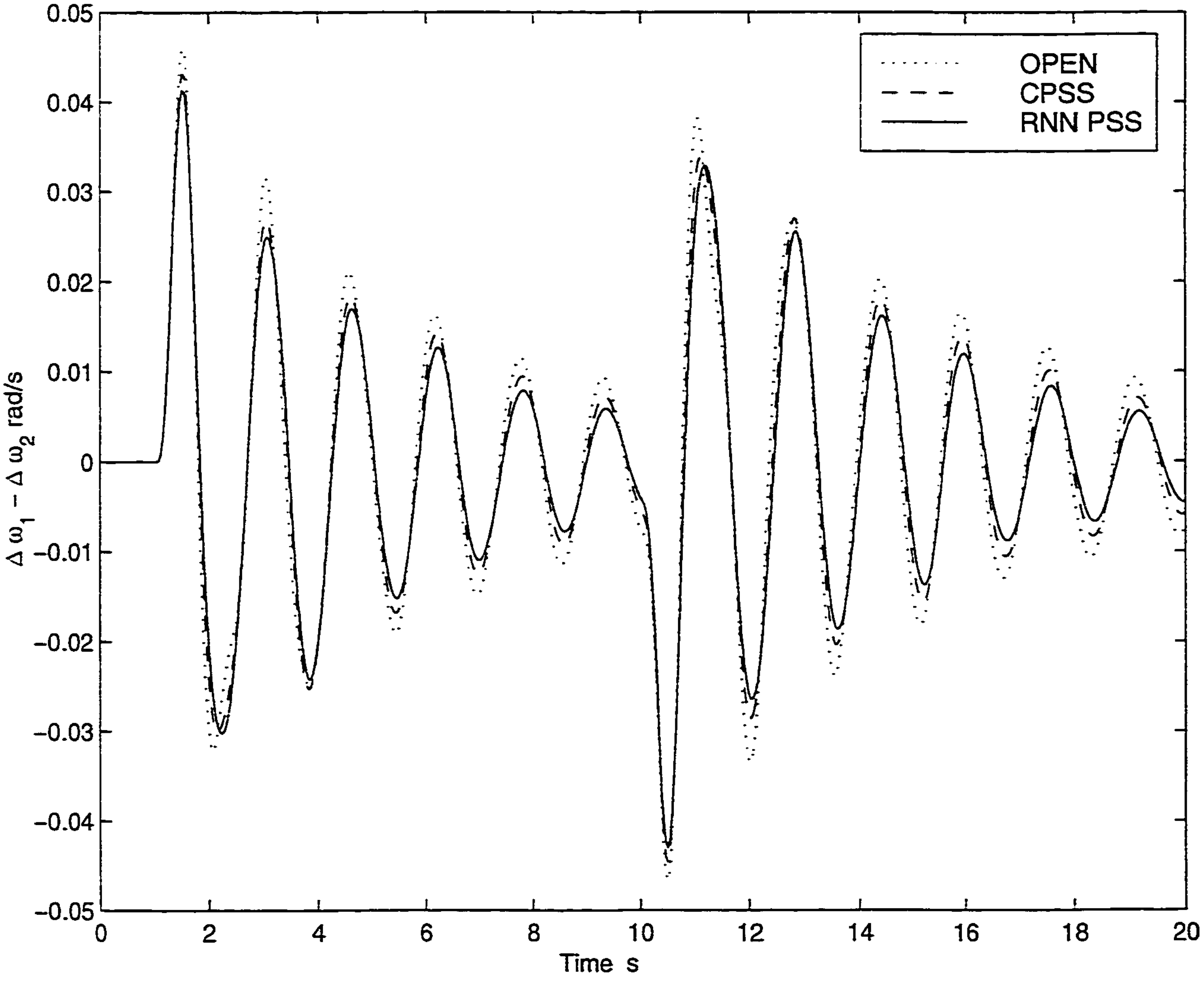


Figure 5.5. System response ($\Delta\omega_1 - \Delta\omega_2$) to input torque step change ±0.10 p.u. at generator 3, and PSS installed on generator 3.

#1 and #2. Generator #3 does not have enough power to control the inter-area oscillation between generators #1 and #2.

To compare the performance of the proposed RNN PSS and conventional PSS (CPSS), a CPSS with the following transfer function is installed on generator #3:

$$U_{pss}(s) = K_s \frac{sT_5}{1+sT_5} \frac{1+sT_1}{1+sT_2} \frac{1+sT_3}{1+sT_4} \Delta P_e(s) \tag{5.7}$$

The parameters of the CPSS are tuned carefully so that the CPSS has almost the same performance as the RNN PSS.

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

The responses with CPSS installed on generator #3 are also shown in Fig. 5.4 and Fig. 5.5.

5.5.2 RNN PSS installed on three generators

In the previous test, the results show that it is not enough to install one RNN PSS to damp both local and inter-area modes of oscillation. Two RNN PSSs with the same initial off-line training weights are installed on generators #1 and #2. The RNN PSS on generator #3 was kept as the previous study. The operating condition and the disturbance are the same as in the previous test. The system responses are shown in Fig. 5.6 and Fig. 5.7. It is very clear that both modes of oscillations are damped out very effectively.

In a seperate test, with one CPSS installed on generator #3, additional CPSSs are installed on generators #1 and #2 to damp the inter-area modes of oscillation. However, their parameters are retuned from CPSS on generator #3. The following parameters are set for the CPSS on generator #1 and #2.

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

The responses of the system with CPSS installed on generators #1, #2 and #3 are also shown in the Fig. 5.6 and Fig. 5.7.

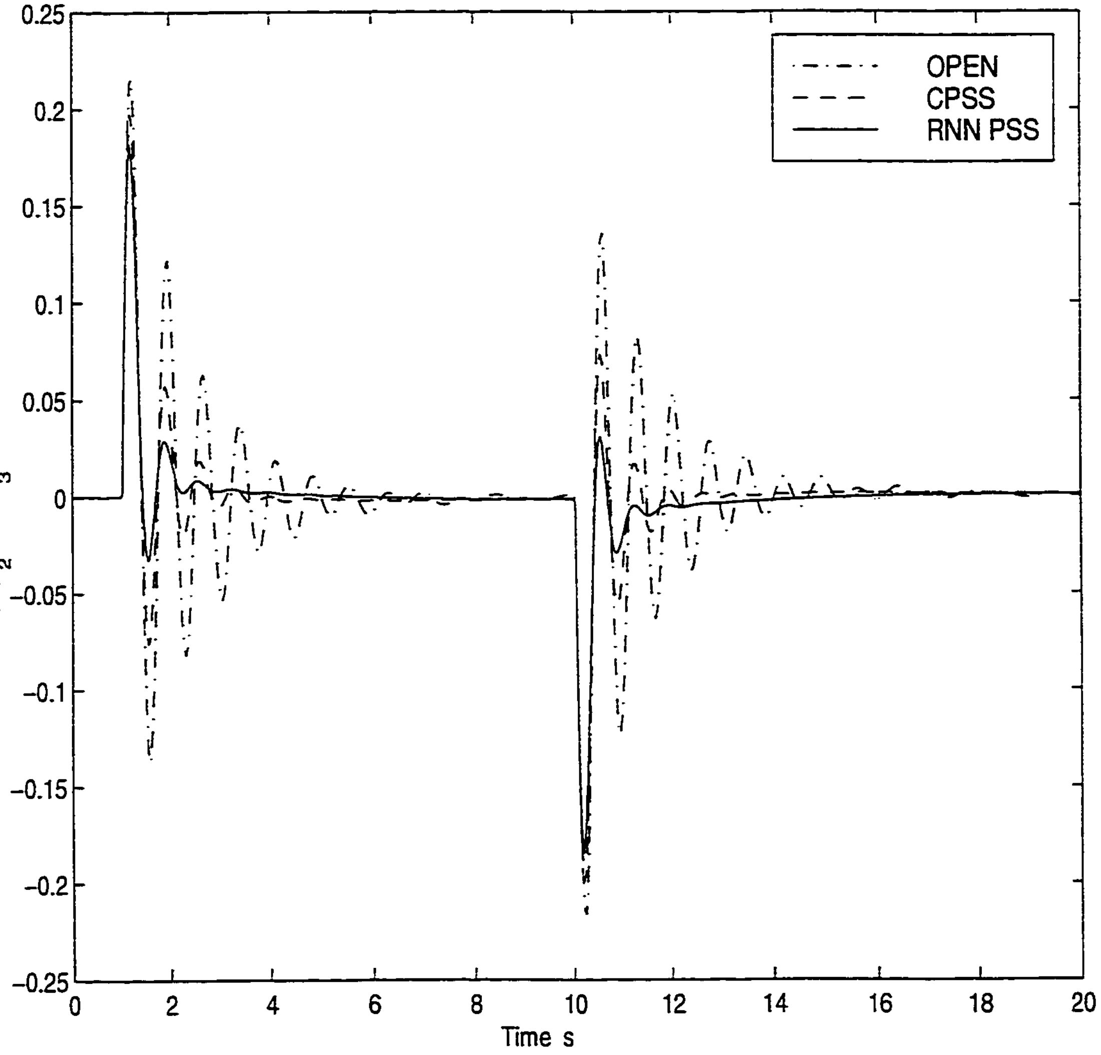


Figure 5.6. System response ($\Delta\omega_2 - \Delta\omega_3$) to input torque step change ±0.10 p.u. at generator 3, PSSs installed on generators 1, 2, 3

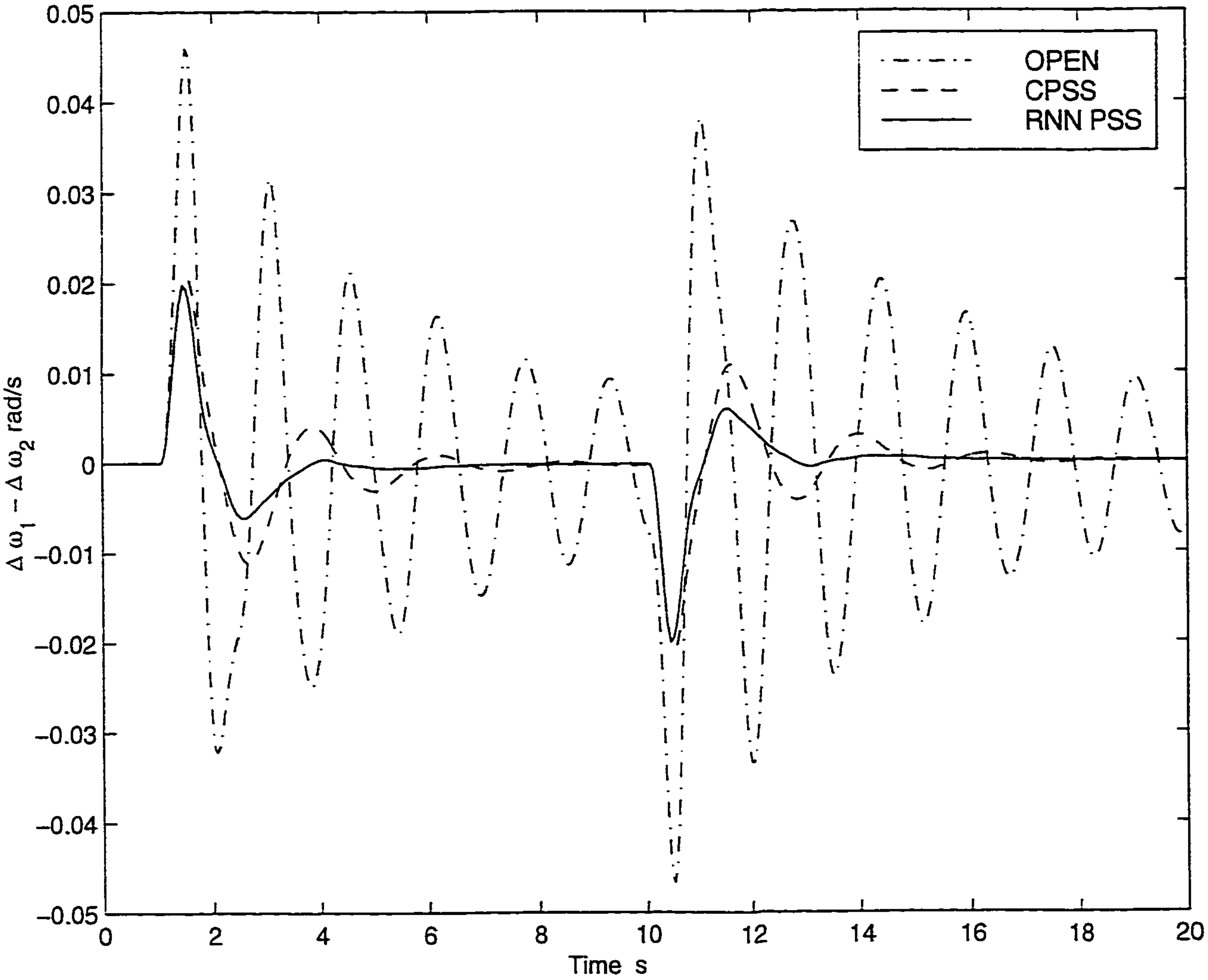


Figure 5.7. System response ($\Delta\omega_1 - \Delta\omega_2$) to input torque step change ±0.10 p.u. at generator 3, and PSSs installed on generators 1, 2, 3.

5.5.3 Coordination Between RNN PSS and CPSS

For the proposed RNN PSS, both of its input signals are local signals. The RNN PSS will coordinate itself with other PSSs based on the signals it received. On the other hand, the advanced PSSs wouldn't replace all the CPSSs being operated in the system at the same time. Therefore, the effect of the RNN PSSs and CPSSs working together needs to be investigated. In this test, the proposed RNN PSSs are installed on generators #1 and #3: CPSSs with proper parameters are installed on generators #2, #4 and #5. The operating condition is the same as in the previous tests. A 0.2 *p.u.* step decrease in the mechanical input torque reference of generator #3 is applied at 1 *s*, and return to the original level at 10 *s*. The system responses are shown in Fig. 5.8 and Fig. 5.9. The results demonstrate that two types of PSSs can work cooperatively to damp oscillations in the system.

5.5.4 Three Phase to Ground Fault Test

In the previous tests, the type of disturbance is input torque step change. In this test, a three phase to ground fault is applied at the middle of one transmission line between buses #3 and #6 at 1 *s* and the faulty line is removed 100 *ms* later. At 10 *s*, the faulty line is restored successfully. The proposed RNN PSSs are installed on generators #1, #2 and #3. The system responses are shown in Fig. 5.10 and Fig. 5.11. The results with CPSSs installed on the same generators are also shown in the same figures. From the system responses, it can be concluded that the CPSS can damp the oscillations caused by such a large disturbance, however, the proposed RNN PSS has a better performance.

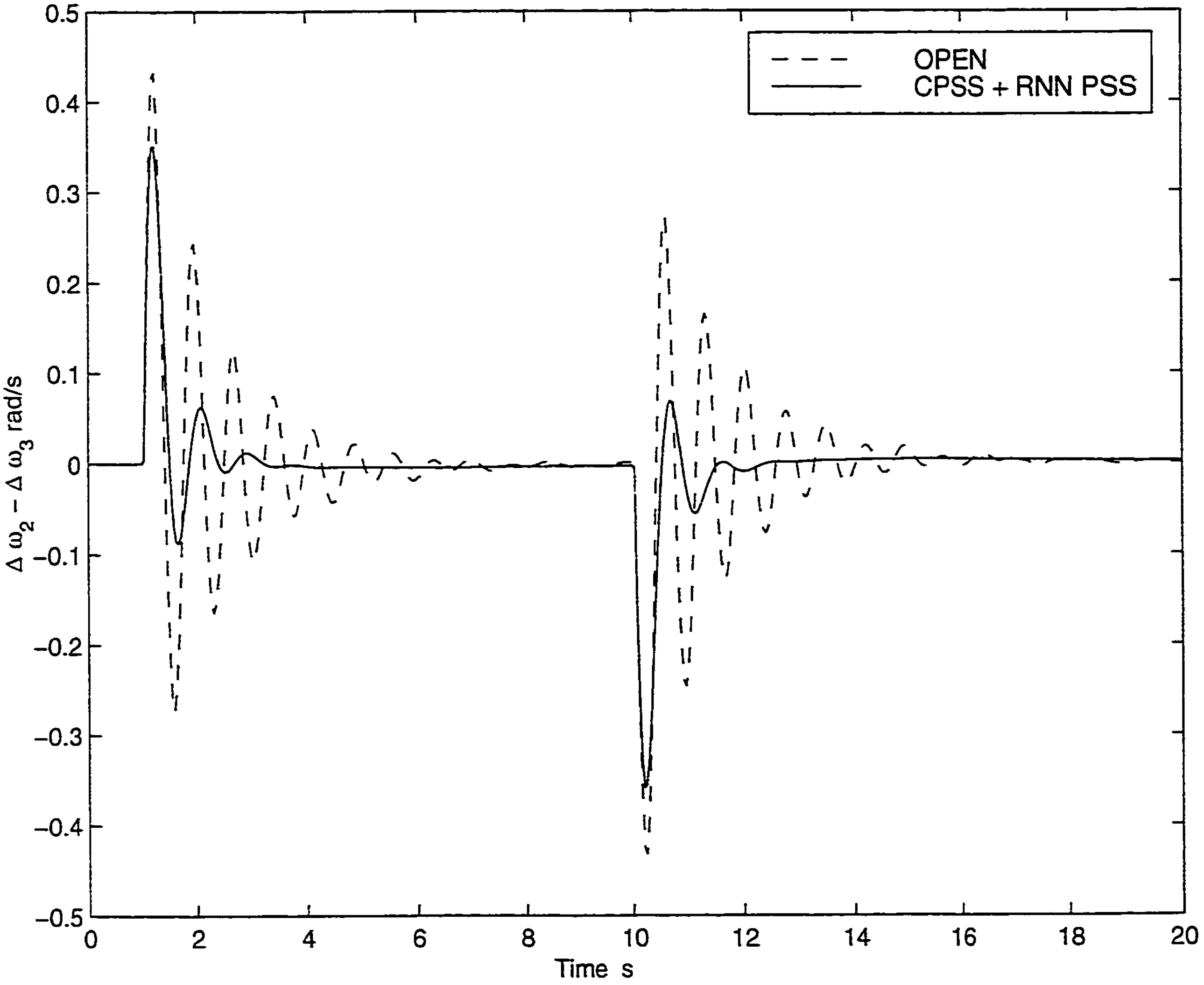


Figure 5.8. System response ($\Delta\omega_2 - \Delta\omega_3$) to input torque step change ±0.20 p.u. at generators 3, and RNN PSS installed at generators 1, 3; CPSS installed on generator 2, 4 and 5

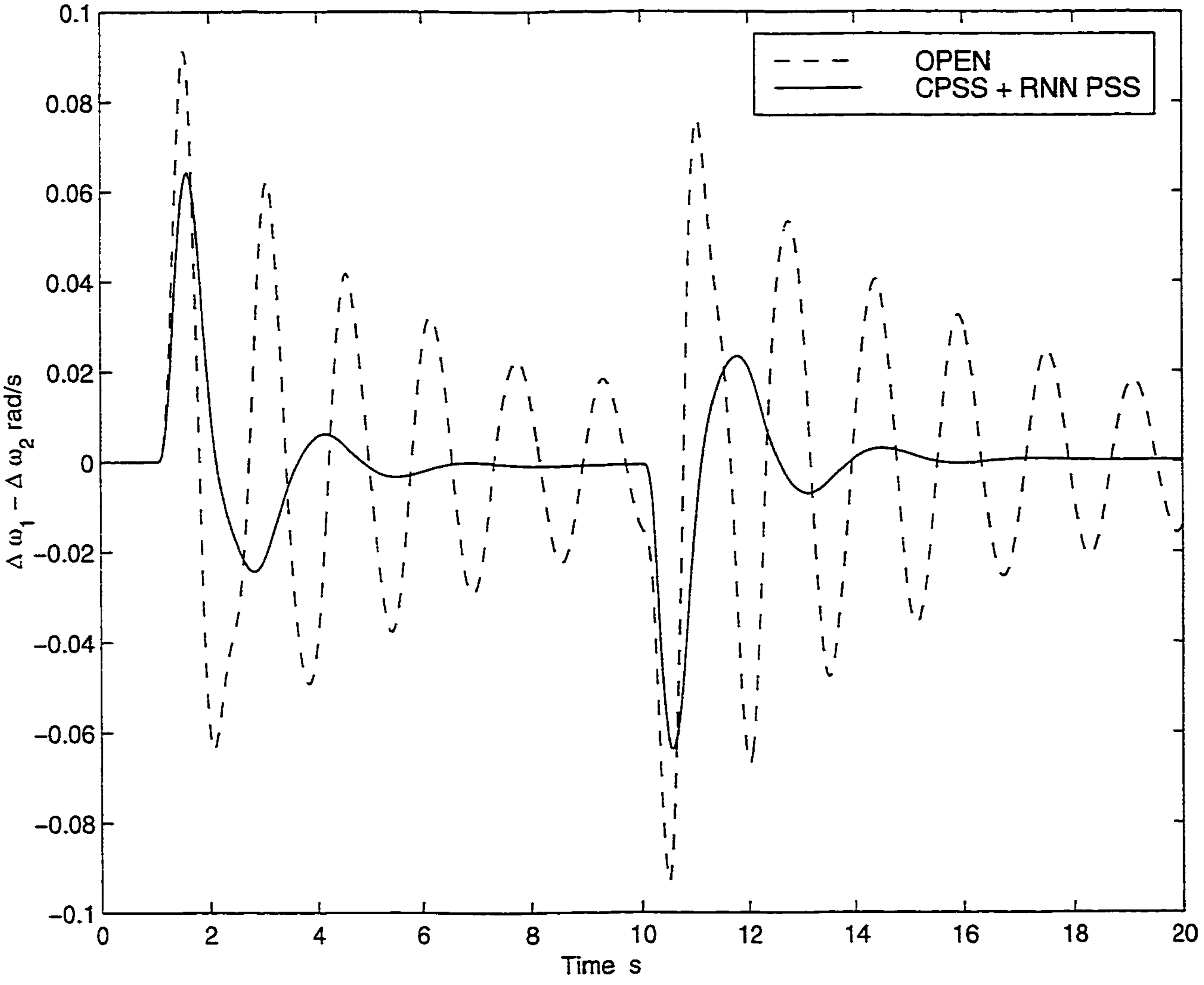


Figure 5.9. System response ($\Delta\omega_1 - \Delta\omega_2$) to input torque step change ±0.20 p.u. at generator 3, and RNN PSS installed at generators 1, 3; CPSS installed on generators 2, 4, 5.

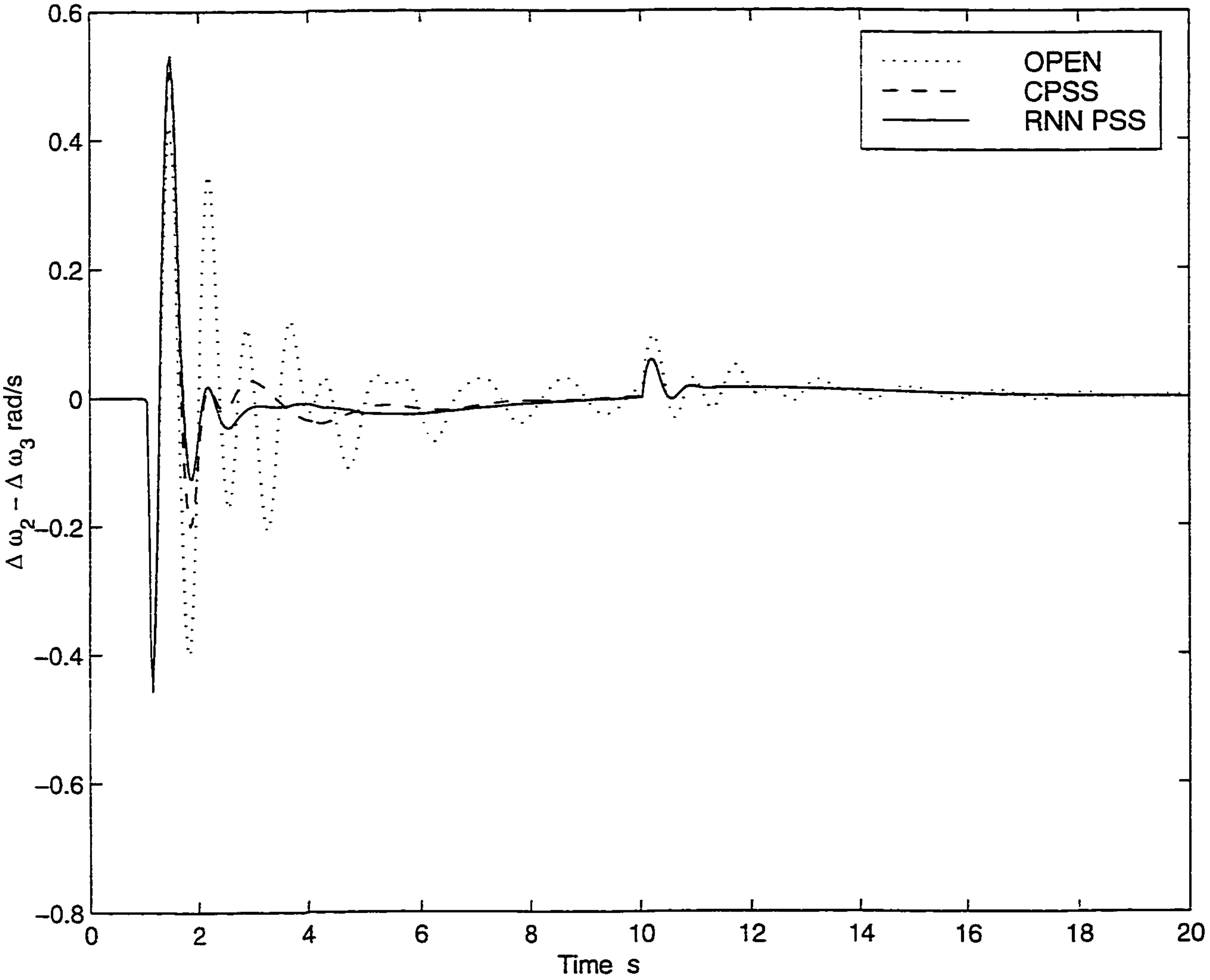


Figure 5.10. System response $(\Delta\omega_2 - \Delta\omega_3)$ to three phase to ground fault, and PSSs installed on generators 1, 2, 3

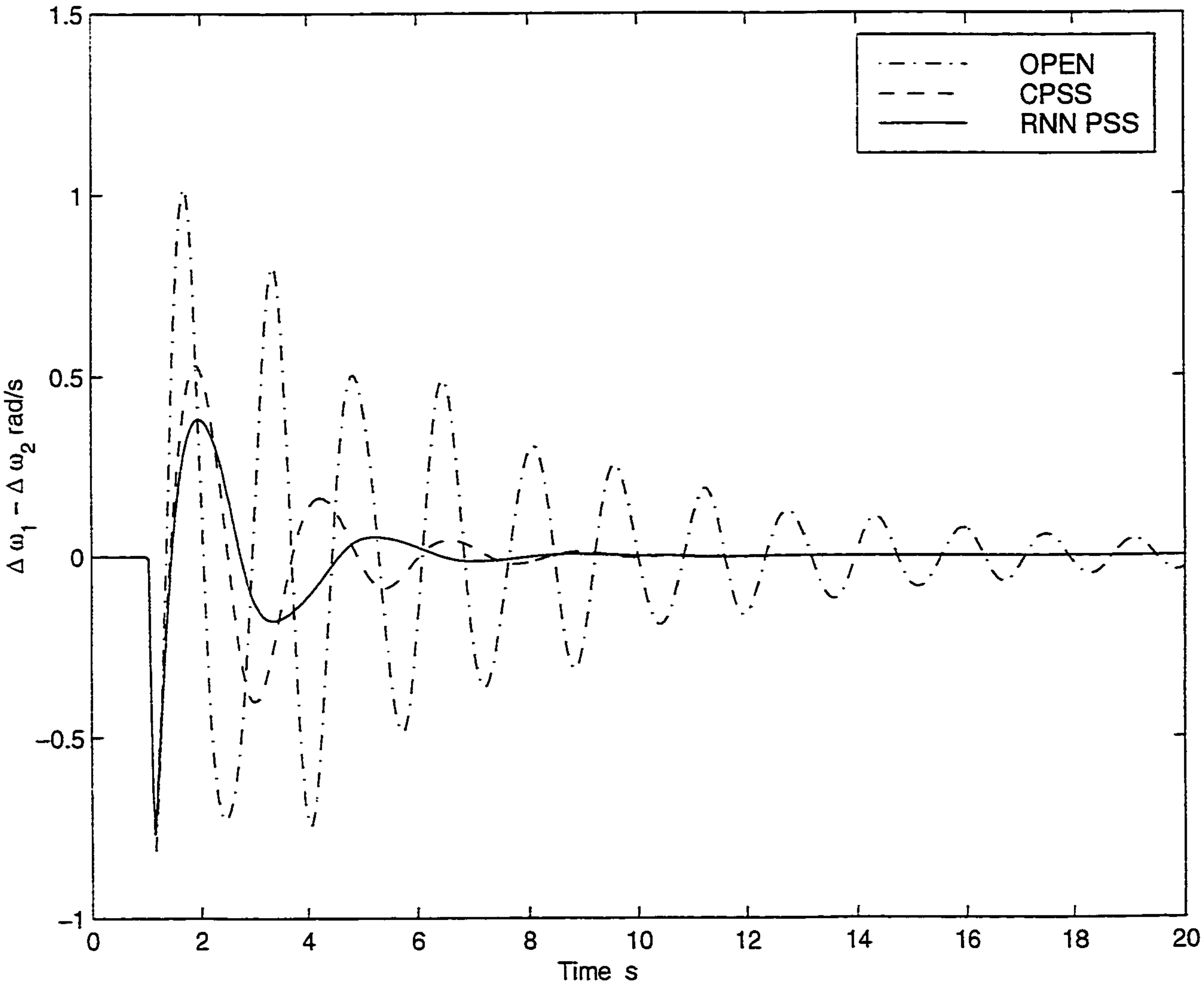


Figure 5.11. System response ($\Delta\omega_1 - \Delta\omega_2$) to input three phase to ground fault, and PSSs installed at generators 1, 2, 3.

5.5.5 New Operating Condition Test

For a power system, its operating condition is changing from time to time. To test the behaviour of the proposed RNN PSS under a different situation, the operating point of the power system is changed to a new point in which the load of the power system is 50% of the previous one, the active and re-active power output of all the generators are decreased. The data information of new operating condition is given in Appendix B.

First, a 0.15 *p.u.* step decrease in the mechanical input torque reference of generator #3 is applied at 1 *s*, and returned to the original level at 10 *s*. The studies are conducted under three different situations: no PSS installed on any generator; RNN PSSs installed on generators #1, #2 and #3; CPSSs installed on generators #1, #2 and #3. The system responses are shown in Fig. 5.12 and Fig. 5.13. It can be seen that the RNN PSS can provide a better performance under the new operating condition.

Second, a three phase to ground fault as described in Section 5.5.4 is applied to the system. The system responses with no PSS installed; RNN PSSs installed on generators #1, #2 and #3; CPSSs installed on generators #1, #2 and #3 are shown in Fig. 5.14 and Fig. 5.15. Again, the results clearly demonstrate that RNN PSS can damp the oscillations more effectively than the CPSS. This is because the proposed RNN PSS is an adaptive controller which can track the changes in the power system, whereas the CPSS is a fixed parameter controller designed based on the system linearized at one operating point.

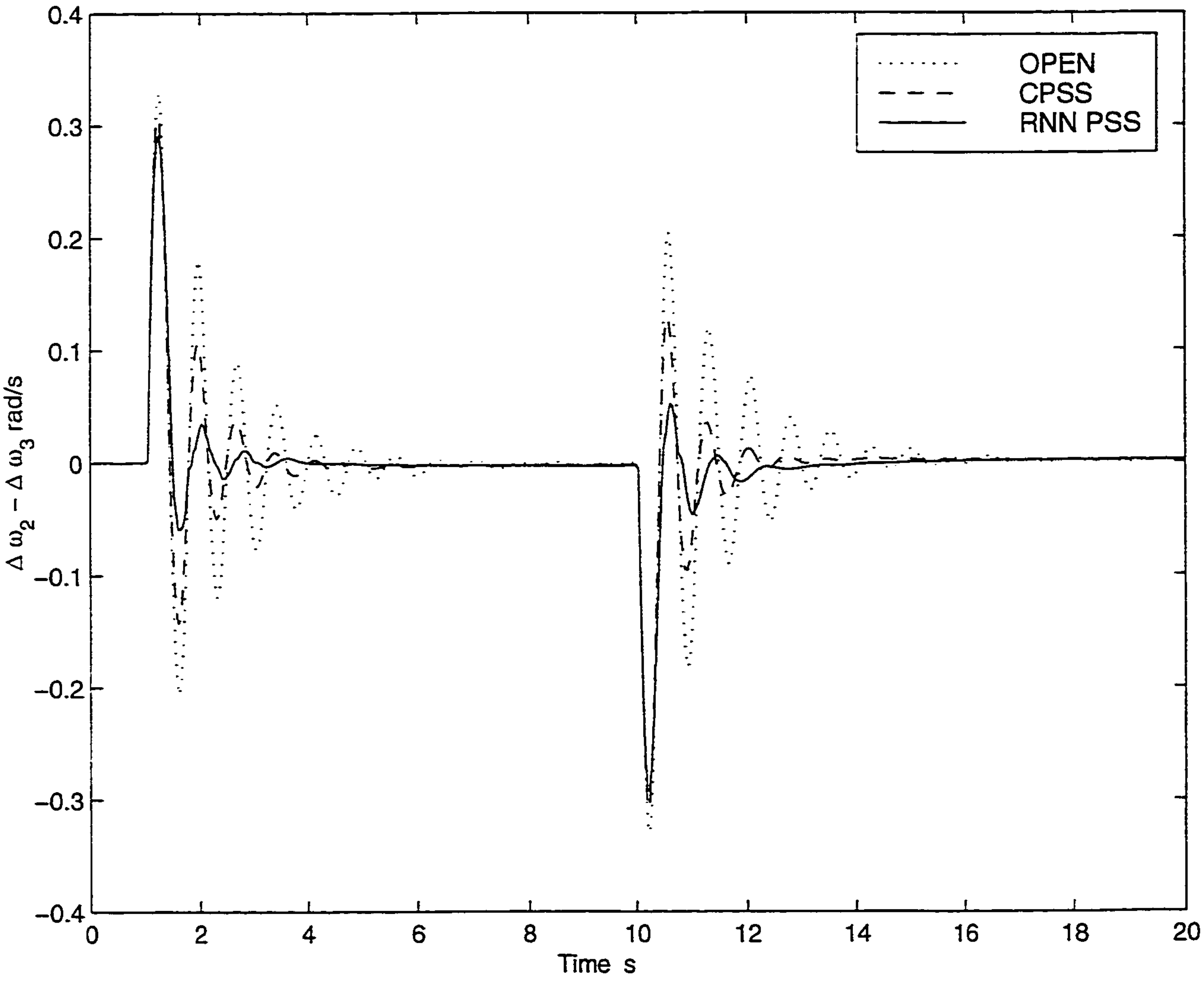


Figure 5.12. System response ($\Delta\omega_2 - \Delta\omega_3$) to torque disturbance, and PSSs installed on generators 1, 2, 3 for the new operating condition

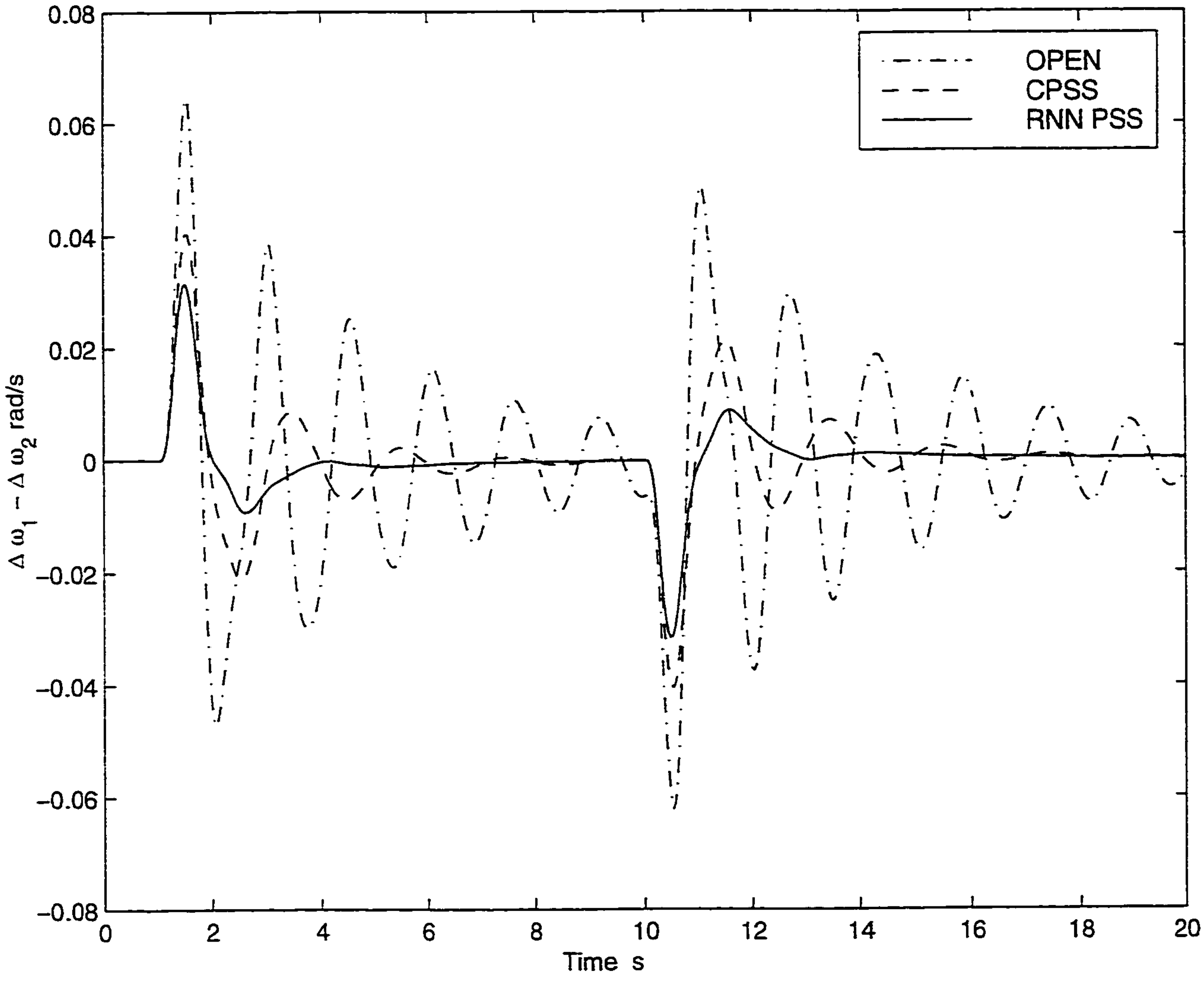


Figure 5.13. System respcnse $(\Delta\omega_1 - \Delta\omega_2)$ to torque disturbance, and PSSs installed at generators 1, 2, 3 for the new operating condition

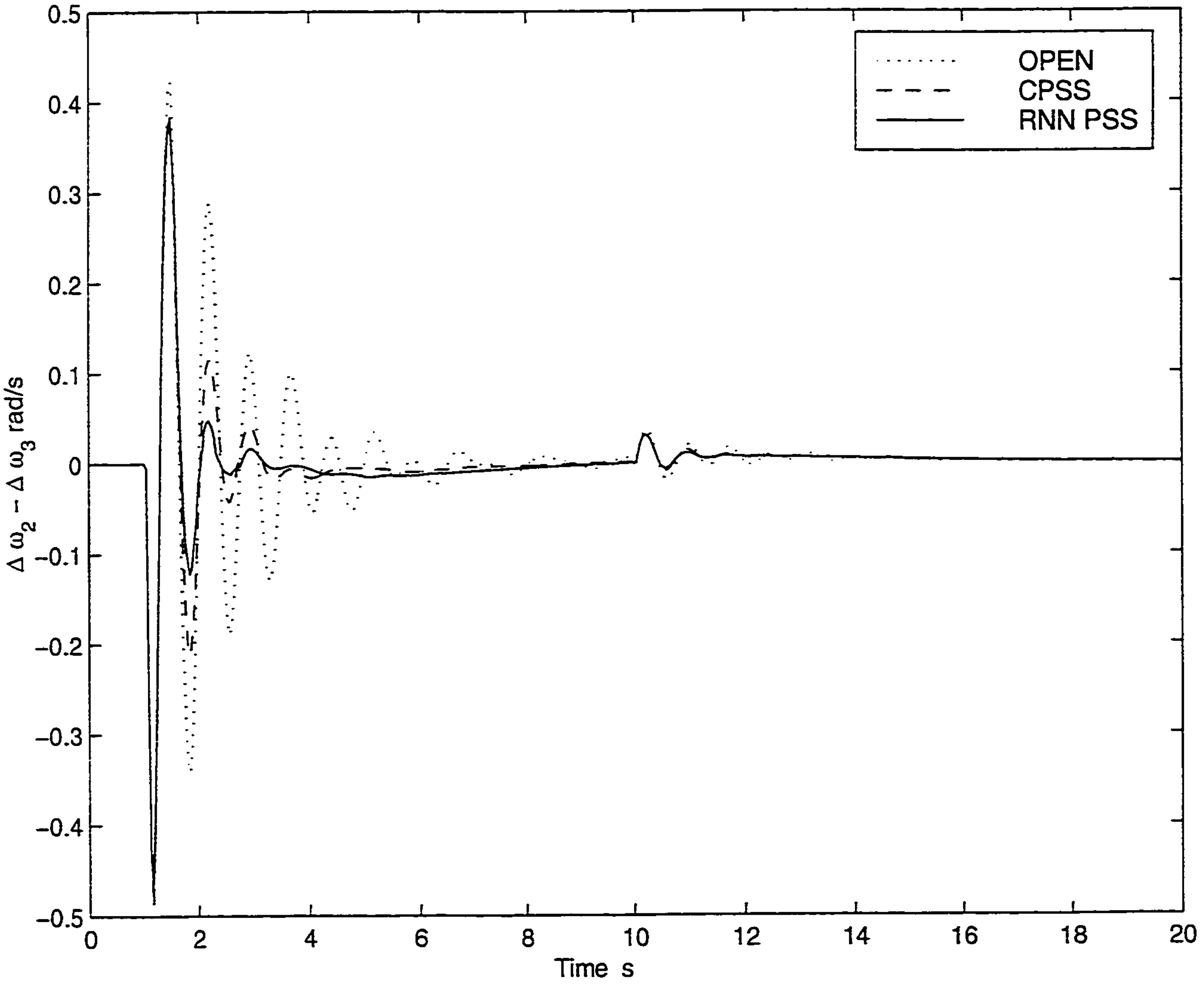


Figure 5.14. System response ($\Delta\omega_2 - \Delta\omega_3$) to three phase to ground fault, and PSSs installed on generators 1, 2, 3 for the new operating condition

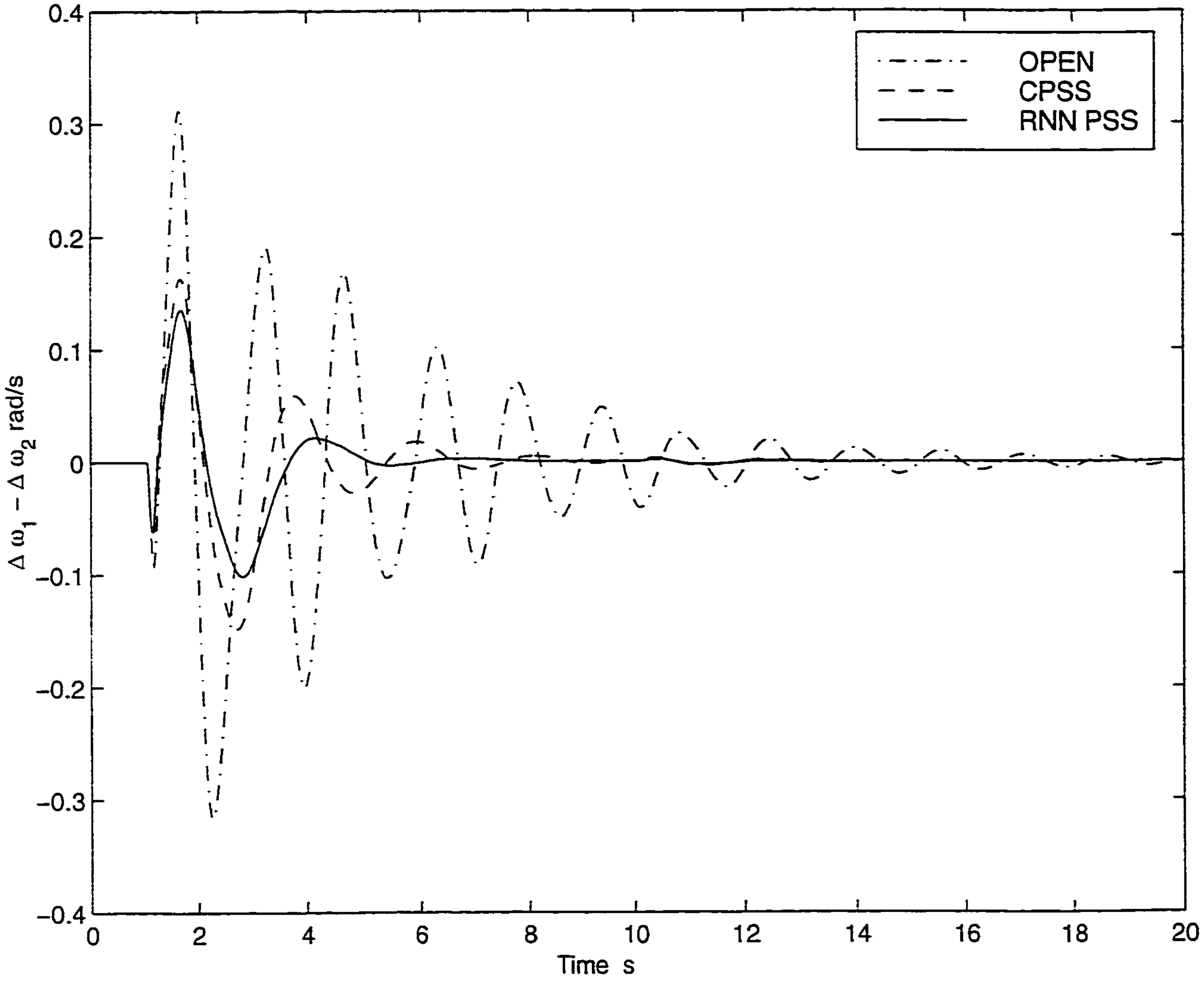


Figure 5.15. System response $(\Delta\omega_1 - \Delta\omega_2)$ to input three phase to ground fault, and PSSs installed at generators 1, 2, 3 for the new operating condition

5.6 Summary

In this chapter, the effectiveness of an RNN PSS to damp multi-mode oscillations in a five machine power system is investigated. The structure of the RNN PSS is the same as that developed in Chapter 4. The generator accelerating power and speed deviation are used as the input signals to the RNN PSS. However, to remove the DC value of the speed deviation, a washout filter is employed in speed deviation signal loop. The RNN PSS was trained over a wide operating range of the generating unit with various disturbances.

Simulation studies show that to damp the oscillations in multi-machine system, it depends on not only the PSS but also on the location at which the PSS is installed and the coordination of the PSSs. The proposed RNN PSS can provide satisfactory results if correctly installed and can cooperate with other RNN PSSs or CPSSs.

Until now, it is demonstrated that the proposed RNN PSS can effectively damp the oscillations both in single machine and multi-machine systems. These studies are based on computer simulation. In the next chapter, the performance of the proposed RNN PSS will be investigated in the real-time environment.

CHAPTER 6

LABORATORY IMPLEMENTATION OF RNN PSS AND EXPERIMENTAL RESULTS

6.1 Introduction

Computer simulation studies in the previous chapters have demonstrated that the proposed RNN PSS has a very good damping ability for both the single mode oscillation in the single-machine infinite bus system and the multi-mode oscillations in the multi-machine system. In these studies, the power system is represented by a selected mathematical model. However, not all the features of a real power system can be accurately modeled by a mathematical model, also the operating conditions of the physical systems are not ideal and there exists noise. The performance of the RNN PSS should be invesigated in a physical environment of a power system. However, the practical and economical way is to do the testing on a scaled physical power system model in the laboratory environment. Scaled physical models, which can simulate the behaviour of the actual plant in the laboratory environment, are extensively used in the universities and research laboratories to test new control schemes [104] [105] [106] [107] [108] [109].

Several laboratory experiments about neural network PSS are reported in recent years. A PSS based on multilayer perceptron network was implemented in [110]. A Fuzzy Logic PSS based on neural network was studied in [111]. Both research results demonstrated that properly trained PSS based on a neural network can also provide satisfactory performance.

In this chapter, implementation and testing of the RNN PSS in a scaled physical power system environment is reported. The RNN PSS is developed on a TMS320C30 Digital Signal Processor mounted on a HOST PC. A digital conventional PSS was also implemented in the same physical environment. The experimental results of RNN PSS and CPSS are discussed.

6.2 Physical Model of a Power System

Schematic of a power system model for ANN PSS implementation test is shown in Fig. 6.1 and its parameters are listed in Appendix C. The model is used to simulate a single-machine infinite-bus power system.

A 3 kVA, 220 V, 3-phase synchronous micro-alternator driven by a separately excited DC machine is acting as the generating unit. A time constant regulator (TCR) is used to set the effective field time constant of the micro-alternator to simulate the time constant of a large generator. With TCR, the effective field time constant of the micro-alternator can be set as large as 10s.

The transmission line in the physical model simulates a 500 kV, 300 km, double circuit transmission line. Each circuit consists of 6 Π sections connected in series. Each Π section represents a 50 km length of the actual transmission line. To simulate the fault conditions on the transmission line, three circuit breakers are installed on one transmission line, two at each end of the line and one in the middle.

An ABB AVR implemented on a PHSC2 Programming Logic Controller (PLC) is used as the micro-alternator AVR. Three phase ac terminal voltages and currents are stepped down, rectified and filtered with a cut-off frequency of 8 *Hz* to form DC signals to be fed to the PLC. Based on these signals, the PLC calculates the field control signal which is output to the TCR, and the active power signal P_e which is the input signal for the PSS.

Various disturbances can be applied to this system model to investigate the performance of the proposed RNN PSS. By adjusting the voltage reference setting of the AVR, the generator terminal voltage can be stepped up or down. By changing the field current of the DC motor, the active power output of the micro-alternator

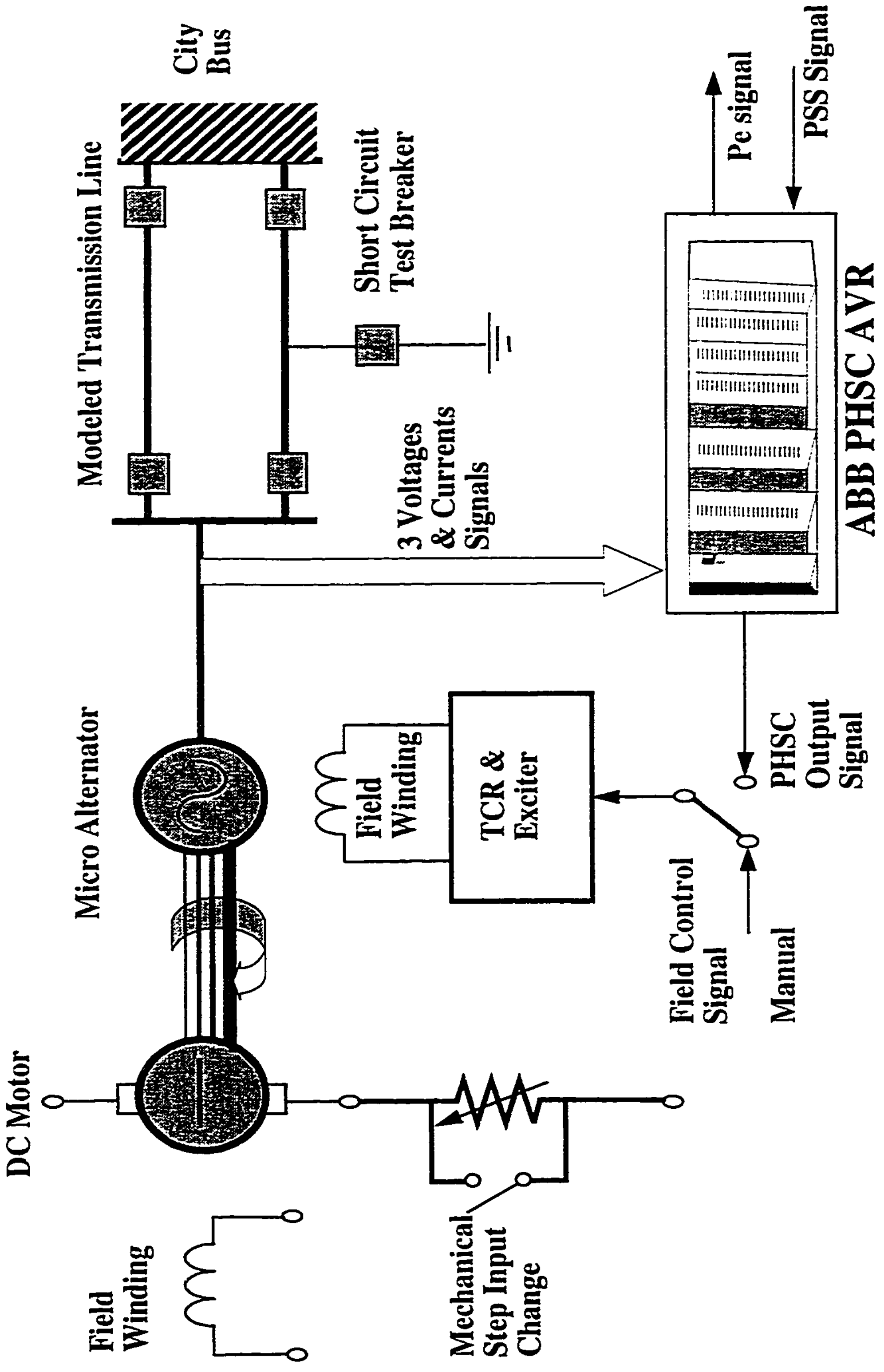


Figure 6.1. Schematic diagram of lab setup

can be changed. By setting the operation sequence of three circuit breakers on the transmission line, three-phase to ground fault can be simulated.

6.3 RNN PSS Implementation

6.3.1 Real Time Digital Control Environment

The structure of a Real Time Digital Control Environment is shown in Fig. 6.2. The ABB PHSC2 System acts as the AVR of the micro-alternator. The core of the digital control system is a TMS320C30 system board supplied by SPECTRUM Signal Processing Inc. [112] [113] The board has a TMS320C30 Digital Signal Processsor (DSP) [114] [115] [116]. It can be operated at 33.3 MHz clock. It is also provided with 8KB on-chip RAM and 32KB on-chip ROM. The board contains a complete "analog I/O subsystem". There are two separate channels, each containing its own sample/hold amplifier, A/D, D/A, and analog filters on input and ouput. The sampling rates can be up to 200 kHz.

The P_e signal required by the DSP board is calculated in the PLC based on 3 phase voltage and current sample sigals. The DSP board receives the P_e signal through A/D channel and stores in the dual access ram (DARAM). The control algotithm loaded on DSP board will read the data in DARAM and calculate the control signal U_{pss}. Then the U_{pss} is sent out through the D/A channel to the PLC. The PLC adds the U_{pss} signal to the voltage reference signal. The combined signal goes through the PLC block to generate the field control signal to the TCR.

The input switch box is used for several purposes. First, it is used to switch on or off the PSS application to the generating system. Second it can set the internal reference voltage to be stepped up or down. Finally, it is used to apply reference voltage disturbances.

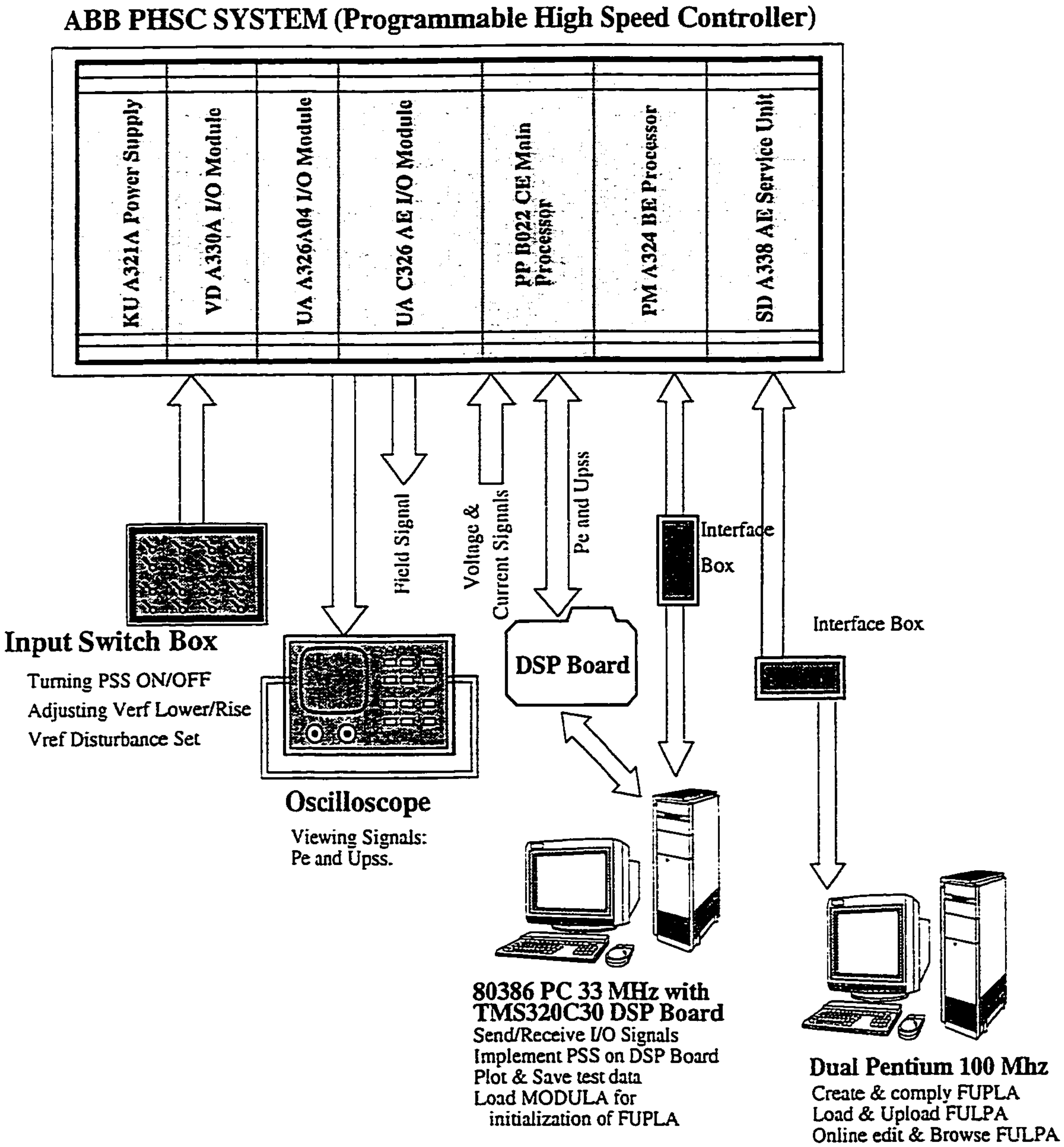


Figure 6.2. Structure of digital control system

6.3.2 Software Structure

The software required for the real-time control environment and design of digital controllers is developed through TMS320 floating-point C compiler [115] [116], assembler and linker. The C compiler can translate standard ANSI C programs into TMS320C3x/C4x assembly language source. Later, the assembler translates assembly language source files into machine language Common Object File Format (COFF) object files. Finally the linker combines object files into executable object module.

The software implementation of the RNN PSS has two parts. The first one is a Man Machine Interface (MMI). It runs on the HOST PC. The flow chart of the MMI is shown in Fig. 6.3. The second one is the core of RNN PSS. It includes the control algorithm of the RNN PSS. The second part of the code is loaded to the DSP board by MMI during initializing procedure, and runs on the DSP board. The flow chart of this program is shown in Fig. 6.4. The two programs communicate with each other using interrupt method, and exchange data through DARAM on the DSP board.

In the real-time operation, the tasks for the module running on HOST PC are:

- initialize the variable address for PC-DSP communication,
- download the second module from HOST PC to DSP board memory,
- read the parameters of controller from data file,
- send the data to DARAM and starts the control action,
- receive sample and control data from DSP board, displays and saves data.

The tasks for the module running on the DSP board are:

- initialize the address for DSP-PC communication,
- receive controller parameters from PC,

- set the timer for sampling,
- read sample data from input channel,
- calculate control signal,
- send results to DARAM and output channel.

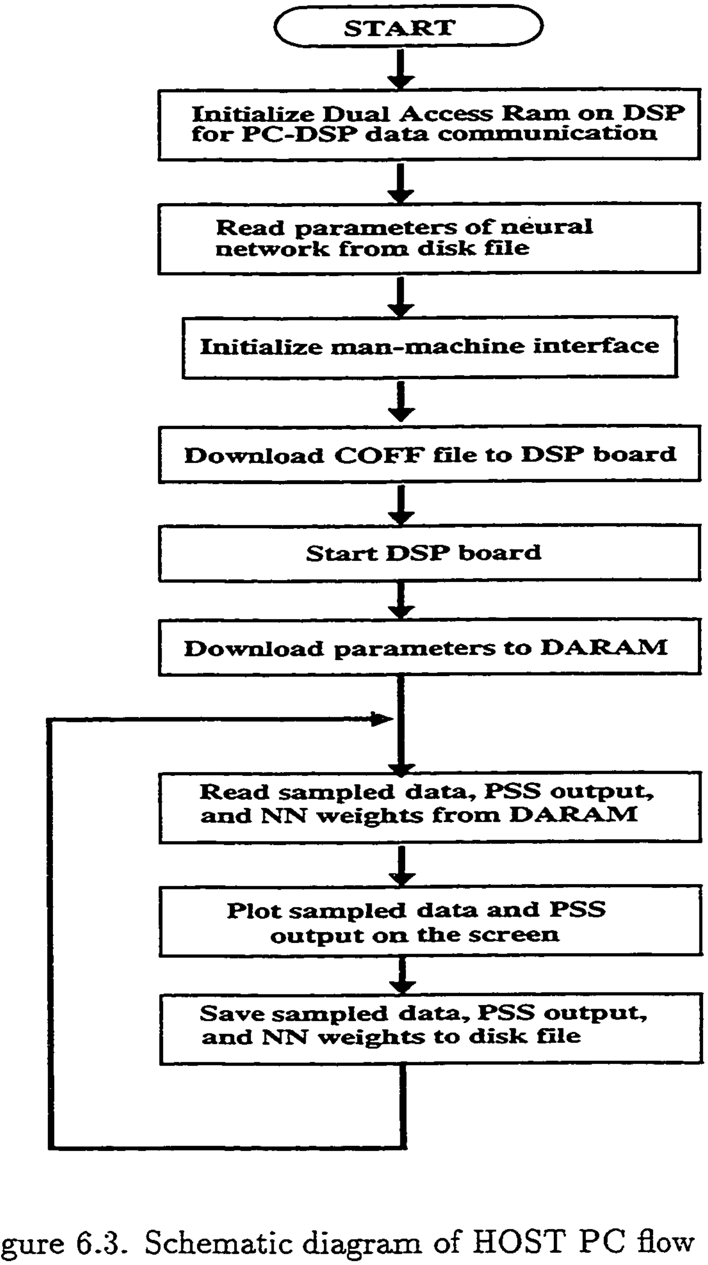

START
Initialize Dual Access Ram on DSP for PC-DSP data communication
Read parameters of neural network from disk file
Initialize man-machine interface
Download COFF file to DSP board
Start DSP board
Download parameters to DARAM
Read sampled data, PSS output, and NN weights from DARAM
Plot sampled data and PSS output on the screen
Save sampled data, PSS output, and NN weights to disk file

Figure 6.3. Schematic diagram of HOST PC flow chart

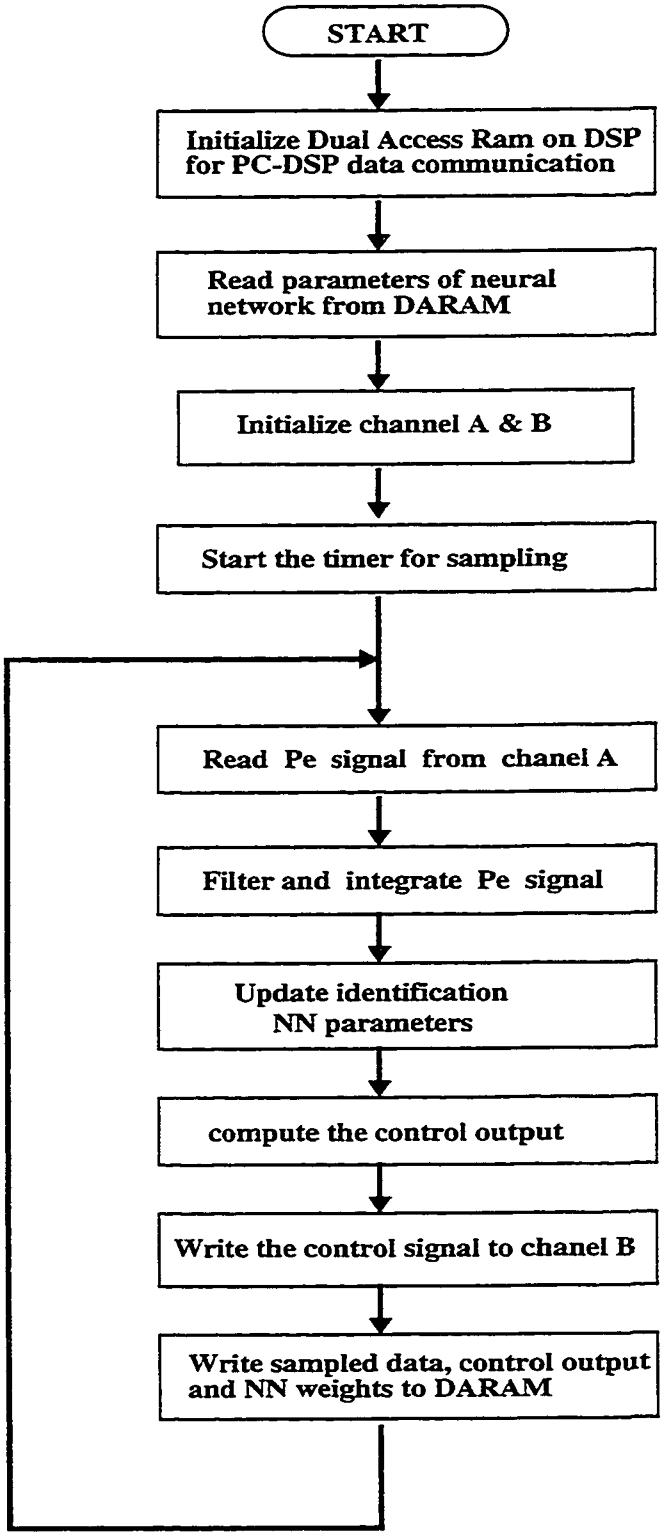

START
Initialize Dual Access Ram on DSP for PC-DSP data communication
Read parameters of neural network from DARAM
Initialize channel A & B
Start the timer for sampling
Read Pe signal from chanel A
Filter and integrate Pe signal
Update identification NN parameters
compute the control output
Write the control signal to chanel B
Write sampled data, control output and NN weights to DARAM

Figure 6.4. Schematic diagram of DSP flow chart.

6.4 RNN PSS Structure and Training

6.4.1 RNN PSS Structure

The structure of the RNN PSS for this study is shown in Fig. 6.5. There are two recurrent neural networks in this architecture which function as neural identifier and neural controller, respectively.

For the neural-identifier, the input vector is:

$$[\Delta\omega'(k), \ldots, \Delta\omega'(k-m), \Delta Pe(k), \ldots, \Delta Pe(k-n), U_{annc}(k), \ldots, U_{annc}(k-p)] \quad (6.1)$$

where $\Delta\omega'(k)$, $\Delta Pe(k)$ and $U_{annc}(k)$ are the integral of the generator accelerating power, accelerating power and the PSS signal, respectively at the k-th time step. The output is $\Delta\hat{P}e(k+1)$, the predicted accelerating power at time step (k+1).

For the neural-controller, the input vector is,

$$[\Delta\omega'(k), \ldots, \Delta\omega'(k-m), \Delta Pe(k), \ldots, \Delta Pe(k-n)] \quad (6.2)$$

The output is $U_{annc}(k)$, the PSS control signal at time step k. For both the neural identifier and controller, n = 3 and m = 3. For the neural identifier, p equals 3. The hidden neurons are 7 and 5 for the identifier and controller, respectively. Both the identifier and the controller each have one output neuron. Therefore, for the neural identifier, there are 12 input neurons, 7 hidden neurons and one output neuron; for the neural controller, there are 8 input neurons, 5 hidden neurons and one output neuron. All inputs to the identifier and the controller are scaled in the range of [-1, +1].

The size of neural networks for both the identifier and the controller is relatively small. It will need less computation time each sampling period, which is very important for the real-time implementation.

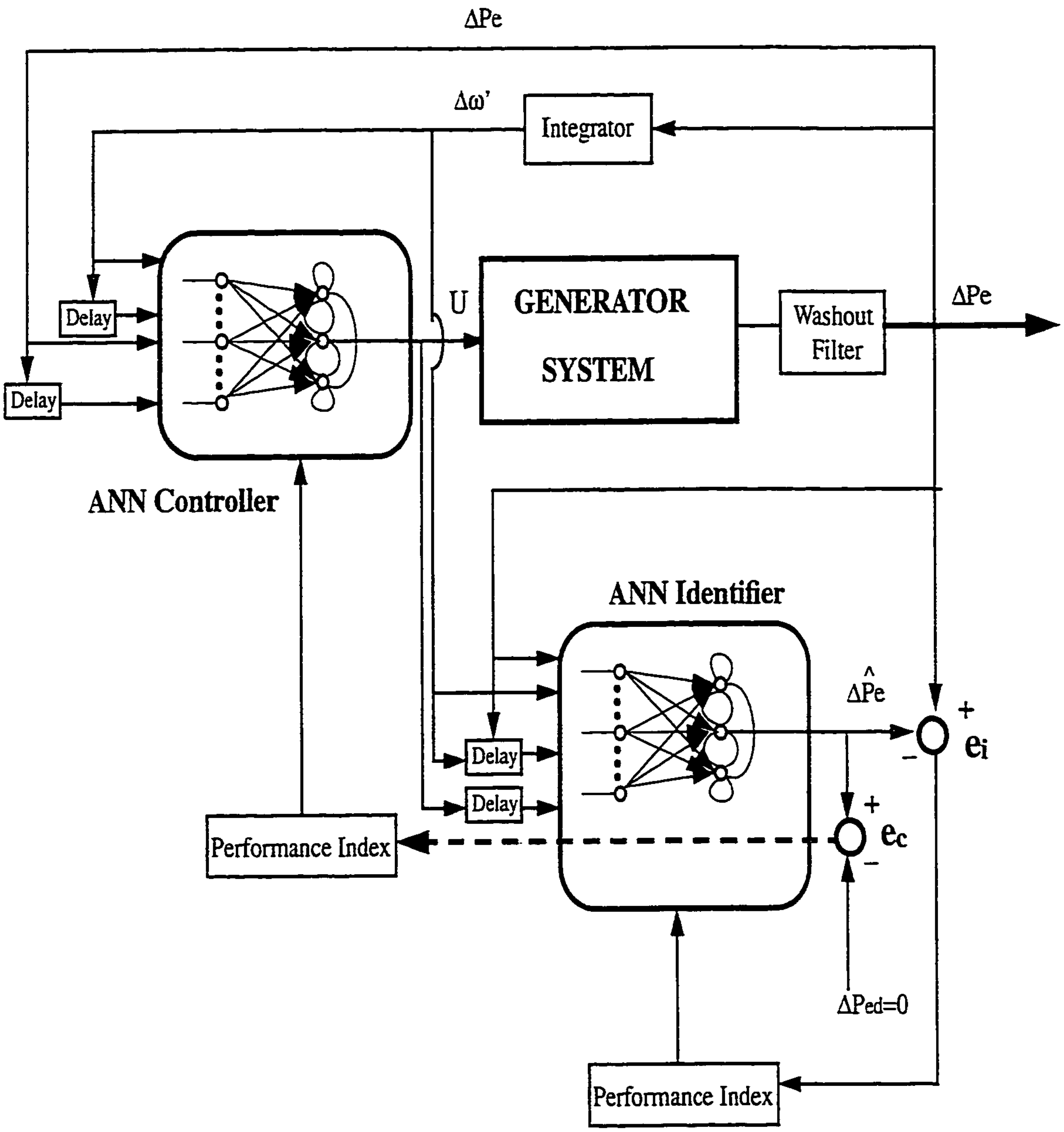


Figure 6.5. Schematic diagram of Neural Network Controller

6.4.2 Training of RNN PSS

The training procedure is the same as in Chapter 4. There are two stages for training. At first, both neural identifier and controller are trained off-line. The training data covers a wide range of the generating unit operating condition, the active power output ranging from 0.1 *pu* to 1.0 *pu*; the power factor ranging from 0.8 lead to 0.8 lag. Disturbances to the system include voltage reference step change, input torque step change and three phase to ground fault. After the off-line training is finished, weights of the identifier and the controller will be updated on-line using the algorithm described in Chapter 4.

By using on-line update method, it makes it possible for the proposed RNN PSS to track the dynamic character of the power system and provide better damping effect to the oscillations of the power system.

6.5 Implementation of CPSS

To compare the performance of the RNN PSS with the conventional PSS , a CPSS with the following transfer function [117] was implemented in the same environment.

$$U(s) = K_s \cdot \frac{sT_5}{1+sT_5} \cdot \frac{1+sT_1}{1+sT_2} \cdot \frac{1+sT_3}{1+sT_4} \cdot \Delta P_e(s) \tag{6.3}$$

The CPSS transfer function was discretized to be implemented on the DSP. The parameters of the CPSS were tuned for a particular operating point of the system described in section 6.6.1. The sampling rate for the CPSS is 100 *Hz*.

6.6 RNN PSS Laboratory Test Results and Discussion

To investigate the behaviour of the proposed RNN PSS, various studies with different disturbances under different operating conditions were conducted. All the experimental data was stored on HOST PC for further analysis. The time axis was adjusted so that the disturbance seems to happen at the desired time point. The sampling frequency for the adaptive control system is set at 25 *Hz*.

The inherent damping of the physical power system model is very high. To better evaluate the performance of the CPSS and the RNN PSS, the system inherent damping ratio was decreased by using only one of the double transmission lines in all tests except in the three phase to ground fault test.

6.6.1 CPSS Tuning

In order to make a reasonable comparison between the CPSS and the RNN PSS, the parameters of the CPSS are carefully tuned under a specific operating condition so that it provides almost the same performance. The following operating condition is set: active power is 0.74 *pu* and power factor is 0.90 *pu* lag. A 0.15 *pu* step decrease in the input torque reference was applied at 3.5 *s* and removed at 11.5 *s*. The CPSS parameters are set as follows:

$$K_s = 6.0, T_1 = T_3 = 0.1s; T_2 = T_4 = 0.075s$$

The generating unit active power deviation responses with CPSS and with RNN PSS are shown in Fig. 6.6. It can be observed that if CPSS has been tuned properly, it also provides effective damping to system oscillations at designed operating point. In the following tests, the CPSS parameters will be kept constant.

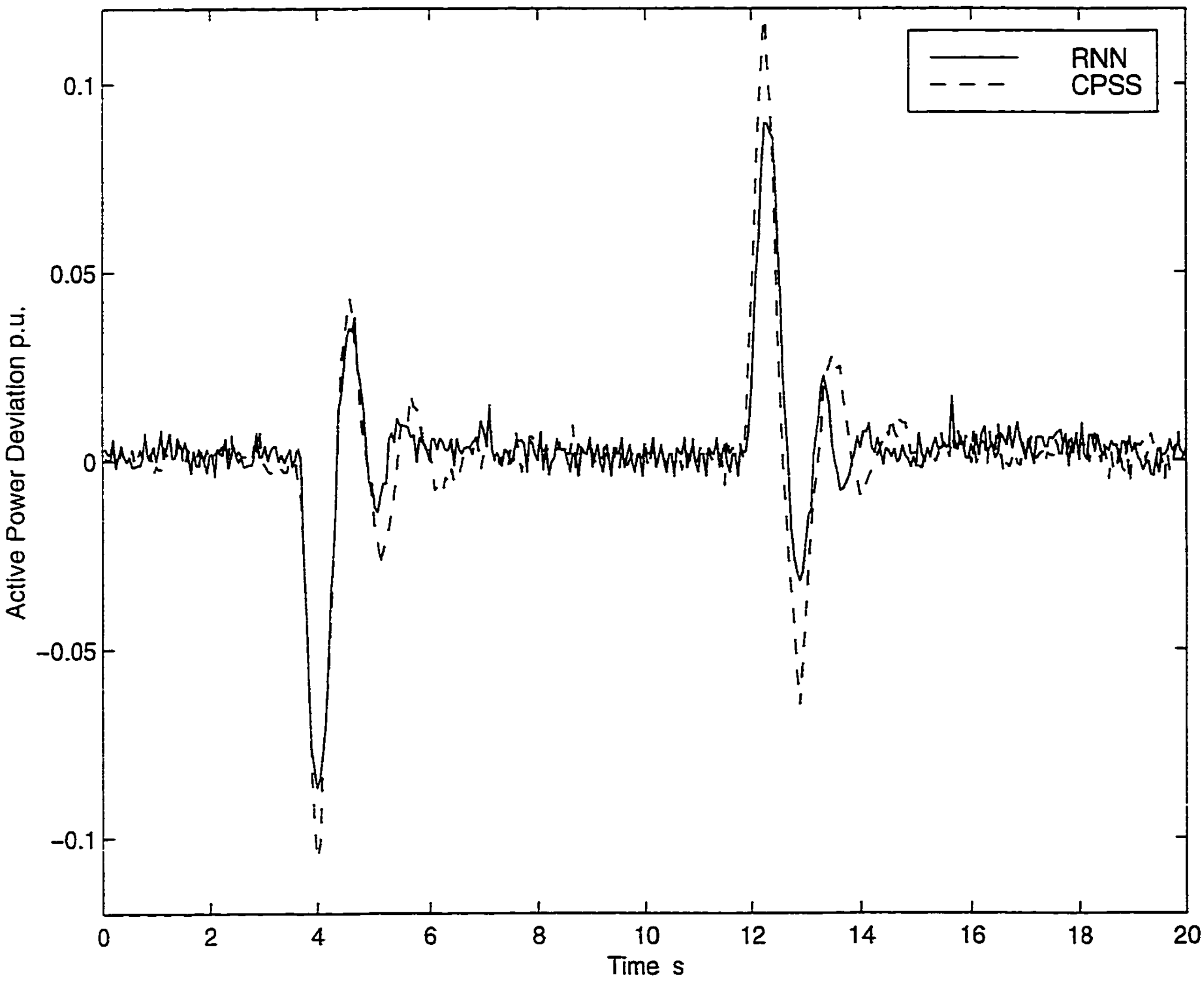


Figure 6.6. System response to a 0.15pu step change disturbance in input torque reference, P= 0.74 *pu*, *pf* = 0.90 *lag*

6.6.2 Voltage Reference Step Change

With the system operating at 0.88 *pu* active power and 0.85 *pf* lag, a 4.5% step decrease in voltage reference is applied at 3 *s*. At 15 *s*, the change in the voltage reference was removed . The generator active power deviation responses with RNN PSS and without PSS (NO PSS) are shown in Fig. 6.7. Since the system stability margin decreases as the reference voltage drops for a certain active power output, the oscillations last for 5 *s* without any PSS, and only last for 3 *s* with the RNN PSS. The generating unit active power deviation response and the supplementary control signals of RNN PSS and CPSS are shown in Figs. 6.8 and 6.9, respectively. Both the RNN PSS and the CPSS provide effective damping in this case, however, the swing magnitude of active power deviation with the RNN PSS is smaller than that with the CPSS.

To further test the performance of the RNN PSS, the operating condition is changed to 0.50 *pu* active power and 0.95 *pf* lag. A 5% step increase in voltage reference is applied at 3 *s*. At 15 *s*, the change in the voltage reference was removed . The generator active power deviation responses of RNN PSS and CPSS are shown in Fig. 6.10. Even though the operating condition is quite different from the previous one, the RNN PSS still provides better damping than the CPSS.

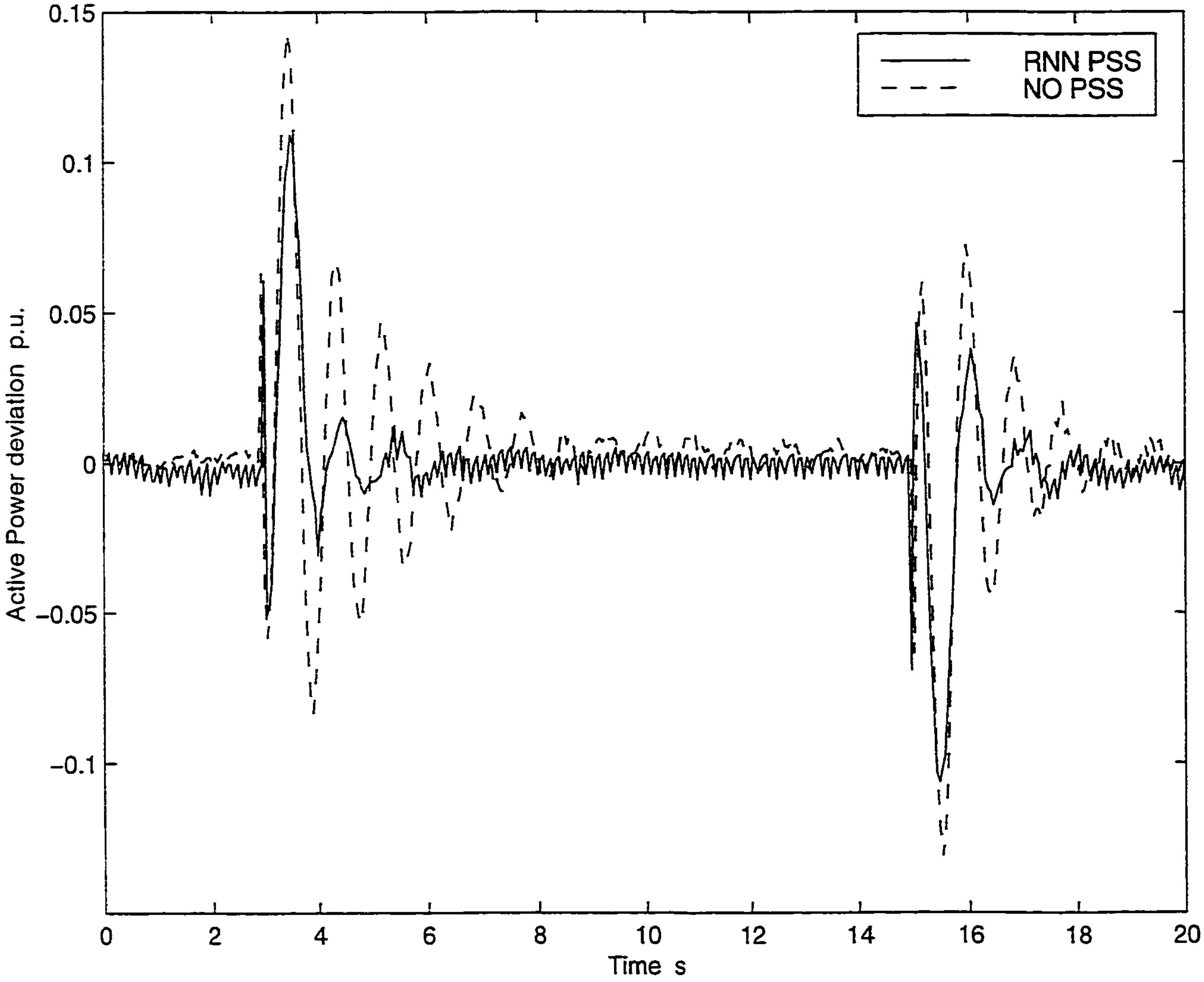


Figure 6.7. System response (with RNN PSS and NO PSS)to a 4.5% step change disturbance in voltage reference, P= 0.88 *pu*, *pf*= 0.85 *lag*

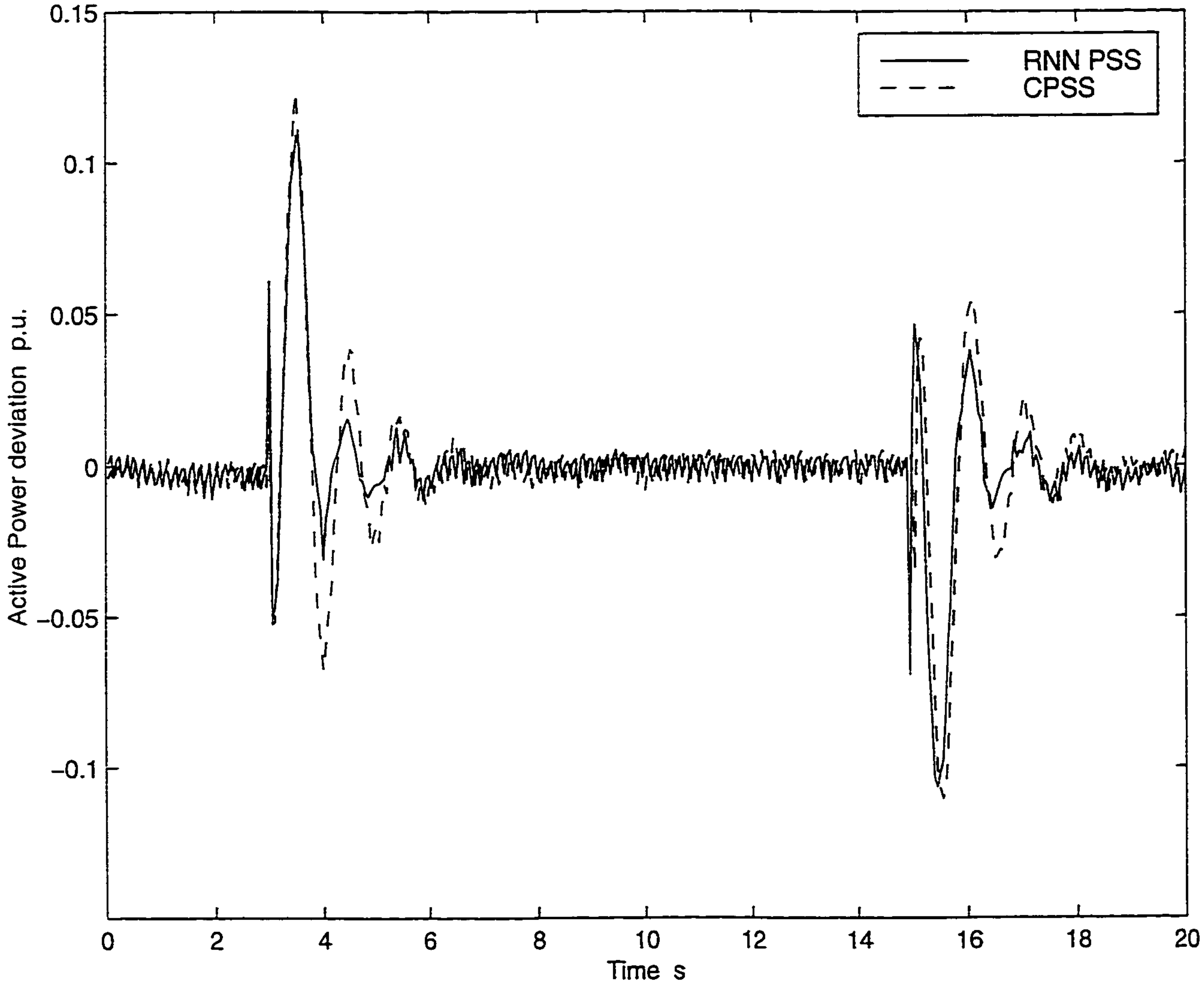


Figure 6.8. System response to a 4.5% step change disturbance in voltage reference, P= 0.88 *pu*, *pf*= 0.85 *lag*

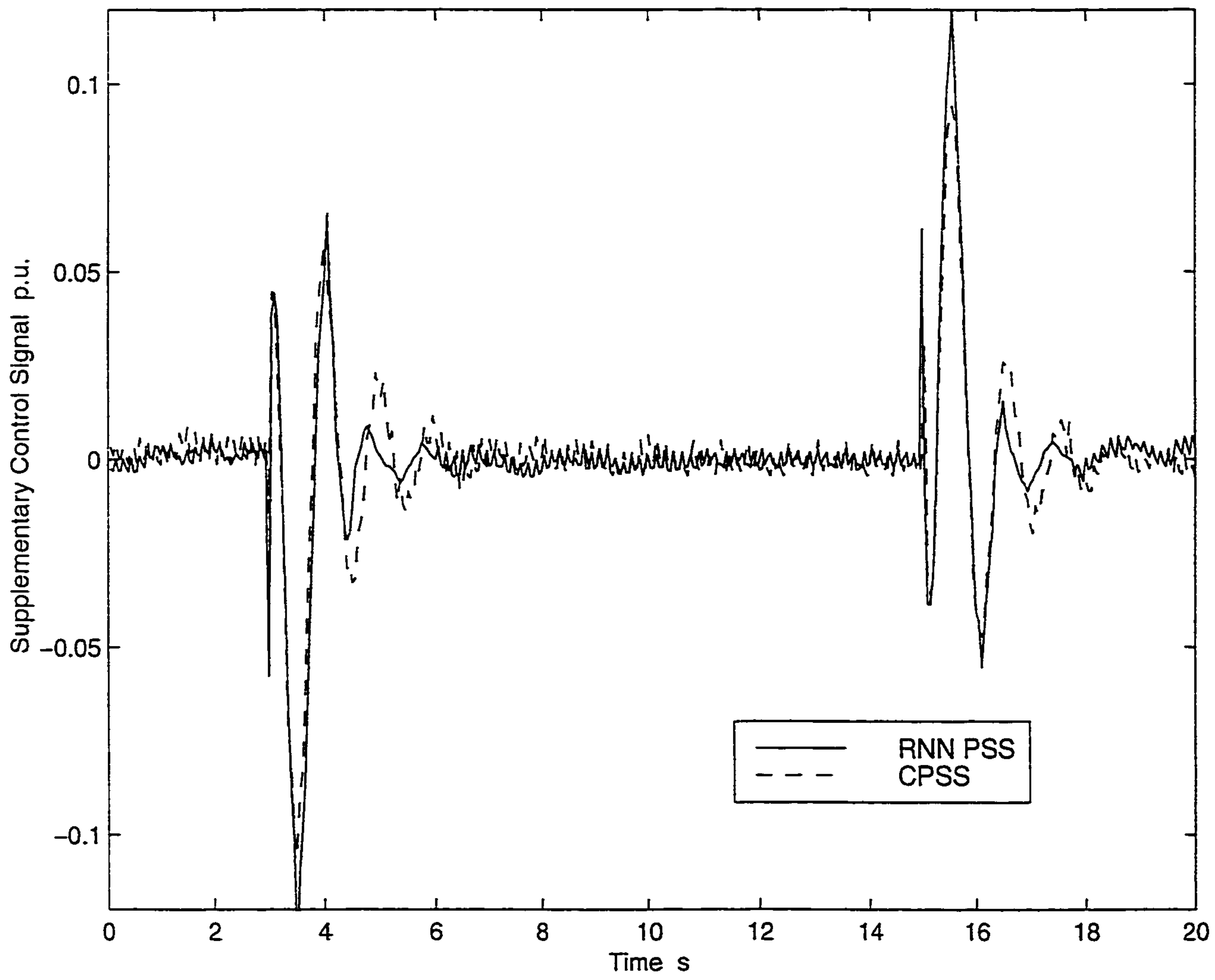


Figure 6.9. Supplementary control signals of RNN PSS and CPSS to a 4.5% step change disturbance in voltage reference, P= 0.88 *pu*, *pf* = 0.85 *lag*

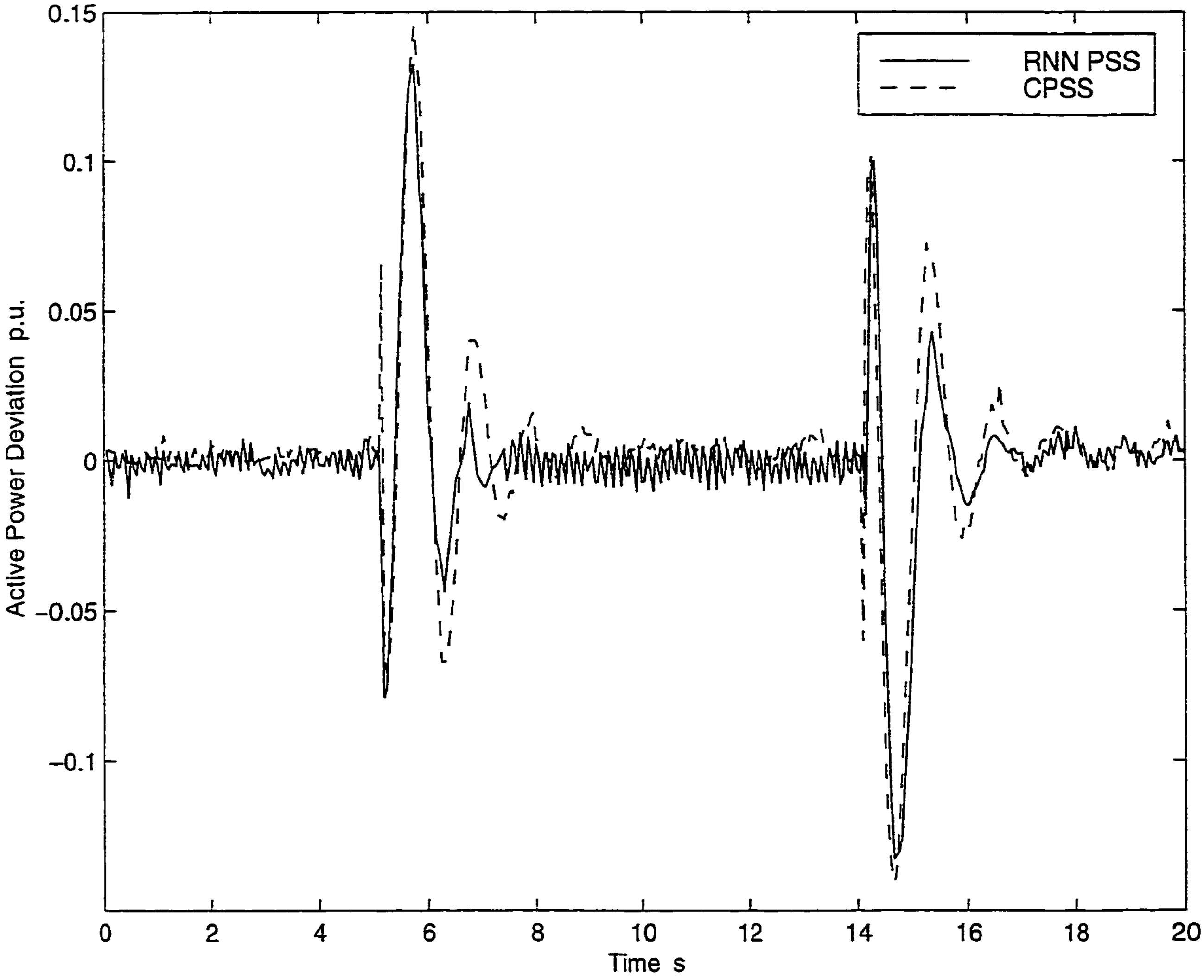


Figure 6.10. System response to a 5.0% step change disturbance in voltage reference, P= 0.50 *pu*, *pf* = 0.95 *lag*

6.6.3 Input Torque Reference Step Change

To test the RNN PSS behaviour under different types of disturbance, with the system operating at 0.90 *pu* active power and 0.95 *pf* lead, a 0.10 *pu* step decrease in the input torque reference is applied at 2.5 *s* and removed at 11 *s*. The comparison of the RNN PSS and the CPSS performance is shown in Fig. 6.11. The stability margin at the leading power factor is reduced, the RNN PSS still can provide a satisfactory response. Especially, when an increase of 0.10 *pu* input torque was applied at 11 *s*, the active power deviation oscillation under CPSS was clearly larger than that under RNN PSS.

With the system operating at 0.50 *pu* active power and 0.90 *pf* lag, a 0.20 *pu* step decrease in the input torque reference is applied at 4 *s* and removed at 11.5 *s*. The comparison of the RNN PSS and the CPSS is shown in Fig. 6.12. The power system becomes more stable at this light load condition. Since the disturbance is larger than the previous one, both RNN PSS and CPSS still need about 2 cycles to reach the stable state. However, the RNN PSS provide a better control effect.

In the third test, with the system operating at 0.90 *pu* active power and 0.90 *pf* lag, a 0.25 pu step increase in input torque reference is applied at 3 *s*. At 13 *s*, the change in the voltage reference was removed . The generator active power deviation responses of RNN PSS and CPSS are shown in Fig. 6.13. Again, the RNN PSS produces much better result in response to the disturbance.

6.6.4 Three Phase to Ground Fault Test

To investigate the performance of the RNN PSS under transient conditions, a three phase to ground fault was applied with the system operating at the following point: 0.90 *pu* power, power factor 0.90 *lag*. The three phase to ground fault was set in the middle of one transmission line at 4.2 s. The circuit breakers at both ends of the faulted line were tripped 100 *ms* later. An unsuccessful reclosure attempt was made

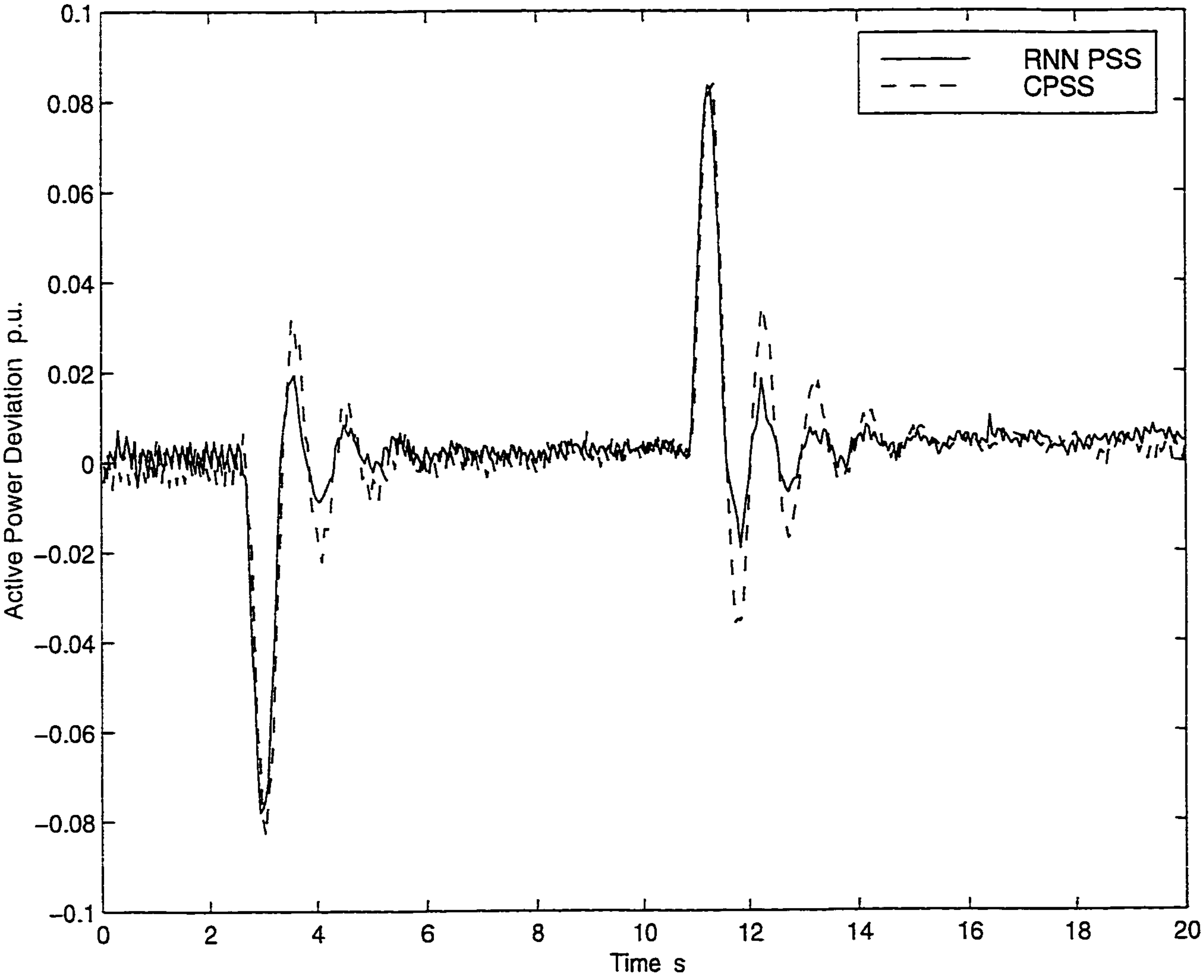


Figure 6.11. System response to a 0.10 pu step change disturbance in input torque reference, P= 0.90 *pu*, *pf* = 0.95 *lead*

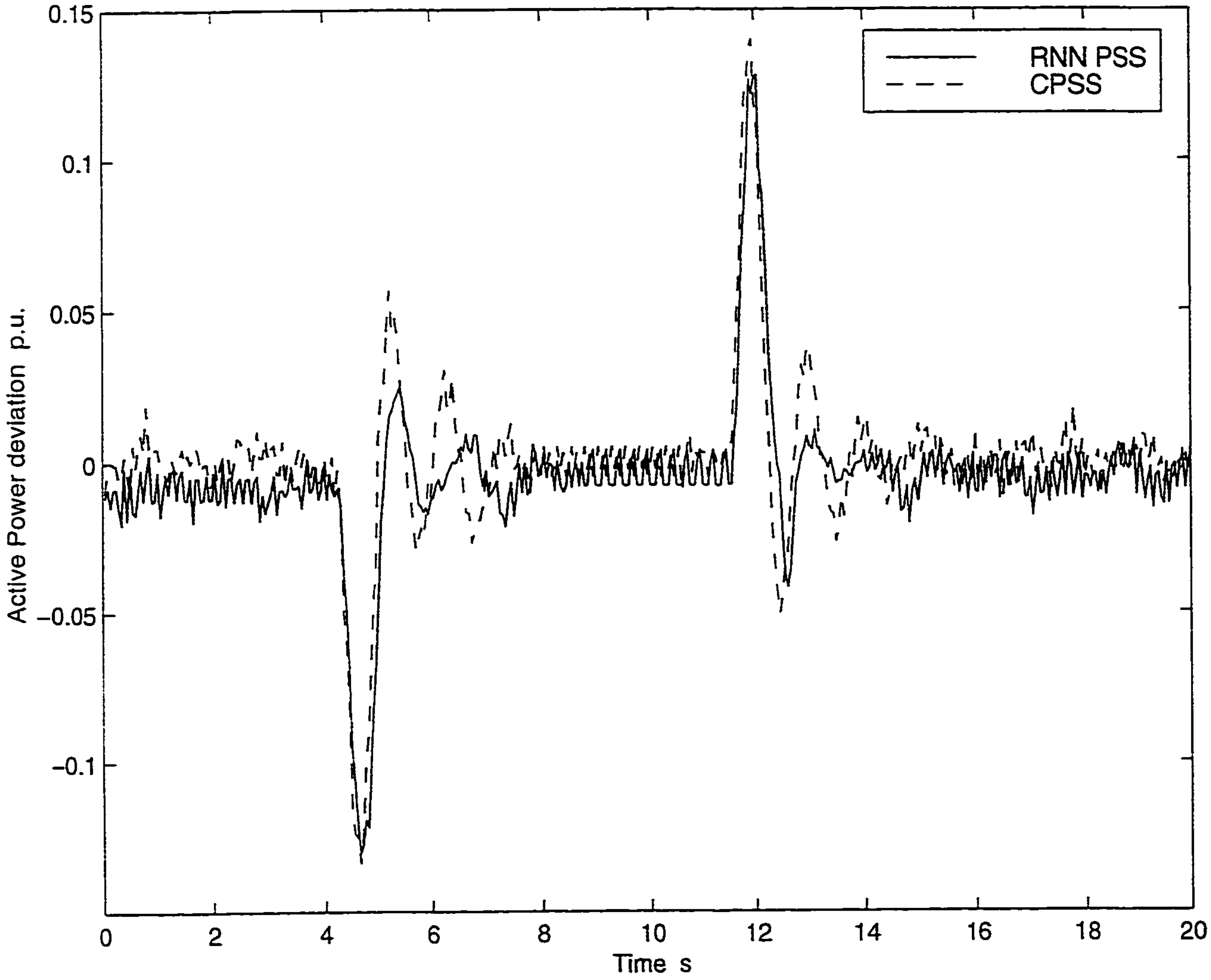


Figure 6.12. System response to a 0.20 pu step change disturbance in input torque reference, P= 0.50 *pu*, *pf* = 0.90 *lag*

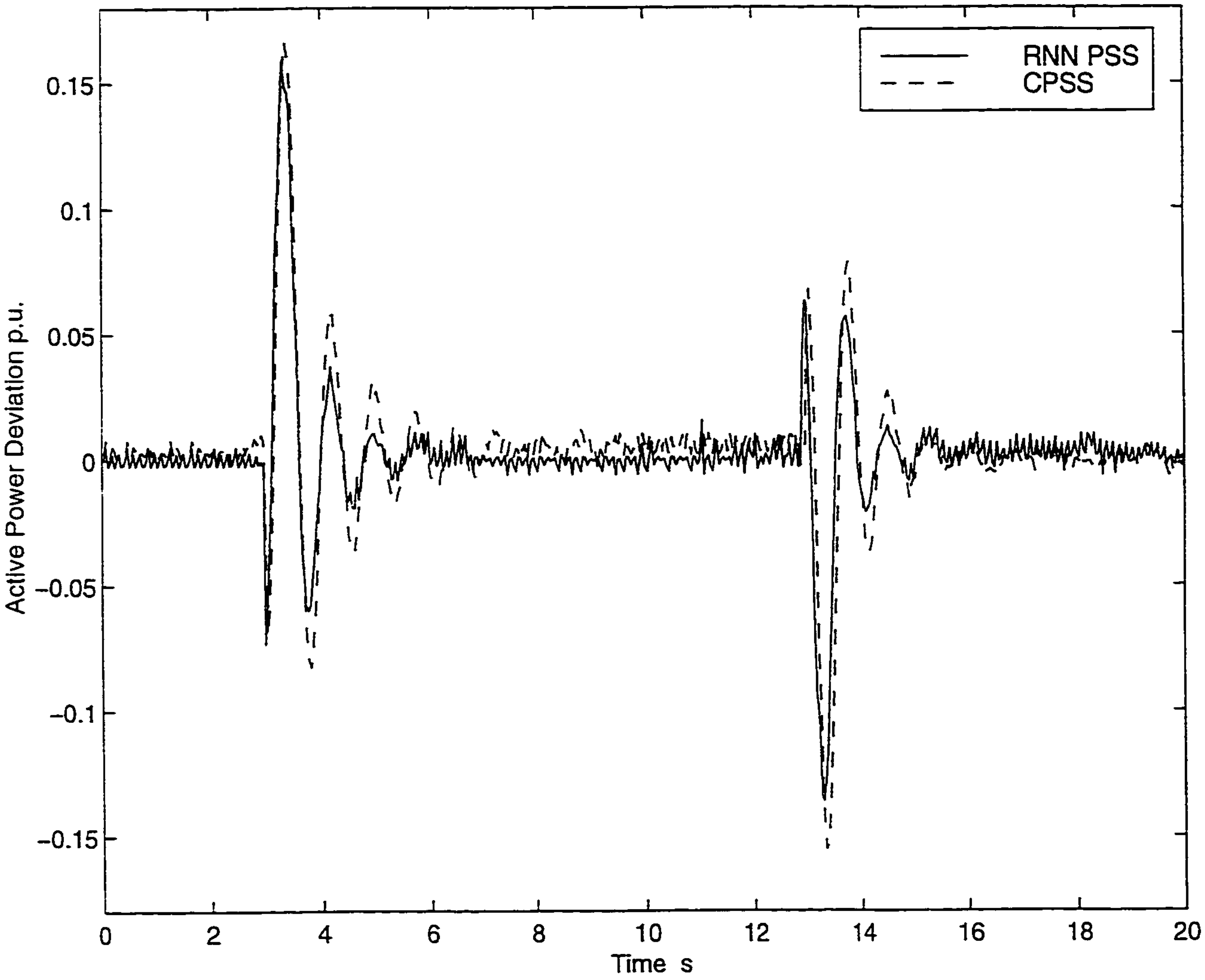


Figure 6.13. System response to a 0.25 pu step change disturbance in input torque reference, P= 0.90 *pu*, *pf* = 0.90 *lag*

600 *ms* after the line was cleared and the line was opened again 100 *ms* later. The second reclosure was successfully made at 14 *s* and the system returned to the initial operating condition.

The system responses with the RNN PSS and the CPSS under above conditions are shown in Fig. 6.14. Both RNN PSS and CPSS can maintain the system stable under the transient condition, and the first swings of both are almost the same. However, for the following swings, the RNN PSS can provide a better damping effect.

6.6.5 Dynamic Stability Test

One of the functions of a power system stabilizer is to increase the system dynamic stability margin. Usually with a properly designed PSS in operation, a system can operate at a load level at which it will be unstable without a PSS in operation.

In this test, the generating unit initially operated at a stable condition with the RNN PSS in operation. At initial stage, the operating condition was active power P = 1.10 *pu* and power factor = 0.90 *lag*. The load was then gradually increased until the active power P reached 1.22 *pu* and the load increase was stopped. At 4 *s*, the RNN PSS was replaced by CPSS. From Fig. 6.15, it can be observed that the RNN PSS can maintain the system stable at P = 1.22 *pu*. However, when CPSS is in operation, the system began to oscillate automatically and tended to lose synchronism without any external disturbance. This means that CPSS can't maintain system stable under this operating condition. When the RNN PSS was switched back at 18 *s*, the oscillations began to decrease, and the system came back to stable condition.

Figure 6.15 clearly demonstrates that the proposed RNN PSS can improve the system dynamic stability margin compared to the CPSS. Hence it is possible for the system to operate at a heavier load level.

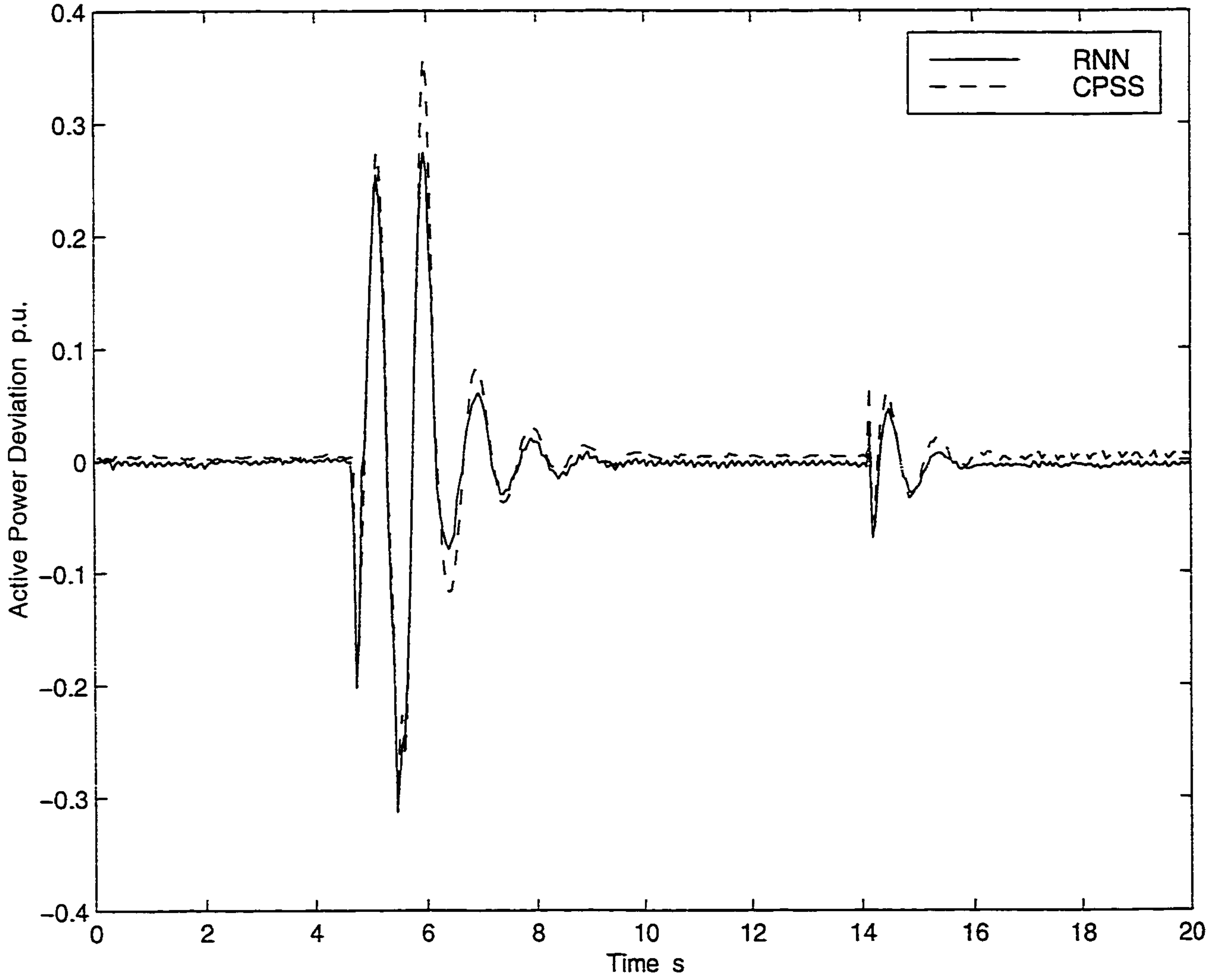


Figure 6.14. System Response with RNN PSS and CPSS for three-phase short circuit test at p = 0.90 *pu*, *pf* = 0.90 *lag*

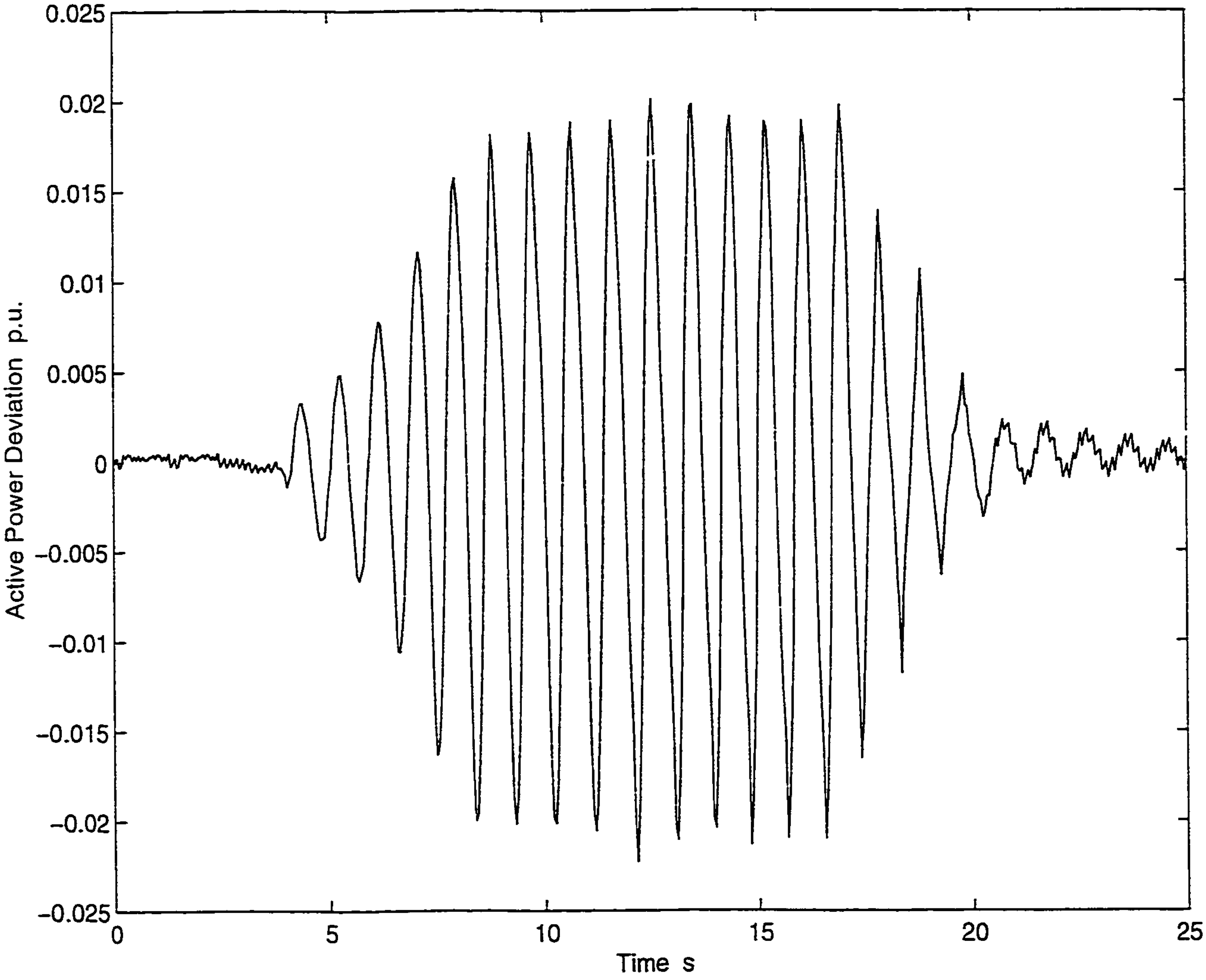


Figure 6.15. Dynamic Stability Test

6.7 Summary

Real-time implementation of the RNN PSS is described in this chapter. A simple power system is modeled by a micro-alternator driven by a DC motor and a double circuit transmission line linking the alternator to the city voltage bus. The RNN PSS is implemented in a real-time digital control environment which include a 80386 PC, TMS320C30 DSP and an ABB PHSC system.

Active power deviation and its integral are used as input signals of the RNN PSS. The RNN PSS is trained off-line and its weights later updated during application. Experimental results of the RNN PSS are compared with those of the CPSS. It is demonstrated that for the CPSS at its designed operating point, it can achieve almost the same performance as the RNN PSS. However, as the operating condition shifts to a new point, the RNN PSS gives better damping effect than the CPSS for power system oscillations. Finally, the RNN PSS also improves the dynamic stability of the system.

CHAPTER 7

CONCLUSIONS AND FUTHER STUDIES

7.1 Conclusions

With the development of large scale electric power systems and the increasing demand for power supply, how to enhance the stabilty of growing power systems has attracted the interest of many researchers. Considerable efforts have been devoted to find suitable methods to improve the stability of power systems. Power System Stabilizer (PSS) has been proved to be a very effective and economic device to improve power system stabilitty on low-frequency oscillations.

The PSS designed by classical control theory has been successfully applied in power systems. However, since it has fixed parameters based on a selected linear system model, it does not have the abilty to adapt to the variety of power system operation conditions. Therefore, it is very difficult for a conventional PSS to maintain fully satisfactory performance under various operating conditions. As the power system is a non-linear dynamic system, the stabilizer should be able to adapt itself to the varying system to provide consistent damping effect.

This dissertation is devoted to the design and developemnt of an adaptive power system stabilizer based on recurrent neural networks. Contributions have been made to the stages of RNN PSS theoretical design, simulation investigations and real-time implementation studies.

First, the basic element of neural network, neuron, is briefly described. Three learning paradigms and four learning rules are discussed. The structures of multilayer perceptron, Hopfield network, real-time recurrent neural network and Kohonen self-organizing map are also discussed. The learning algorithms related to these neural networks are reviewed. For its ability to deal with time varying input or output through its own natural temporal operation, the recurrent neural network is selected for PSS design. The real time recurrent learning algorithm is employed to train the neural network.

There are two recurrent neural networks in the proposed control system architecture; a neural identifier and a neural controller. Both neural networks are first trained off-line using data from the full working range of the generator with various disturbances. Then the weights of neural networks are updated on-line. The two neural networks operate with different time stages in a sample period. At first, the weights of the neural identifier will be updated according to the error between the output of the plant and the identifier. Then the weights of neural controller will be modified based on the error between the neural identifier output and the desired system output (which is zero in this study). At second stage, neural identifier acts as a channel to back propagate the error to the neural controller with its weights fixed. The output of the neural controller is sent to the generating unit as PSS control signal and to the neural identifier for the next step identification. The purpose of adaptive identification is to adapt the model over time to reflect any changes which occur in the dynamic characteristics of the power system. On the other hand, the design of the RNN PSS is not based on a selected operating condition of power system. All the input and output variables of the neural network are measurable for the generating unit, no internal state variables of the generating unit are required.

The behaviour of the proposed RNN PSS is investigated in the single-machine infinite bus system through computer simulation. Various operating conditions such as lightly loaded to heavily loaded, lagging power factor to leading power factor are applied. Different disturbances such as input torque step change, reference voltage step change and 3 phase to ground fault are also tested. At the same time, all the simulation cases are also studied based on a conventional PSS. Simulation results show that the proposed RNN PSS can provide satisfactory damping effect to the power system oscillations over a wide operating range, and significantly improve the system stability.

In a multi-machine power system, there are multi-mode oscillations due to the different inertia of interconnected generator units and weak connection between them. The performance of the proposed RNN PSS in a 5 machine power system is also investigated in this dissertation. The simulation results show that the RNN PSS can not only damp the specific mode of oscillation mainly related to the generating unit on which the RNN PSS is applied, but also cooperate with other RNN PSSs or CPSSs to damp the local and inter-area oscillations.

The final stage of the RNN PSS research is to implement and test the performance of RNN PSS in a real laboratory-scale model. By using ABB PHSC2 Programmable Logic Controller , a 80386 PC, a TMS320C320C30 DSP board, a real time digital control system has been set up for this purpose. With a physical model of a power system, an excellent testing environment is set up for real time study of novel PSS design. The proposed RNN PSS has been implemented in this environment and tested under various conditions. A digital CPSS is also implemented in the same environment for comparison. The experimental results not only show that RNN PSS can provide a good performance in damping power system low frequency oscillation and increase the system dynamic stability margin, but also demonstrate that proposed RNN PSS is pratical in the sense that it is feasible to install RNN PSS by using commonly available computer hardware.

Research in this dissertation demonstrates that the proposed RNN PSS has many promising features that the conventional PSS lacks. From practical view, the proposed RNN PSS is feasible since it can provide satifactory performance when implemented with ordinary hardware.

7.2 Future Studies

Applications of neural networks for power system stabilization control have been reported since early nineties. Based on the research of this dissertation, the following topics are recommanded for further research:

1. For training the neural networks, most of the algorithms applied in the PSS design are focused on the adjustment of the weights of the neural network. The structure of neural network, i.e. the number of layers and the neurons in each layer, is decided by trial and error method. However, the performance of the neural network is decided not only by its weights but also by its structure. How to apply an algorithm which can consider weight adjustment and structure modification together is an interesting topic.

2. Generally, the neural network based controller can act as a multi-input multi-output controller. The PSS is just a part of excitation control system for the generating unit. To improve the performance of the excitation control system extending ANN technique in PSS design to excitation control system design will be worth studying.

3. At present, most neural networks in PSS design use supervised learning algorithms, and the results are satisfactory. On the other hand, very little attention has been paid to the application of reinforcement learning method in PSS design. The reinforcement learning is suitable for a control system in which the most appropriate control actions may not be known. The reinforcement learning has been successfully applied in process control, it is worth looking into.

4. Other artificial intelligence (AI) techniques such as expert systems, fuzzy logic, genetic algorithms etc. are also intended to tackle highly complex, non-linear problems beyond the capability of conventional methods or too costly or time

consuming for them. ANN and these AI techniques both have their own strengths and weaknesses. In PSS design, how to combine ANN with other AI techniques through building on their strengths and offseting their weaknesses to get a better PSS will also be worth investigating.

REFERENCES

[1] William D. Stevenson, Jr.,"Elements of Power System Analysis", McGraw-Hill Book Company, 1989

[2] Y.N. Yu,"Electric Power System Dynamics", Academic Press, 1983.

[3] P.M. Anderson and A.A. Fouad, "Power System Control and Stability", IEEE Press, 1994.

[4] Graham Rogers, "Demystifying Power System Oscillations", *IEEE Computer Applications in Power*, July, 1996, pp.41-44.

[5] N.A. Vovos and G.D. Galanos, "Enhancement of the Transient Stability of Integrated AC/DC Systems Using Active and Reactive Power Modulation", *IEEE Trans. PAS*, Vol.104, 1985, pp.1696-1702.

[6] M.A. Choudhry, A.S. Emarah, K.A. Ellithy and G.D. Galanos, "Stability Analysis of a Modulated AC/DC System Using the Eigenvalue Sensitivity Approach". *IEEE Trans. PWRS* Vol.1, 1986, pp.128-137.

[7] N. Rostamkolai, A.G. Phadke, W.F. Long and J.S. Thorp, "An Adaptive Optimal Control Strategy for Dynamic Stability enhancement of AC/DC Power Systems." *IEEE Trans. Power Systems*, Vol.3, No.3, 1988, pp.1139-1145.

[8] T. Smed and G. Andersson, "Utilising HVDC to Damp Power Oscillations", *IEEE Trans. Power Delivery*, Vol.8, No.2, 1993, pp.620-625.

[9] E. Lerch, D. Povh and L. Xu, "Adavanced SVC control for Damping Power System Oscillations". *IEEE Trans. Power System*, Vol.6, No.2, 1991, pp524-531.

[10] Einar V. Larsen, Juan J. Sanchez-Gasca and Joe H. Chow, "Concepts for Design of FACTS Controllers to Damp Power Swings", *IEEE Trans. Power System*, Vol.10, No.2, 1995, pp.948-954.

[11] James.F. Grouquist, William A. Sethares, Fernando L. Alvarado and Robert H. Lasseter, "Power Oscillation Damping Control Strategies for FACTS Devices Using Locally Measurable Quantities." *IEEE Trans. Power System*, Vol.10, No.3, 1995, pp.1598-1605.

[12] P.S. Dolan, J.R. Smith and W.A. Mittelstadt, "A Study of TCSC Optimal Damping Control Parameters for Different Operating Conditions". *IEEE Trans. Power System*, Vol.10, No.4, 1995, pp.1972-1977.

[13] Narain G. Hingorani, "Flexible AC Transmission", *IEEE Spectrum*, April, 1993, pp.40-45.

[14] F.P. deMello and C. Concordia, "Concepts of Synchronous Machine Stability as affected by Excitation Control", *IEEE Trans. PAS*, Vol.88, 1969, pp.316-329.

[15] F.P. deMello and T.F. Laskowski, " Concepts of Power System Dynamic Stability", *IEEE Trans. PAS*, Vol.94, 1975, pp.827-833.

[16] M.K. El-Sherbiny and A.A. Fouad, "Digital analysis of excitation control for interconnected power systems", *IEEE Trans. PAS*, Vol.90, March/Apr. 1971, pp.441-447.

[17] 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. PAS*, Vol.92, Sept./Oct. 1973, pp.1538-1546.

[18] P.O. Bobo, J.W. Skooglund, and C.L. Wagner,"Performance of excitation system under abnormal conditions", *IEEE Trans. PAS*, Vol.87, 1968, pp.547-553.

[19] J.L. Dineley, A.J. Morris, and C. Preece, "Optimal transient stability from excitation control of synchronous generators", *IEEE Trans. PAS*, Vol.87, August, 1968, pp.1696-1705.

[20] P.C. Krause and J.N. Towel, "Synchronous machine damping by excitation control with direct and quadrature axis field windings", *IEEE Trans. PAS*, Vol.88, August, 1969, pp.1266-1274.

[21] R.T. Byerly, F.W. Keay, and J.W. Skoolund, "Damping of power oscillations in salient-pole machines with static exciters", *IEEE Trans. PAS*, Vol.89, July/August 1970, pp.1009-1021.

[22] J.P. Bayne, D.C. Lee, and W. Watson, " A power sysetm stabilizer for thermal units based on deviation of accelerating power",*IEEE Trans. PAS* 1977, pp.1777-1783.

[23] Lee, D.C., R.E. Beaulieu, J.R.R. Service, "A power system stabilizer using speed and electrical power input – design and field experience', *IEEE Trans. PAS*, 1981, pp.4151-4157.

[24] E.V. Larsen and D.A. Swann, "Applying Power System Stabilizers", *IEEE Trans. PAS*, 1981, pp.3017-3046.

[25] A. Ghosh, G. Ledwich, O.P. Malik, G.S. Hope "Power System Stabilizer Based on Adaptive Control Techniques" *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-103,No.8,August 1984 pp.1983-1989.

[26] Shi-jie Cheng, Y.S. Chow, O.P. Malik and G.S. Hope, "An Adaptive Synchronous Machine Stabilizer", *IEEE Trans. on Power Systems*, Vol.PWRS-1,No.3,August 1986 pp.387-396.

[27] D.A. Pierre,"A Perspective on Adaptive Control of Power Systems" *IEEE Trans. on Power Systems*, Vol.PWRS-2,No.2,May 1987 pp.387-396.

[28] S.J. Cheng, O.P. Malik, G.S. Hope "Damping of Multi-Model Oscillations in Power Systems Using a Dual-Rate Adaptive Stabilizer" *IEEE Trans. on Power Systems*, Vol.PWRS-3,No.1, February 1988 pp.101-107.

[29] 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.4, December 1993, pp.639-645.

[30] M.A.M Hasson, O.P. Malik and G.S. Hope,"A Fuzzy Logic Based Stabilizer for a synchronous machine", *IEEE Trans. on Energy Conversion*, Vol.6, 1991, pp.407-413.

[31] E. Handschin, W. Hoffmann, F. Reyer, T. Stephanblome, "A New Method of Excitation Control Based on Fuzzy Set Theory", *IEEE Trans. on Power Systems*, Vol.9, No.1, February 1994, pp.533- 539

[32] K.A. El-Metwally, O.P. Malik, "Fuzzy Logic Power System Stabilizer", *IEE Proc.-Gener. Transm. Distrib.*, Vol.142, No.3, May 1995, pp. 277-281.

[33] H.A. Toliyat, J. Sadeh and R. Ghazi, "Design of Augmented Fuzzy Logic Power System Stability", *IEEE Trans. on Energy Conversion*, Vol.11, 1996, pp.111-117.

[34] T. Hiyama, Y. Lleki and H. Andou, "Integrated Fuzzy Logic generator controller for stability enhancement".*IEEE Trans. on Energy Conversion*, Vol.12, No.4, Dec. 1997, pp.400-406.

[35] Y. Kitauchi and H. Taniguchi, "Experimental Verification of Fuzzy Excitation Control System for Multi-Machine Power System". *IEEE Trans. on Energy Conversion*, Vol.12, No.1, March 1997, pp.94-p99.

[36] H. Mori, "State-of-the-art Overview on Artificial Neural Networks in Power Systems", *Application of Artificial Neural Networks to Power System*, IEEE PES 96 TP 112-0, 1996.

[37] Yuan-Yih Hsu, Chao-Rong Chen,"Tuning of Power System Stabilizers Using an Artificial Neural Network", *IEEE Trans. on Energy Conversion,* Vol.6, No.4, December 1992, pp.612-617.

[38] Q.H. Wu, B.W. Hogg and G.W. Irwin, "A neural Network Regulator for Turbogenerators", *IEEE Trans. on Neural Networks*, Vol.3, No.1, Jan. 1992, pp.95-100.

[39] 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, March 1993 pp.71-77.

[40] Y. Zhang, O.P. Malik, G.S. Hope and G.P. Chen "Application of an inverse input-output ANN as a Power System Stabilizer", *IEEE Trans. on Energy Conversion,* Vol.9, No.3, 1994 pp.433-441.

[41] Y. Zhang, G.P. Chen, O.P. Malik and G.S. Hope, " A multi-input power system stabilizer based on artificial neural networks", *proc. IEEE WESCANEX 93*, 1993, pp.240-246.

[42] L. Guan, S. Cheng and R. Zhou, "Artificial neural network power system stabilizer trained with an improved BP algorithm", *IEE Proc. Generation, Transmission and Distribution*, Vol.143, No.2, March 1996, pp135-141.

[43] T. Kobayashi and A. Yokoyama, "An Adaptive Neural-Control System of Synchronous Generator for Power System Stabilization." *IEEE Trans. on Energy Conversion*, Vol.11, 1996 pp.621-627.

[44] P. Shamsollahi and O.P. Malik, "An Adaptive Power System Stabilizer Using on-line Trained Neural Networks", *IEEE Trans. on Energy Conversion,* Vol.12, No.4, Dec. 1997, pp.382-387.

[45] A. Hariri and O.P. Malik, "A Fuzzy Logic Based Power System Stabilizer with Learning Ability", *IEEE Trans. on Energy Conversion,* Vol.11, 1996, pp721-728.

[46] M. Djukanovic, M. Novicevic, D. Dobrijevic, B. Babic, Dejan J. Sobajic, and Yoh-Han Pao, "Neural-Net Based coordinated Stabilizing Control for the Exciter and Governor Loops of Low Head Hydropower Plants", *IEEE Trans. on Energy Conversion,* Vol.10, No.4, December 1995, pp760-767.

[47] A. Hariri and O.P. Malik, "Self-learning adaptive network based fuzzy logic power system stabilizer", *Proc. IEEE Int Conf. on Intelligent Systems applications to Power Systems*, Vol. 1, 1996, pp.299-303.

[48] W.S. McCullon and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, vol.9, 1943, pp.127-147.

[49] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. 1: Foundations,* MIT Press, cambridge, MA, 1986.

[50] R.P. Lippmann, "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, Vol.4, No.2, April 1987, pp4-22.

[51] Brnard Widrow and Michael A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation", *Proc. of the IEEE*, Vol.78, No.9, September 1990, pp.1415-1439.

[52] Teuvo Kohonen, "The Self-Organizing Map", *Proc. of the IEEE*, Vol.78, No.9, September 1990, pp.1465-1477.

[53] Simon Haykin, "Neural Networks, a comprehensive foundation", Macmillan College Publishing Company, 1994.

[54] Anil K. Jain, Jianchang Mao and K.M. Mohiuddin, "Artificial Neural Networks: a Tuturial" *IEEE Computer*, Vol.29, No.3, March 1996, pp31-44.

[55] Don R. Hush and Bill G. Horne, "Progress in Supervised Neural Networks–What's new since Lippmann?", *IEEE Singnal Processing Magazine*, Jan. 1993, pp.8-39

[56] C. Lee Giles, Gray M. Kuhn, and Ronald J. Williams, "Dynamic Recurrent Neural Networks: Theory and Applications." *IEEE Trans. on Neural Networks,* Vol.5, No.2, March 1994, pp.153-155

[57] Tsoi, A.C. "Locally recurrent globally feedforward networks: a critical review of architectures", *IEEE Trans. Neural Network*, No.2, 1994, pp.375-385.

[58] Fransconi, P. Gori, M. and Soda, G. "Local feedback multilayered networks",*Neural Comput.* No.4, 1992, pp.120-130

[59] Back, A. and Tosi, A.C., "FIR and IIR synapses , a new neural network architecture for time series modelling", *Neural Comput.* No.3, 1991, pp.375-385.

[60] Sreeram V.B. Aiyer, Mahesan Niranjan and Frank Fallside, "A Theoretical Investigation into the Performance of the Hopfield Model", *IEEE Trans. on Neural Networks.* Vol.1, No.2, June 1990, pp.204-215.

[61] Chinchuan Chiu, Chiayiu Maa and Michael A. Shanblatt, "Energy Function Analysis of Dynamic Programming Neural Networks", *IEEE Trans. on Neural Networks.* Vol.2, No.4, July 1991, pp.418-425.

[62] Paul J. Werbos,"Backpropagation Through Time: What It Does and How to Do It.", *Proc. of the IEEE*, Vol.78, No.10, 1990, pp.1550-1560

[63] R.J. Williams and D. Zipser, "Gradient based learning algorithms for recurrent connectionist networks", Technical Report NU-CCs-90-9, Northeastern University, 1990.

[64] R.J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks." *Neural Computation* Vol.1, 1989, pp.270-280.

[65] Kumpati S. Narendra and Kannan Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks." *IEEE Trans. on Neural Networks.* Vol.1, No.1, March 1990, pp.4-27

[66] K.J. Hunt, D. Sbarbaro, R. Zbikowski and P.J. Gawthrop, " Neural Networks for Control Systems - A Survey." Automatica, Vol.28, No.6, 1992, pp.1083-1112

[67] L. Jin, P.N. Nikiforuk, M.M. Gupta, "Direct Adaptive output tracking control using multilayered neural networks",*IEE Proc. Control Theory and Application*, Vol.140, No.6, Nov. 1993, pp393-398.

[68] George A. Rovithakis and Manolis A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks", *IEEE Trans. on System, Man and Cybernetics,* Vol.24, No.3, March 1994 pp.400-411

[69] Sanjay I. Mistry and Satish S. Nair, "Identification and Control Experiments Using Neural Designs.",*IEEE Control Systems*, June 1994, pp.48-57

[70] V. Etxebarria, "Adaptive control of discrete systems using neural networks", *IEE Proc. Control Theory and Application*, Vol. 141, No.2, March 1994, pp.204-215.

[71] G. Lightbody and G.W. Irwin "Direct neural model reference adaptive control", *IEE Proc. Control Theory and Application*, Vol.142, No.1, Jan. 1995, pp31-43.

[72] D.A. Linkers and H.O. Nyongesa "Learning Systems in intelligent control, an appraisal of fuzzy neural and genetic algorithm control application", *IEE Proc. Control Theory and Application*, Vol.143, No.4, July 1996, pp367-386.

[73] Tirthankar rayChaudhuri, Leonard G.C. Hamey, and Rodney D. Bell "From Conventional Control to Autonomous Intelligent Methods" *IEEE Control Systems*, Oct. 1996, pp78-84.

[74] Walter, J.A., and Schulten, K.J., "Implementation of self-organizing neural networks for visuo-motor control of an industrial robot",*IEEE Trans. on Neural Network*, Vol.4, No.1, 1993, pp.86-95.

[75] L. Behera, S. Chaudhury and M. Gopal, "Application of Self-organising Neural Networks in Robot tracking control". *IEE Proc. Control Theory and Application*, Vol. 145, No.2, March 1998, pp.135-140.

[76] Funahashi, K.I., and Nakamura, Y., "Approximation of dynamical systems by continuous time recurrent networks", *Neural Networks*, No.6, 1993, pp.801-806.

[77] JIN L., Nikiforuk, P.N., and Gupta, M.M. "Approximation of discrete-time state-space trajectories using dynamic recurrent neural networks", *IEEE Trans. AC*, AC-40, 1995, pp.1266-1270.

[78] Chi, S.R., R. Shoureshi and M. Tenorio, "Neural networks for system identification", *IEEE Control Systems Magazine*, No.10, 1990, pp.31-34.

[79] J.J. Hopfield, "Neural networks and physical system with emergent collective computational abilities", *Proceedings of the National Academy of Science*, Vol.79, 1982, pp.2554-2558.

[80] Karl Johan Astrom, Bjorn Wittenmark, "Adaptive Control", Addison-Wesley, 1995.

[81] Rolf Isermann, Karl-Heinz, Lachmann, Drago Matko, "Adaptive Control Systems", Prentice-Hall, 1992.

[82] W. Hahn, "Theory and Application of Liapunov's Direct Method", Prentice Hall, 1963.

[83] J. P. Lasalle and S. Lefschetz, "Stability by Liaypunov's Direct Method with Applications", Academic Press, 1961.

[84] Petros A. Ioannou and Jing Sun, "Robust Adaptive Control", Prentice Hall, 1996.

[85] Maufved Morari and Eoanghelos Zafiriou "Robust Process Control", Prentice Hall, 1989.

[86] Frank L. Lewis, Vassilis L. Syromos, "Optimal Control", J. Wiley 1995.

[87] A. P. sage, "Optimal System Control", Prentice Hall, 1968.

[88] J.R. Raol and H. Madhuranath "Neural network architectures for parameter estimation of dynamical systems". *IEE Proc. Control Theory and Application*, Vol.143. No.4, July 1996, pp387-394.

[89] M.A. Brdys, G.J. Kalauski and J.Q. Uevedo, "Recurrent Networks for non-linear Adaptive Control", *IEE Proc. Control Theory and Application*, Vol. 145, No.2, March 1998, pp.177-188.

[90] Gintaras V. Puskorius and Lee A. Feldkamp,"Neurocontrol of nonlinear dynamical systems with Kalman Filter trained recurrent networks", *IEEE Trans. on Neural networks,* Vol. 5, No.2, March 1994, pp.279-297.

[91] P.S. Sastry, G. Santharam and K.P. Unnikrishnan, "Memory Neuron Networks for Identification and Control of Dynamical Systems." *IEEE Trans. on Neural Networks,* Vol.5, No.2, March 1994, pp.306-319.

[92] Stephen W. Piche, "Steepest Descent Algorithms for Neural Network Controllers and Filters", *IEEE Trans. on Neural Networks,* Vol.5, No.2, March 1994, pp.198-212.

[93] B. Srinivasan, U.R. Prasad and N.J. Rao, "Back Propagation Through Adjoints for the Identification of Nonlinear Dynamic Systems Using Recurrent Neural Models", *IEEE Trans. on Neural Networks,* Vol.5, No.2, March 1994, pp.213-227.

[94] Jerome T. Connor, R. Douglas Martin and L.E. Altas, "Recurrent Neural Networks and Robust Time Series Prediction", *IEEE Trans. on Neural Networks,* Vol.5, No.2, March 1994, pp.240-254.

[95] Alexander G. Parlos, Kil T. Chong and Amir F. Atiya, "Application of the Recurrent Multilayer Perceptron in Modeling Complex Process Dynamics", *IEEE Trans. on Neural Networks,* Vol.5, No.2, March 1994, pp.255-266.

[96] Oercy P.C. Yip and Yoh-Han Pao, "A Recurrent Neural Net Approach to One-step Ahead Control Problems." *IEEE Trans. on System, Man and Cybernetics,* Vol.24, No.4, April 1994 pp.678-683.

[97] J. He and O.P. Malik, "An Adaptive Power System Stabilizer based on Recurrent Neural Networks", *IEEE Trans. on Energy Conversion,* Vol.12, No.4, Dec. 1997, pp.413-418.

[98] Hong-Chan Chang, Mang-Hui Wang, "Neural-Network Based Self-Organizing Fuzzy Controller for Transient Stability of Multi-machine Power System". *IEEE Trans. on Energy Conversion*, Vol.10, 1995, pp.339-347

[99] G.P. Chen and O.P. Malik "Tracking constrained adaptive power system stabilizer", *IEE Proc.- Gener. Transm. Distrib.*, Vol. 142, No.2, March 1995, pp.149-155.

[100] Y. Zhang, O.P. Malik and G.P. Chen "Artificial Neural Network Power System Stabilizers in multi-machine power system environment", em IEEE Trans. on EC, Vol. 10, No.1, March 1995, pp.147-153.

[101] Yoshihiro Kitauchi and Haruhito Taniguchi "Experimental verification of fuzzy excitation control system for multimachine power system" *IEEE summer meeting*, 1996.

[102] Young-Moon Park, Myeon-Song Choi, and Kwang Y.Lee, "A Neural Network-Based Power System Stabilizer using Power Flow Characteristics", *IEEE Trans. on Energy Conversion*, Vol.11, No.2, June 1996, pp.435-441.

[103] Lim Choo Min and Li Qing, "An enhanced adaptive neural network control scheme for power systems", *IEEE Trans. on Energy Conversion*, Vol.12, No.2, June 1997, pp.166-172.

[104] K. Jagannathan, O.P. Malik, and G.S. Hope, "Excitation control of synchronous generators using adaptive regulators – Part II: implementation and test results", *IEEE Trans. on Power Apparatus and Systems*, Vol. PAS-103, May 1984, pp.904-910.

[105] E. Vaahedi, A.D. Noble, and D.C. Macdonald, "Generator on-line estimation and optimal control: laboratory implementation and prospects for development", *IEEE Trans. on EC*, Vol. EC-1, Sept. 1986, pp.55-60.

[106] K.S. Prakash, O.P. Malik, G.S. Hope, G.C. Hancock, and K.K. Wong, "Laboratory investigation of an amplitude comparator based directional comparison digital protection scheme", *IEEE Trans. on Power Delivery*, vol.5, 1990, pp.1687-1694.

[107] M.A.M. Hassan, O.P. Malik, "Implementation and Laboratory test results for a fuzzy logic based self-tuned power system stabilizer", *IEEE Trans. on EC*, Vol-8, No. 2, June 1993, pp.221-227.

[108] S. Roy, O.P. Malik and G.S. Hope, "Real time test results with adaptive speed controllers for a diesel prime-mover", *IEEE Trans. on EC*, Vol-8, No. 3, Sept. 1993, pp.499-505.

[109] Tanzo Nitta, Takao Okada, Yasuyuki Shirai, Takuya Kishida and Yoshihiro Ogawa, "Experimental Studies on Power System Stability of a Superconducting Generator with High Response Excitation", *IEEE Summer Meeting* 1996.

[110] Y. Zhang and O.P. Malik, "Experimental studies with a neural network based power system stabilizer", ISAP 96, pp104-pp108.

[111] 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.*

[112] "TMS320C30 System Board User's Manual, TMS320C30 Version 1.10", Spectrum Signal Processing Inc., 1990.

[113] "SPOX Application Programming Guide, TMS320C30 Version 1.10", Spectrum Signal Processing Inc., 1990.

[114] "TMS320C3x User's Guide",Texas Instruments, 1990.

[115] "TMS320C3x C Source Debugger User's Guide", Texas Instruments, 1991.

[116] "TMS320Floating-Point DSP Optimizing C Compiler User's Guide", Texas Instruments, 1995.

[117] 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.

APPENDIX A

SINGLE-MACHINE POWER SYSTEM

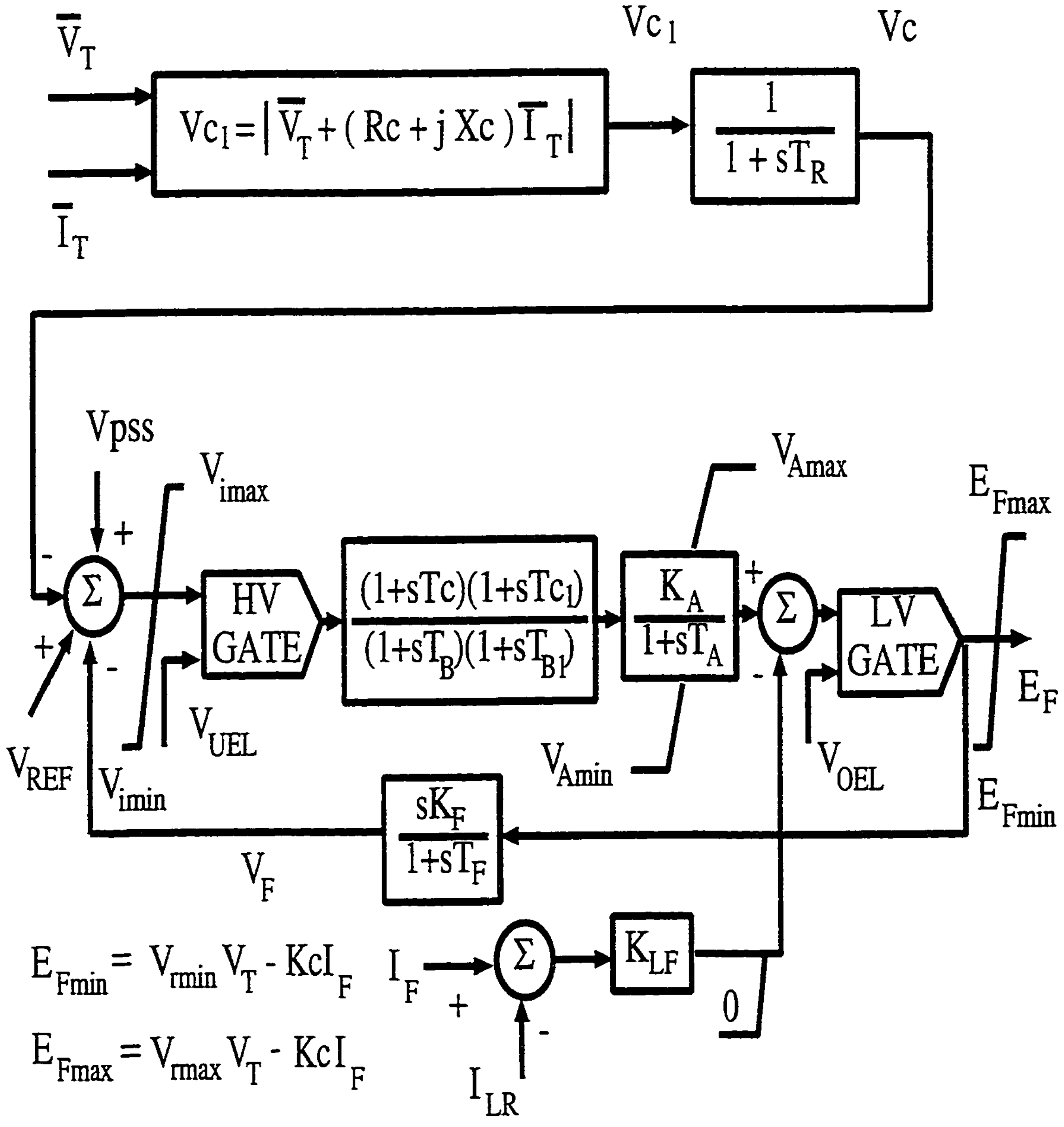
1. Generator

$$\begin{aligned}
\dot{\delta} &= \omega_0\omega \\
\dot{\omega} &= \frac{\omega_0}{2H}(T_m + g + K_d\dot{\delta} - T_e) \\
\dot{\lambda}_d &= v_d + r_a i_d + \omega_o(\omega + 1)\lambda_q \\
\dot{\lambda}_q &= v_q + r_a i_q - \omega_o(\omega + 1)\lambda_d \\
\dot{\lambda}_f &= e_f - r_f i_f \\
\dot{\lambda}_{kd} &= -r_{kd} i_{kd} \\
\dot{\lambda}_{kq} &= -r_{kq} i_{kq} \\
\lambda_d &= (L_{md} + l_a) i_d + L_{md} i_{kd} + L_{md} i_f \\
\lambda_{kd} &= L_{md} i_d + L_{kd} i_{kd} + L_{md} i_f \\
\lambda_f &= L_{md} i_d + L_{md} i_{kd} + L_f i_f \\
\lambda_q &= (L_{mq} + l_a) i_d + L_{mq} i_{kq} \\
\lambda_{kq} &= L_{mq} i_q + L_{kq} i_{kq}
\end{aligned}$$

2. Transmission network

$$\begin{aligned}
v_d &= v_b \sin\delta + r_e i_d - x_e i_q \\
v_q &= v_b \cos\delta + r_e i_q + x_e i_d
\end{aligned}$$

3. IEEE standard type ST1A AVR and exciter model, Fig.A.1.



$$E_{Fmin} = V_{rmin} V_T - KcI_F$$

$$E_{Fmax} = V_{rmax} V_T - KcI_F$$

Figure A.1. Schematic diagram of AVR and exciter model

4. Governor transfer function

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

5. IEEE standard PSS1A type conventional PSS, Fig.A.2.

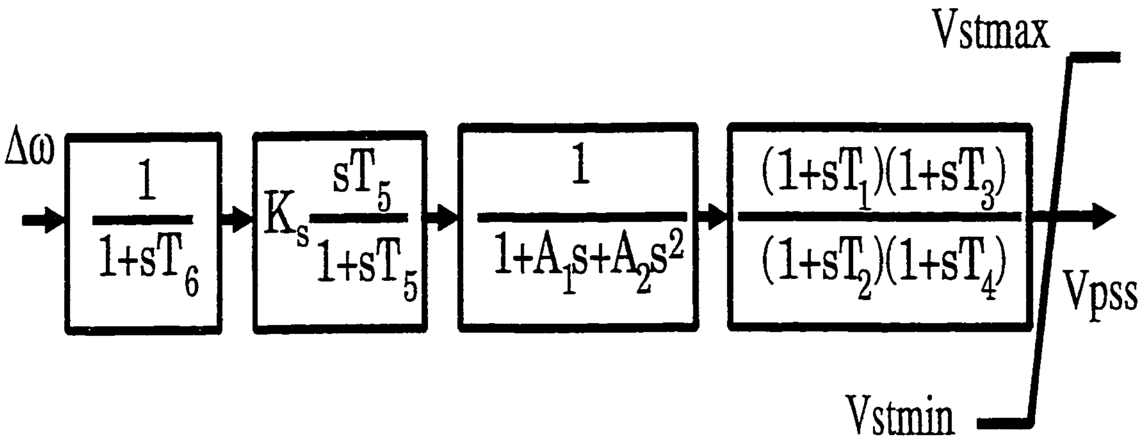


Figure A.2. Schematic diagram of PSS1A type CPSS

6. Parameters used in the simulation studies

$r_a = 0.007$	$r_f = 0.00089$	$r_{kq} = 0.023$
$r_{kd} = 0.023$	$x_q = 0.743$	$x_d = 1.24$
$x_{md} = 1.126$	$x_{mq} = 0.626$	$x_f = 1.33$
$x_{kd} = 1.1500$	$x_{kq} = 0.625$	$H = 6.0$
$K_d = 0.0$	$r_t = 0.05$	$x_t = 0.3$
$R_C = 0.0$	$X_C = 0.0$	$K_C = 0.08$
$T_C = 1.0$	$T_B = 10.0$	$T_{C1} = 0.0$

$T_{B1} = 0.0$	$T_A = 0.0$	$K_A = 190.0$
$T_F = 1.0$	$K_F = 0.05$	$T_R = 0.04$
$V_{IMIN} = -999$	$V_{IMAX} = 999$	$V_{AMAX} = 999$
$V_{AMIN} = -999$	$V_{RMAX} = 999$	$V_{RMIN} = -999$
$V_{UEL} = -999$	$V_{OEL} = 999$	
$a = -0.00133$	$b = -0.17$	
$T_1 = 0.25$	$T_2 = 0.08$	$T_3 = 0.25$
$T_4 = 0.08$	$T_5 = 4.85$	$T_6 = 0.005$
$A_1 = 0.0$	$A_2 = 0.0$	$K_s = 0.12$
$V_{STMIN} = -0.1$	$V_{STMAX} = 0.1$	

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

APPENDIX B

MULTI-MACHINE POWER SYSTEM

1. Generator model

$$
\begin{aligned}
\dot{\delta} &= \omega_0 \omega \\
\dot{\omega} &= \frac{\omega_0}{2H}(T_m + g + K_d \dot{\delta} - T_e) \\
T'_{do} \dot{e'}_q &= e_f - (x_d - x'_d) i_d - e'_q \\
T''_{do} \dot{e''}_q &= [e'_q - (x'_d - x''_d) i_d - e''_q] + T''_{do} \dot{e'}_q \\
T''_{qo} \dot{e''}_d &= (x_q - x''_q) i_q - e''_d
\end{aligned}
$$

2. Generator parameters

	G_1	G_2	G_3	G_4	G_5
X_d	0.1026	0.1026	1.0260	1.0260	1.0260
X_q	0.0658	0.0658	0.6580	0.0658	0.6580
X'_d	0.0339	0.0339	0.3390	0.0339	0.3390
X''_d	0.0269	0.0269	0.2690	0.0269	0.2690
X''_q	0.0335	0.0335	0.3350	0.0335	0.3350
T'_{do}	0.3670	0.3670	0.3670	0.3670	0.3670
T''_{do}	0.0314	0.0314	0.0314	0.0314	0.0314
T''_{qo}	0.0623	0.0623	0.0623	0.0623	0.0623
H	20.000	20.000	3.0000	20.000	3.0000

3. Parameters of AVR and simplified ST1A exciters

	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

The output of all exciters is limited within -6.7 to 7.8 *p.u.*

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	-0.00133	-0.00133	-0.00133	-0.00133	-0.00133
b	-0.0150	-0.0150	-0.0150	-0.0150	-0.0150

5. Transmission line parameters

$BusNo.$	R_t	X_t	$B_t/2$
$1-7$	0.00435	0.01067	0.01536
$2-6$	0.00213	0.00468	0.0404
$3-6$	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

6. Operating condition #1:

	G_1	G_2	G_3	G_4	G_5
$P(pu)$	5.1076	8.5835	0.8055	8.5670	0.8501
$Q(pu)$	6.8019	4.3836	0.4353	4.6686	0.2264
$V(pu)$	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 *pu*:

$L_1 = 7.5 - j5.0$ $L_2 = 8.5 - j5.0$ $L_3 = 7.0 - j4.5$

7. Operating condition #2:

	G_1	G_2	G_3	G_4	G_5
$P(pu)$	3.1558	3.8835	0.4055	4.0670	0.4501
$Q(pu)$	2.9260	1.4638	0.4331	2.1905	0.2574
$V(pu)$	1.0500	1.0300	1.0250	1.0500	1.0250
$\delta(rad)$	0.0000	0.1051	0.0943	0.0361	0.0907

Loads in admittances in *pu*:

$L_1 = 3.755 - j2.5$ $L_2 = 4.25 - j2.5$ $L_3 = 3.5 - j2.25$

APPENDIX C

PHYSICAL MODEL POWER SYSTEM

1. The parameters of micro-alternator:

$$\begin{array}{llll} x_d = 1.2 & x_q = 1.2 & r_d = 0.0026 & r_q = 0.0026 \\ x_{md} = 1.129 & x_{mq} = 1.129 & x_{kd} = 1.25 & x_{kq} = 1.25 \\ r_{kd} = 0.0083 & r_{kq} = 0.0083 & x_f = 1.27 & r_f = 0.000747 \\ H = 4.75 & & & \end{array}$$

2. The parameters of each π-section of transmission line:

   R = 0.036 X = 0.0706 B = 18.779

3. The parameter of the CPSS

   K_s=-0.5 T_1=T_3=0.1 T_2=T_4=0.08