# ACTIVE SECURITY CONSTRAINED OPTIMAL POWER FLOW USING MODIFIED HOPFIELD NEURAL NETWORK

by

Rukmi Sari Hartati

Submitted

in partial fulfilment of the requirements

for the degree of

**DOCTOR OF PHILOSOPHY**

Major Subject : Electrical and Computer Engineering

at

**DALHOUSIE UNIVERSITY**

Halifax, Nova Scotia October, 2001

0-612-77593-3

## Dalhousie University

## Faculty of Engineering

The undersigned hereby certify that they have examined, and recommend to the Faculty of Graduate Studies for acceptance, the thesis entitles "Active Security Constrained Optimal Power Flow Using Modified Neural Network" by Rukmi Sari Hartati in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dated: December 10/01 [redacted]

Supervisor: [redacted]

Dr. M. E. El-Hawary

External Supervisor: [redacted]

Dr. Magdy Salama

University of Waterloo

Examiners: [redacted]

[redacted] Dr. T. A. Little

[redacted]

Dr. M. Rahman

**Dalhousie University**

**Faculty of Engineering**

DATE : November 13, 2001

| | |
|---|---|
| **Author :** | Rukmi Sari Hartati |
| **Title :** | Active Security Constrained Optimal Power Flow Using Modified Hopfield Neural Network |
| **Major Subject :** | Electrical and Computer Engineering / Power System |
| **Degree :** | Doctor of Philosophy |
| **Convocation :** | May, 2002 |

# Table of Contents

## List of Tables

## List of Figures

## List of Symbols

$C_T$ = Total operating cost of the system

$C_i$ = cost function of generating plant at bus $i$

NG = number of generating units

$a_i$ = the fixed fuel cost at generator bus $i$

$b_i$ = the variable fuel cost in proportion to active power at generator bus $i$

$c_i$ = the variable fuel cost in proportion to second order term of active power at generator bus $i$

$P_{Gi}$ = the generation of $i^{th}$ plant

$P_D$ = the total load demand

$P_L$ = transmission power losses

$[J]$ = Jacobian matrix

$P_{Gi\ \min}$ = The minimum active power generation limit of unit $i$

$P_{Gi\ \max}$ = The maximum active power generation limit of unit $i$

$Q_{Gi\ \min}$ = The minimum reactive power generation limit of unit $i$

$Q_{Gi\ \max}$ = The maximum reactive power generation limit of unit $i$

$P_k$ = the active power generation at bus k

$Q_k$ = the reactive power generation at bus k

$V_k$ = voltage at bus k

$Y_{km}$ = the element $km$ of admittance matrix of the transmission network

$G_{km}$ = real elements of the nodal admittance matrix between bus k and bus m

$B_{km}$ = imaginary elements of the nodal admittance matrix between bus k and bus m

$\delta_k$ = voltage angle at bus k

$\theta_{km}$ = phase angle of $Y_{km}$

$\Delta P_{PV}$ = changes in active powers at PV buses

$\Delta P_{PQ}$ = changes in active powers at load buses

$\Delta P_S$ = changes in active powers at slack buses

$\Delta Q_{PV}$ = changes in reactive powers at PV buses

$\Delta Q_{PQ}$ = changes in reactive powers at load buses

$\Delta P_{S,PV}$ = column vector of the active power changes at swing bus and PV buses

$\Delta \delta_{PV}$ = changes in phase angles at PV buses

$\Delta \delta_{PQ}$ = changes in phase angles at PQ buses

$\Delta V_{PV}$ = changes in magnitude voltages at PV buses

$\Delta V_{PQ}$ = changes in magnitude voltages at PQ buses

min = minimum permissible limits

max = maximum permissible limits

$u_i$ = the total input to neuron i.

$W_{ij}$ = the synaptic interconnection strength from neuron j to neuron i.

$I_i$ = the external input to neuron i.

$v_j$ = the output of neuron j.

$e_j(n)$ = the error signal at the output of neuron j at iteration $n$

$E(n)$ = the instantaneous error energy

$E_{av}$ = the average squared error energy

$\hat{Y}(n)$ = the actual output of the neurons

$\delta_k$ = the error signal at an output unit k

$\eta$ = learning rate parameter

$\alpha$ = constant which determines the effect of past weights changes on the current direction of movement in weight space.

## ACKNOWLEDGEMENT

All gratitude is due to Allah Almighty who guided and aided me to get this work done. Only from him do we receive our knowledge.

I would like to express my thanks and gratitude to my supervisor, Professor Dr. M.E.El-Hawary, Professor in Electrical & Computer Engineering at Dalhousie University, for his excellent supervision and encouragement throughout the course of this project. His advice and contribution in the form of financial assistance in the last phase of this project is highly appreciated.

I wish also to express my sincere thanks to Dr. T.A. Little for their kind help and valuable suggestions. I thank also my friends Ahmed Al-Gallad, Said Mekhamer, and El-Bahlul Fgee who helped in many different ways in bringing this project to a successful conclusion.

Special thanks are due to EEDP project and WUSC for providing financial assistance in form of a scholarship. I will always be grateful to all the case and concern they showed throughout the years of my studies.

Very special thanks go out to my husband, son and daughter, and to my parents for their patient, understanding and moral support, and for the many sacrifices they made on my behalf.

# Abstract


A Hopfield neural network is modified to handle inequality constraints by introducing an exterior penalty function. The use of a penalty function converts constrained optimization problems into unconstrained problems.

The optimal power flow is a general non-linear programming problem with a non-linear objective function and non-linear functional equality and inequality constraints. Security Constrained Dispatch is defined as an Optimal Power Flow problem, in which the objective function is the total cost of generations and the security constraints are placed on the bus voltage magnitudes, phase angles and the generated active powers.

This thesis presents an alternative method for solving optimal active power flow and active security-constrained dispatch using a modified Hopfield neural network. The objective function of security-constrained dispatch is the incremental generation cost function in quadratic form which is expanded in a second-order Taylor series. The equality and inequality constraints are modelled using a linearized network and appended to the objective function using suitable penalty functions to form an augmented cost function

The goal of this research is to model and study the applicability of the modified Hopfield Neural Network for solving optimal active power flow and security-constrained dispatch problem. In addition, this thesis aims to discover the advantages and disadvantages of using this technique instead of the methods that currently exist.

The Hopfield Neural Network was simulated on a digital computer for four standard IEEE test systems varying in size from a 5-bus system to a 57-bus system. The optimal solution obtained using this approach is consistent with the solution obtained using the conventional method.

The advantage of this method is in the ease of formalization of the problem. It is simple, straightforward, and easy to apply. The method requires modest memory resources and is efficient in computation time. This representation is applicable to many problems other than the economic load-dispatching problem.

# Chapter 1

# INTRODUCTION

A significant requirement of any modern society is the economic secure operation of its electric power system. It is a challenging task for the electric power systems engineer to determine optimum economic dispatch for this complex system. Economic Dispatch means allocating the demand among available generating resources while meeting network constraints and minimizing the cost of operations. In addition, it is important to consider other objectives such as minimizing the environmental impact of the operation. Other major considerations needed in the operational plan are continuity of service, reliability, and safety for both personnel and equipment.

The optimum operation of a power system will depend upon the restrictions imposed by factors other than operating economics. This is due to the fact that the consumption of electric energy has grown tremendously, and on the other hand, utilities have grown in size to meet the demand to a point where significant savings in operating costs can be achieved with even a fraction of percent improvements in operating efficiency such as 0.5%. Thus, it is vital to encourage any efforts by electric utilities to operate as efficiently as possible. In addition, the ability to solve the optimal power flow problem can be extremely useful for the planning and design of future equipment additions to power systems.

### 1.1. Economic Dispatch Problem

Economic dispatch, as an approach to optimize active power flow, was first proposed in early 1920s when two or more units were committed to take load on a power system whose total capacities exceeded the required load. During these years, the operator was confronted by the problem of how to divide the active load between the two units, within which the total load is served and the total cost is minimized. Specifically, economic dispatch is a computational process whereby the total active generation required is allocated among the generating units available so that the constraints imposed are satisfied and the energy requirements are minimized.

Conventional economic dispatch includes the active power balance equation which uses the loss formula in order to obtain transmission losses. Yet, this formulation does not account for network security constraints, such as the generator voltage magnitude or phase angle difference.

In the late 1950s the power flow problem formulation made its first appearance. This problem is characterized by inputs concerning the network under study, and injected positive and negative active (P) and reactive (Q) powers at all the busses of the network. The objective of power flow is to determine the voltages and angles at all busses of the network from which all other quantities can be calculated.

Optimization later emerged as a requirement and was applied to power flows during the 1960s. An optimal power flow (OPF) is intended to find a power flow solution which optimizes a performance function such as fuel costs, or network losses, while at the same time enforcing the loading limits imposed by system equipment. A power flow solution is then obtained that is both feasible and has a minimum value of the objective function.

When the total fuel cost is minimized, the optimal power flow results in an appropriate economic dispatch. In addition, to determining the active power output of generators and phase angles, the optimal power flow program also determines the reactive power output of generators and other VAR sources as well as transformer tap settings. The optimal power flow problem involves optimization of static operating conditions of an electric power system by computing optimal schedules for the controllable variables in the system. These variables are the real power (P) and reactive power (Q) injected into each node, the magnitude of voltage (V) at each node, and the phase angle ($\delta$) of the voltage. The main purpose of an OPF study is to schedule the power system controls in order to achieve operation at the desired security level [1] while, at the same time, optimizing (minimizing) such scalar objective functions as the cost of operation, the transmission losses, the reactive power, etc.

The classic OPF problem is usually subdivided into two sub problems, namely the real power and the reactive power optimization tasks. Current practice executes OPF on-line and in such a manner as to observe security-constraints. This implies that the simulation has to be on-line, and provisions should be made for the power system operator to communicate interactively with the computer. Strict reliability requirements demand that the security-constrained scheduling calculations, including results of OPF simulations, are initiated, completed, and dispatched automatically.

One of the central components of an OPF problem solution is how to formulate the objective function. The objective function may be one or more of the operating costs, system transmission losses, or reactive power. It is often difficult to describe the best operating state of a power system by a single scalar function. Wood and Wollenberg [1] describe several techniques for setting up suitable objective functions.

### 1.2. Security-Constrained Dispatch

The Optimal Power Flow where the objective function is total cost of generation, and the constraints on bus voltage magnitudes, phase angles, and generated reactive power are considered, called a Security Constrained Dispatch program.

Several techniques for solving power optimization problems have been developed. Lee et al. [27], for instance, presented a technique for optimal active and reactive power scheduling by suggesting the use of the gradient projection method. The method is based upon three modules coupled to each other. First, the P optimization module, and the second is Q optimization module. The objective function is the total power production cost. In the P-optimization module, the reactive power Q (swing bus, generator bus and capacitor bus) behaves as a dependent variable, while, in the Q-optimization, P (swing bus and generator bus) is a dependent variable. Finally, the load-flow module is used to make fine adjustments of the results on P- and Q-optimization modules. This optimization technique is performed by uniting the two decoupled optimization problems into one framework. By doing this, the switching of objective functions from one to another can be avoided. But, it is important to note that the OPF problem has been attempted sequentially, rather than simultaneously.

The security-constrained dispatch problem developed by Salgado *et al.* [12] is also somewhat similar to Lee's. The technique involves the optimization of weighted reactive power injections as a second objective function in the reactive power dispatch. The formulation of the objective function for the optimization problem is also in order to minimize the cost function. In order to have it in terms of the same decision variables as the constraints, a modified objective function is used. The quadratic functions are expanded in a second-order Taylor series such that the resulting objective function is an incremental cost function [12, 27, 46]. In this Security Constrained Dispatch, the dependent variable constraints are on the bus voltage magnitude and the generated reactive powers [12]. Sjoholm and Boye [46], propose a new method that modifies the constraints, i.e., by replacing the bus voltage magnitude and generated reactive power constraints with the constraints on the bus voltage angles. The objective function is the same as that developed by Lee [27] and Salgado [12].

### 1.3. Hopfield Neural Network

Over the past few years, a number of approaches using Artificial Neural Networks have been proposed as alternative methods in power system optimal operation. In general, the neural network methodology should be applied in areas where conventional techniques have not achieved the desired speed and accuracy. There are several types of neural networks used in various applications. Among them are the layered perceptron, the Kohonen and the Hopfield neural networks. The layered perceptron is trained using supervised learning. The perceptron receives the desired output of each input pattern. Multi-layer perceptrons can be used for both classification and system identification. In classification, layered perceptrons are known to have superior generalization and noise rejection capability, which makes them well suited for the task. In system identification, the ability to accurately interpolate on a single or multi-dimensional output surface is the key to a successful identification model.

The Kohonen network, on the other hand, uses unsupervised learning that does not require knowledge of the output. A Kohonen-based classifier is a powerful tool for system classification [2]. One key advantage is that the net can classify the patterns during training without explicit knowledge of its class (status).

Hopfield Neural Networks have also been proposed for solving several combinatorial search application problems. Since Hopfield applied the artificial neural network to travelling salesman problems, its application to optimization problems has been studied. This method can be applied to optimization problems such as dynamic economic load dispatching by replacing the spontaneous reduction of energy with the minimization of objective function. Since a parallel calculation by hardware can be considered, its application to on-line operations of power systems, which requires high-speed calculation, can be expected. In problems of optimization, the Hopfield Neural Network has a well-demonstrated capability of finding solutions to difficult optimization problems [2].

Solving optimization problems requires minimization of some cost functions that are subject to a set of constraints. These cost functions are known in the neural network literature as energy functions, and the Neural Network can produce good solutions by minimizing the energy function.

Recently, many researchers have studied and developed new simulation techniques to solve the OPF problem by using artificial Neural Networks of the Hopfield type and its extended version in the Lin-Kennedy type. This Hopfield/Lin-Kennedy network type has been applied to optimal power flow and economic load dispatch problems [11,16,17,18,19,20,21]. Kasangaki *et al.* [11] presented a Hopfield artificial Neural Network (ANN) for solving a constrained OPF problem. The objective function involves minimizing the system transmission losses. To form an augmented cost function, they introduced an extended interior penalty function. Park et al. [14] proposed solving the economic load dispatch for piecewise quadratic cost functions using the Hopfield Neural Network. They applied the method to the three generator units problem and the results of this method were compared successfully with those of the numerical method in an hierarchical approach. King et al. [19] introduced an improved Hopfield for the economic environmental dispatching of electric power system problems. Their method has been compared with the Newton-Raphson algorithm for a 12-generator test system and they show that the execution time was approximately the same. Gee et al [15] improved the mapping process and provided a computational method for obtaining the weights and biases for the Hopfield networks in order to solve quadratic problems with linear equality and inequality constraints. Abe et al. [13] introduced a slack variable to convert inequality constraints into equality constraints. Yalcinoz et al [18] presented an improved Hopfield Neural Network which modified Gee and Prager's (GP) method in order to solve Economic Dispatch with transmission capacity constraints. Constraints are handled using a combination of the GP model and the model of Abe et al. [13]. This method has achieved efficient and accurate solutions for two-area power systems with 3, 4, 40 and 120 units. Yalcinoz et al. [47] again proposed an improvement on the Hopfield Neural Network approach [17, 18], for solving real time economic dispatch problem with security constraints. They modified the activation function for which a symmetric ramp function is chosen for the input-output function and applied to each element of the variable set. They tested the network on the IEEE 30-bus system for different demands. Gosh and Chowdhury [48] used the Hopfield Neural Network to solve the security constrained optimal rescheduling. They combined the minimum deviations in real power generations and loads at buses to form the objective function for optimization. The inequality constraints are on active line flow limits and equality constraints are on real power generation load balance. Transmission losses are also taken into account in the constraint function.

This thesis presents an extended Hopfield model to handle inequality constraints by using the exterior penalty function method [4, 30, 31], and then applies the algorithm to determine the weights in the energy function. This model is used for solving the optimal active power-flow where the objective function is an incremental cost function.

### 1.4 Objective and Scope of the Thesis

Since the original Hopfield model cannot handle the inequality constraints, Abe et al. [13] introduced a slack variable in order to convert inequality constraints into equality constraints. The difficulties arise in large scale power systems, which include a large number of inequality constraints, requiring a large number of slack variable. In this thesis, another way of handling the inequality constraints by introducing exterior penalty function method is presented. This avoids difficulties associated with using many additional slack variables. Chapter 3 gives a more detailed description of the Abe's method.

The technique mentioned above along with the Hopfield Neural Network are to be tested via practical applications in power system optimization. Optimal power flow and security-constrained dispatch problem are good examples of highly constrained non-linear problems. Therefore, the objective of this research is to model and study the applicability of the modified Hopfield Neural Network on solving optimal active power flow and security-constrained dispatch problem. In addition, this thesis aims to discover the advantages and disadvantages of using this technique to replace existing methods. The modification introduced is compared with some conventional method (gradient projection & Lagrangian multiplier methods) and with techniques that already use the Hopfield neural network but adopt different problem formulations.

In this thesis, the objective function is the same as used in references [12, 27, 46]. The constraints used by Sjoholm and Boye [46] are modified by adding constraints on bus voltage magnitudes, to avoid the low voltages that may result from using the decoupled power flow method as mentioned in Chapter 2.

### 1.5 Thesis Outline

This thesis consists of six chapters. Chapter 1 is an introduction to the main ideas of the thesis providing a background of economic power dispatch, optimal power flow, and the Neural Network used. The objectives, the organization of the thesis and the work covered are also outlined in this chapter.

Chapter 2 covers the description of the optimum operation problem. Within this chapter, some discussions on dispatch, dispatch with losses, decoupled power flow, and full power flow are presented.

Chapter 3 describes the concept of Artificial Neural Networks, in which two significant approaches for optimization of power systems are discussed.

Chapter 4 shows the application of Hopfield model for solving the Economic Power Dispatch problem. The discussion includes background and formulation. Results will be given in detail according to the output from different networks that are implemented in the model.

Chapter 5 discusses active security dispatch using the Hopfield network. Some background information on this topic will be covered, followed by the formulation used for this study. Results obtained using the formulation will be presented as the conclusion section of this chapter.

Chapter 6 covers the summary and conclusions drawn from the discussions presented in the previous chapters. Several recommendations concerning future extensions of the present work are made. A list of references used in this thesis will be presented after chapter 6 and followed by several appendices.

# Chapter 2
# OPTIMUM OPERATION PROBLEM

## 2.1 Introduction

The solution to the optimal operation problem of power system is required to assist in the optimal planning of facilities or devices for the system. In general, these facilities consist of generating plants, transformers, reactive-power compensation devices and transmission networks. Since Dommel and Tinney [40] introduced the optimal power-flow method for the first time, many articles have appeared in the literature on this subject [10, 11, 12, 22, 35, 36].

Generally, an optimal power flow problem deals with the optimization of both active and reactive powers. Conventionally, the emphasis in performance optimization of fossil-fuelled power systems has been on economic operation only, using the economic dispatching approach. Known as economic Load Dispatch, the active power optimization itself aims at minimizing the active generation cost, with the generated active powers as the control variables, subject to satisfying system constraints. Whereas, reactive power optimization on the other hand may be defined as the minimization of system real-power transmission loss by controlling bus voltages, transformer tap settings and switchable shunt capacitors/ reactors within the limits specified.

Costs, power system security, and pollutant emissions are, in fact, all areas of concern in power plant operation, and in practice, these three areas are treated in a system to affect a compromise between the frequently conflicting requirements.

## 2.2 Economic Power Dispatch

The economic considerations are important when considering how to operate a power system to supply all the loads at minimum cost. Within this system, we assume that we have some flexibility in adjusting the power delivered by each generator. Economic dispatch assumes an available unit commitment and seeks to optimize the operational schedule such that power balance and physical feasibility constraints such as power flows in the lines and voltage magnitudes in each node are satisfied.

### 2.2.1 Formulation of Economic Power Dispatch

In calculating the optimal dispatch, it is reasonable to neglect line losses, if all generators are located in one plant or are otherwise very close geographically. The problem is to minimize an objective function, $C_T$, which is equal to total fuel costs subject to the constraints requiring that the sum of the powers generated must be equal to the power demanded by the load. Note that any transmission losses are neglected.

Mathematically, this problem can be expressed as minimizing

$$C_T = \sum_{i=1}^{m} C_i(P_{Gi}) = \sum_{i=1}^{m} (a_i + b_i P_{Gi} + c_i P_{Gi}^2) \qquad \text{dollar/hr} \qquad (2.1)$$

where $a_i, b_i$ *and* $c_i$ are fuel cost model parameters.

Subject to the equality constraints:

The equality constraint is simply a statement of conservation of active power in the case of a loss-free transmission system.

$$\sum_{i=1}^{m} P_{Gi} = P_D = \sum_{i=1}^{n} P_{Di} \qquad (2.2)$$

where

$C_T$ : the total production cost,

$C_i$ : the production cost of $i^{th}$ plant.

$P_{Gi}$ : the generation of $i^{th}$ plant

$P_D$ : the total load demand

$P_{Di}$ : the total load demand at bus $i$

$m$ : the total number of dispatchable generating plants

$n$ : the total number of load nodes

$a_i$ : the fixed fuel cost at generator bus $i$

$b_i$ : the variable fuel cost in proportion to active power at generator bus $i$

$c_i$ : the variable fuel cost in proportion to second order term of active power at generator bus $i$

$P_G$ is expressed in MW

The inequality constraints are

$$P_{Gi\ min} \leq P_{Gi} \leq P_{Gi\ max} \qquad (2.3)$$

It is convenient to consider first the case without the inequality constraints (i.e., without the generator limits) given in (2.3). We augment the constraints into the objective function by using the Lagrange multipliers [1]

$$F = C_T + \lambda\left(P_D - \sum_{i=1}^{m} P_{Gi}\right) \qquad (2.4)$$

where

$\lambda$ : incremental cost (IC), is a slope of fuel-cost curve.

The condition for optimum dispatch is [1]

$$\lambda = \frac{dC_i(P_{Gi})}{dP_{Gi}} \qquad i = 1,\ldots\ldots\ldots\ldots m \qquad (2.5)$$

or

$$b_i + 2c_i P_{Gi} = \lambda \qquad (2.6)$$

By substituting for $P_{Gi}$ in the equality constraint (2.2), we have [1]

$$\lambda = \frac{P_D + \sum_{i=1}^{m} \frac{b_i}{2c_i}}{\sum_{i=1}^{m} \frac{1}{2c_i}} \qquad (2.7)$$

The value of $\lambda$ found from (2.7) is substituted in (2.6) to obtain the optimal scheduling of generation.

When the generators limits are included, the Kuhn-Tucker conditions complement the Lagrangian ones including the inequality constraints as the additional terms. The necessary conditions for the optimal dispatch with losses neglected become [1]

$$\frac{dC_i}{dP_{Gi}} = \lambda \qquad \text{for} \qquad P_{Gi\min} < P_{Gi} < P_{Gi\max}$$

$$\frac{dC_i}{dP_{Gi}} \leq \lambda \qquad \text{for} \qquad P_{Gi} = P_{Gi\max}$$

$$\frac{dC_i}{dP_{Gi}} \geq \lambda \qquad \text{for} \qquad P_{Gi} = P_{Gi\min}$$

The numerical solution is the same as before [1].

### 2.3 Economic Power Dispatch with Losses

Transmission losses are a major factor and affect the optimum dispatch of generation in a large interconnected network where power is transmitted over long distances with low load density areas. It then becomes necessary to consider them when developing an optimal dispatch strategy.

The objective function is the same as that defined for Eq. (2.1). However, the equality constraint equation previously shown in Eq. (2.2) must now be expanded to the one shown below.

The active power balance equation:

$$\sum_{i=1}^{m} P_{Gi} - P_L(P_{G2}, \ldots\ldots\ldots\ldots, P_{Gm}) - P_D = 0 \tag{2.8}$$

$P_D$: Total load

$P_L$: Transmission loss

One common practice for including the effect of the transmission losses is to express the total transmission loss as a quadratic function of the generator power outputs.

A general formula, known as Kron's loss formula [3,7], containing a linear term and constant term is frequently used

$$P_L = \sum_{i=1}^{Ng}\sum_{j=1}^{Ng} P_{Gi} B_{ij} P_{Gj} + \sum_{i=1}^{Ng} B_{oi} P_{Gi} + B_{oo} \qquad (2.9)$$

where

$B_{ij}$ = loss coefficients or B coefficients.

B coefficients are assumed constant.

The augmented cost function will become

$$F = C_T + \lambda\left(P_D + P_L - \sum_{i=1}^{n} P_{Gi}\right) \qquad (2.10)$$

Then the necessary optimality conditions turn out to be

$$\partial F / \partial P_{Gi} = \partial C_{Ti} / \partial P_{Gi} - \lambda\left(1 - \partial P_L / \partial P_{Gi}\right) = 0$$

$$\partial L / \partial \lambda = P_D + P_L - \left(\sum_{i=1}^{n} P_{Gi}\right) = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots i = 1,\ldots\ldots, n$$

From these necessary conditions we obtain

$$\lambda = \frac{\partial C_{Ti} / \partial P_{Gi}}{\left(1 - \partial P_L / \partial P_{Gi}\right)}, \ldots\ldots\ldots\ldots\ldots\ldots i = 1,\ldots\ldots, n$$

The quantities $\left(1 - \partial P_L / \partial P_{Gi}\right)^{-1}$ are referred to as the penalty factors.

In addition to the equality constraint, there are inequality constraints that apply to state, control, and output variables. These are classified as follows:

(i) Maximum and Minimum Limits of Power

The generating constraints give the maximum and minimum generating capacity, outside of which it is not feasible to generate due to technical or economic reasons. The generating limits are expressed as follows:

$$P_{Gi\ \min} \leq P_{Gi} \leq P_{Gi\ \max} \tag{2.11}$$

$$Q_{Gi\ \min} \leq Q_{Gi} \leq Q_{Gi\ \max} \tag{2.12}$$

where

$P_{Gi\ \min}$ : The minimum active power generation limit of unit $i$

$P_{Gi\ \max}$ : The maximum active power generation limit of unit $i$

$Q_{Gi\ \min}$ : The minimum reactive power generation limit of unit $i$

$Q_{Gi\ \max}$ : The maximum reactive power generation limit of unit $i$

(ii) Transmission limits

These constraints represent the maximum power which a given transmission line is capable of carrying and are usually based on thermal and dynamic stability considerations and these constraints can be expressed as follows;

$$P_{ij\min} \leq P_{ij} \leq P_{ij\max} \ldots\ldots\ldots\ldots\ldots\ldots i, j = 1, \ldots\ldots, N \tag{2.13}$$

where

$$P_{ij} = |V_i|^2 G_{ij} - |V_i||V_j|G_{ij}\cos(\delta_i - \delta_j) - |V_i||V_j|B_{ij}\sin(\delta_i - \delta_j)$$

N : The number of branches

$P_{ij\min}$ : the minimum active power flow

$P_{ij\max}$ : the maximum active power flow

$P_{ij}(t)$: the power flow between bus $i$ and $j$

$\delta_i$ : voltage angle at bus $i$

$G_{ij}$ : the real elements of nodal admittance matrix between bus $i$ and bus $j$

(iii) Voltage Limits

This is usually a service quality requirement. Thus, to satisfy legal requirements and design limitations, the voltage magnitudes are restricted to lie between specific upper and lower limits expressed as follows:

$$V_{\min i} \le V_i \le V_{\max i} \qquad (2.14)$$

where

$V_{\min}$ and $V_{\max}$ are the minimum and maximum voltage levels respectively. The limits used in this thesis are 0.9 p.u. and 1.10 p.u. respectively.

(iv) Phase Angle Limits

If the power flow between buses $i$ and $j$ is already constrained as expressed by Equation (2.13), there is no need to constrain the voltage phase angles.

The bounds for the phase angle may be varied depending on the problem under consideration or the loading condition.

$$\delta_{\min i} \le \delta_i \le \delta_{\max i} \qquad (2.15)$$

where

$\delta_{\min}$ and $\delta_{\max}$ are the minimum and maximum voltage levels respectively.

In this thesis, by constraining the power flows in the lines, we bound the voltage phase angles from $-\pi/9$ degrees and $\pi/12$ degrees. These values are just for the experiment purpose only.

To augment the objective function if the inequality constraints are included, we need a method for converting constrained optimization problems into unconstrained problems.

Many methods have been proposed to handle the inequality constrained minimization problems in order to obtain faster solutions. In this thesis the exterior penalty function is applied.

### 2.3.1 Exterior Penalty Function Method

In order to convert constrained optimization problems into unconstrained problems we used the exterior penalty-function methods [31]. This method is the easiest to incorporate into the optimization process. We deal with the following problem

Minimize the objective function:

$$C = f_i(x) \qquad (2.16)$$

Subject to:

Equality constraints

$$g_i(x) = 0, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots i = 1,2, \ldots\ldots\ldots, m. \qquad (2.17)$$

Inequality constraints

$$h_i(x) \leq 0, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots i = 1,2, \ldots\ldots\ldots, n. \qquad (2.18)$$

A classical approach uses the sequential unconstrained minimization technique (SUMT) to create a pseudo-objective function of the form:

$$\phi(x, K) = f(x) + KF(x) \qquad (2.19)$$

where

$f(x)$ : the original objective function

$F(x)$ : an imposed penalty function

$K$ : a specified parameter, which determines the magnitude of the penalty.

By considering the equality-constrained problem, using the exterior penalty function method gives the new objective function as

$$\phi(x, K) = f(x) + \sum_{i=1}^{m} K_i \{g_i(x)\}^2 \qquad (2.20)$$

where $K$ is a positive constant. To satisfying the $i^{th}$ constraint, we increase $K_i$ from zero to infinity to give more and more weighting. When $K_i$ is equal to zero, means that the constraint is ignored and when $K_i$ is infinity, the constraint is satisfied exactly. We specify these weighting factors depend on how strongly we feel about satisfying the constraints.

To apply the exterior penalty function to the inequality constraints we have a new objective function as

$$\phi(x,K) = f(x) + \sum_{i=1}^{n} K_i \{h_i(x)\}^2 u_i(h_i) \qquad (2.21)$$

where

$$u_i(h_i) = \begin{cases} 0 & \text{if } h_i(x) \le 0 \\ 1 & \text{if } h_i(x) > 0 \end{cases}$$

When $x$ is located inside the feasible region, the step function $u_i(h_i)$ serves to ignore the constraint. On the other hand, when $x$ is outside the feasible region, the step function $u_i(h_i)$ treats the constraint as an equality constraint. From equation (2.21), we see that no penalty function is imposed if all constraints are satisfied, but whenever one or more constraints are violated, the square of $h_i(x)$ is included in the penalty function.

By considering both of the constraints we have the new objective function as

$$\phi(x,K) = f(x) + \sum_{i=1}^{m} K_i \{g_i(x)\}^2 + \sum_{i=1}^{n} K_i \{h_i(x)\}^2 u_i(h_i) \qquad (2.22)$$

### 2.4 Power Flow Algorithm

The studies of power flow problem, commonly referred to load flow, are the backbone of power system analysis and design. They are necessary for planning, operation, economic scheduling, and exchange of power between utilities. The programs of the power flow are used to study power systems under both normal operating conditions and disturbance conditions.

The main problem includes determination of the magnitudes and phase angle of the voltages at each bus in a power system under balanced three-phase steady-state conditions. Active and reactive power flows in equipment such as transmission lines and transformers, as well as equipment losses, can be computed.

In order to obtain a solution to a power flow problem, the system is assumed to be operating under balanced conditions within which a per/phase model is being used. Four quantities will be included and are associated with each bus. These are voltage magnitude $|V|$, phase angle $\delta$, real power P, and reactive power Q. The system buses are generally classified into three types.

- Swing bus : There is only one swing bus, is taken as reference where the magnitude and phase angle of voltage are specified. The power flow program computes Ps and Qs.
- Load bus : At these buses the active and reactive powers are specified. The power flow program computes V and $\delta$. Most buses in a typical power flow program are load buses.
- Voltage controlled bus : These buses are the generator buses. At these buses, the real power and voltage magnitude are specified. The power flow program computes Q and $\delta$. Examples are buses to which generators, switched shunt capacitors, or static var systems are connected.

These methods will be explained in the next sections.

### 2.4.1 Newton-Raphson Method

Newton-Raphson method is the most widely used method for solving simultaneous non-linear algebraic equations. This method is a successive approximation procedure based on an initial estimate of the unknown and is derived from a Taylor's series expansion, as seen in the following formulae [3]

$$P_k = \sum_{k=1}^{n} |V_k||V_m||Y_{km}|\cos(\theta_{km} - \delta_k + \delta_m)................k = 1,2,......,n$$
$$Q_k = -\sum_{k=1}^{n} |V_k||V_m||Y_{km}|\sin(\theta_{km} - \delta_k + \delta_m)..............k = 1,2,......,n \quad (2.23)$$

where

$P_k$ : the real part of complex power at bus k

$Q_k$ : the imaginary part of complex power at bus k

n : the number of buses

$V_k$ = voltage at bus k

$Y_{km}$ = the element $km$ of admittance matrix of the transmission network

$\theta_{km}$ : phase angle of $Y_{km}$

$\delta_k$ = voltage angle at bus k

The real power and reactive power mismatches at bus k are the difference between the injected power at the bus and the sum of the powers flowing through the various lines connected to other buses can be expressed as

$$\Delta P_k = P_k^{sp} - \sum_{k=1}^{n} |V_k||V_m||Y_{km}|\cos(\theta_{km} - \delta_k + \delta_m) \text{..........} k = 1,2,......,n$$
$$\Delta Q_k = Q_k^{sp} + \sum_{k=1}^{n} |V_k||V_m||Y_{km}|\sin(\theta_{km} - \delta_k + \delta_m) \text{............} k = 1,2,......,n \quad (2.24)$$

where

$P_k^{sp}$ : scheduled real part of power at bus k

$Q_k^{sp}$ : scheduled imaginary part of power at bus k

Equation (2.23) can be expanded in Taylor's series at the present value of the control variables and neglecting all second and higher order terms results in the following set of linear equations.

$$\begin{bmatrix} \Delta P_2^{(0)} \\ \cdot \\ \cdot \\ \Delta Q_n^{(0)} \end{bmatrix} \approx [J] \begin{bmatrix} \Delta \delta_2^{(0)} \\ \cdot \\ \cdot \\ \Delta V_n^{(0)} \end{bmatrix}$$

To distinguish the three different types of buses in the power system, the following notation is used [54].

$$\begin{bmatrix} \Delta P_{PV} \\ \Delta P_{PQ} \\ \Delta Q_{PV} \\ \Delta Q_{PQ} \end{bmatrix} = [J] \begin{bmatrix} \Delta \delta_{PV} \\ \Delta \delta_{PQ} \\ \Delta V_{PV} \\ \Delta V_{PQ} \end{bmatrix} \quad (2.25)$$

where

$$J = \text{the Jacobian matrix} = \begin{bmatrix} \frac{\partial P_{PV}}{\partial \delta_{PV}} & \frac{\partial P_{PV}}{\partial \delta_{PQ}} & \frac{\partial P_{PV}}{\partial V_{PV}} & \frac{\partial P_{PV}}{\partial V_{PQ}} \\ \frac{\partial P_{PQ}}{\partial \delta_{PV}} & \frac{\partial P_{PQ}}{\partial \delta_{PQ}} & \frac{\partial P_{PQ}}{\partial V_{PV}} & \frac{\partial P_{PQ}}{\partial V_{PQ}} \\ \frac{\partial Q_{PV}}{\partial \delta_{PV}} & \frac{\partial Q_{PV}}{\partial \delta_{PQ}} & \frac{\partial Q_{PV}}{\partial V_{PV}} & \frac{\partial Q_{PV}}{\partial V_{PQ}} \\ \frac{\partial Q_{PV}}{\partial \delta_{PV}} & \frac{\partial Q_{PV}}{\partial \delta_{PQ}} & \frac{\partial Q_{PV}}{\partial V_{PV}} & \frac{\partial Q_{PV}}{\partial V_{PQ}} \end{bmatrix}$$

Thus, the relationships between bus power mismatches can be generalized as [3]

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} \qquad (2.26)$$

where

$$\begin{aligned} H_{km} &= \frac{\partial P_k}{\partial \delta_m} = -|V_k||V_m||Y_{km}|\sin(\theta_{km} - \delta_k + \delta_m) \qquad (k \neq m) \\ H_{kk} &= \frac{\partial P_k}{\partial \delta_k} = \sum_{k \neq m} |V_k||V_m||Y_{km}|\sin(\theta_{km} - \delta_k + \delta_m) \end{aligned} \qquad (2.27)$$

$$\begin{aligned} N_{km} &= \frac{\partial P_k}{\partial |V_m|} = |V_k||Y_{km}|\cos(\theta_{km} - \delta_k + \delta_m) \qquad (k \neq m) \\ N_{kk} &= \frac{\partial P_k}{\partial |V_k|} = 2|V_k||Y_{kk}|\cos(\theta_{kk}) + \sum_{k \neq m} |Y_{km}||V_m|\cos(\theta_{km} - \delta_k + \delta_m) \end{aligned} \qquad (2.28)$$

$$\begin{aligned} M_{km} &= \frac{\partial Q_k}{\partial \delta_m} = -|V_k||V_m||Y_{km}|\cos(\theta_{km} - \delta_k + \delta_m) \qquad (k \neq m) \\ M_{kk} &= \frac{\partial Q_k}{\partial \delta_k} = \sum_{k \neq m} |V_k||V_m||Y_{km}|\cos(\theta_{km} - \delta_k + \delta_m) \end{aligned} \qquad (2.29)$$

$$L_{km} = \frac{\partial Q_k}{\partial |V_m|} = -|V_k||Y_{km}|\sin(\theta_{km} - \delta_k + \delta_m) \qquad (k \neq m)$$
$$L_{kk} = \frac{\partial Q_k}{\partial |V_k|} = -2|V_k||Y_{kk}|\sin(\theta_{kk}) - \sum_{k \neq m} |Y_{km}||V_m|\sin(\theta_{km} - \delta_k + \delta_m) \qquad (2.30)$$

Equation (2.26) is solved directly for the incremental voltage angles and incremental voltage magnitudes that are used as corrections to the current estimate of the solution. These computed corrections are used to determine new estimates of the voltage phase angles and voltage magnitudes as follows

$$\delta^{new} = \delta^{old} + \Delta\delta^{new}$$
$$V^{new} = V^{old} + \Delta V^{new} \qquad (2.31)$$

The iterative process is said to have converged when the bus real power mismatch vector $\Delta P$ and reactive power mismatch vector $\Delta Q$ have been reduced to within a specified tolerance.

### 2.4.2 Decoupled Power Flow

The decoupled method assumes that the voltage magnitudes are insensitive to active power computations and that the voltage phase angles insensitive for reactive power computations. This decoupling allows fast computation as e.g. in fast decoupled load flow.

There are several advantages of the decoupled approach [28], namely:

1. To improve computational efficiency, especially for large systems. This is due to a possible situation that each sub problem has approximately half the dimension of the original problem.
2. To enable it to use different optimization techniques in solving the active power and the reactive power OPF sub problems.
3. To allow it to have a different optimization cycle for each sub problem.

The disadvantage of decoupled approach :

For some classes of problems, the constraints (e.g. branch flows) are dependent on both active power and reactive power variables. In these cases, the original OPF problem is not completely separable into two sub problems. A second case involves a decoupled active power OPF that may schedule the transfer of large amounts of active power over long transmission lines when cost-effective generating units are located at a great distance from the major load centers. This could result in unacceptable low voltages at each node, which might not be correctable through the rescheduling of reactive controls in the reactive power OPF alone [24].

#### 2.4.2.1 Fast-Decoupled Power Flow

The Newton-Raphson method is more complicated and requires more computations per iteration and more storage space than the Gauss-Seidel technique. To improve some of the weaknesses of this method, a fast-decoupled method has been developed.

The real power changes are less sensitive to the changes that occur in the voltage magnitude and are most sensitive to the changes in the phase angle. The reactive power changes are less sensitive to those changes in the phase angle and are most sensitive to the ones in voltage magnitude.

Under steady-state conditions, the real power flows, P, are strongly associated with the bus voltage angles δ, and likewise the reactive power flows, Q, with the bus voltage magnitudes, V. The coupling between these P-V and Q-δ variable sets is weak compared to the coupling between P and δ or between Q and V. Therefore; the control variables divide themselves, according to whether they primarily affect the P-δ or the Q-V system operating conditions. The relevance of this is that the optimal power flow calculation may be required to address only the P-δ sub problem, or only the Q-V sub problem. Alternatively, it may schedule all controls simultaneously, giving a more globally optimal solution by taking into account the in-reality non-negligible coupling between the two sub problems.

So the first step in the fast-decoupled approach is to neglect the sub matrices M and N in Eq. (2.26) to yield:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix} \qquad (2.32)$$

or in component form

$$\Delta P = H\,\Delta\delta = \left[\frac{\partial P}{\partial \delta}\right]\Delta\delta$$
$$\Delta Q = L\,\Delta|V| = \left[\frac{\partial Q}{\partial |V|}\right]\Delta|V| \qquad (2.33)$$

where the matrices H and L are given by Eqs. (2.20) and (2.23) and are reformulated as

$$H_{km} = \sum_{m=1}^{n}|V_k||V_m||Y_{km}|\sin\langle\theta_{km} - \delta_k + \delta_m\rangle - |V_k|^2|Y_{kk}|\sin\theta_{kk}$$

$$H_{kk} = -Q_k - B_{kk}|V_k|^2 \qquad (2.34)$$

$$L_{km} = -|V_k||Y_{kk}|\sin\theta_{kk} - \sum_{m=1}^{n}|V_k||V_m||Y_{km}|\sin\langle\theta_{km} - \delta_k + \delta_m\rangle$$

$$L_{kk} = -|V_k|B_{kk} + Q_k \qquad (2.35)$$

where

$B_{kk}$ : the imaginary part of the diagonal elements of the bus admittance matrix, is the sum of susceptances of all the elements incident to bus k.

According to [3], in a typical power system, the self-susceptance $B_{kk} >> Q_k$, and we may neglect $Q_k$. Further simplification is obtained by assuming $|V_k|^2 \approx |V_k|$, and $|V_m| \approx 1$. Under operating conditions, $\delta_k - \delta_m$ is quite small. In addition, the transformer and line reactances are normally greater than the corresponding resistances.

$\cos(\delta_k - \delta_m) = 1 \qquad G_{km} << B_{km}$

$\sin(\delta_k - \delta_m) = 0 \qquad Q_k << B_{kk}V_k^2$

we have [3]:

$$H_{km} = -|V_k| B_{km}$$
$$H_{kk} = -|V_k| B_{kk} \tag{2.36}$$

and

$$L_{km} = -|V_k| B_{km}$$
$$L_{kk} = -|V_k| B_{kk} \tag{2.37}$$

With these assumptions, equation (2.19) can be rewritten as follows

$$\left[\frac{\Delta P_k}{V_k}\right] = -B'\Delta\delta \tag{2.38}$$

$$\left[\frac{\Delta Q_k}{V_k}\right] = -B''\Delta V \tag{2.39}$$

or

$$\Delta\delta = -\left[B'\right]^{-1}\frac{\Delta P}{|V|} \tag{2.40}$$

$$\Delta|V| = -\left[B''\right]^{-1}\frac{\Delta Q}{|V|} \tag{2.41}$$

The matrices B' and B" are the imaginary parts of the bus admittance matrix Ybus. Matrix B' may be obtained from $Y_{bus}$ by stripping away the first row and column and then taking the imaginary part. So, B' is of order of (n-1), and B" is of order of (n-1-m), where m is the number of voltage-regulated buses.

These matrices are real, sparse and constant matrices, which respectively have the same size as the Jacobian submatrices H and L, and have the same sparsity characteristics as Ybus.

Convergence is achieved when both the real and reactive power mismatches are within specified tolerances. The fast decoupled power flow solution requires more iterations than the Newton-Raphson method, but requires considerably less time per iteration, and a power flow solution is obtained very rapidly.

# Chapter 3

# Artificial Neural Network

## 3.1 Introduction

A neural network is defined as a set of highly interconnected processing elements, i.e. nodes or neurons that are capable of learning information presented to them. Neural networks are inspired by biological nervous systems and usually consist of one input layer, one or more hidden layers, and one output layer. As in nature, the network function is determined largely by the connections between the elements. These connections which exist between the nodes of adjacent layers relay the output signals from one layer to the next. All inputs to a node are weighted, combined, and then processed through a transfer function that controls the strength or the signal relayed through the output connections of the nodes. It is also possible to train the neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

In general, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning mode, to train a network.

In recent years, artificial neural networks (ANN) have been proposed as an alternative method for solving certain difficult power system problems where the conventional techniques have failed to achieve the desired speed, accuracy, or efficiency [20].

An optimization problem is usually defined in terms of the minimization of a scalar function of a number of variables. If the variables are not constrained by inequality or equality relationships, the optimization is said to be unconstrained. The function thus in short, is an objective function, sometimes also called a cost function or criterion function.

In neural network studies, the objective functions take the following forms [52]:

1. For feed forward networks: The scalar error function E (w) in the weight space is the objective function.
2. For recurrent networks: The scalar energy function E (v) is the objective function in the network output space.

There are three basic elements of the neural network model (Fig. 3.1)

1. A set of synapses or connecting links, each of which in characterized by a weight or strength of its own.
2. An adder for summing the input signals, weighted by the respective synapses of the neuron.
3. An activation function for limiting the amplitude of the output of a neuron. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0,1] or alternatively [-1,1]

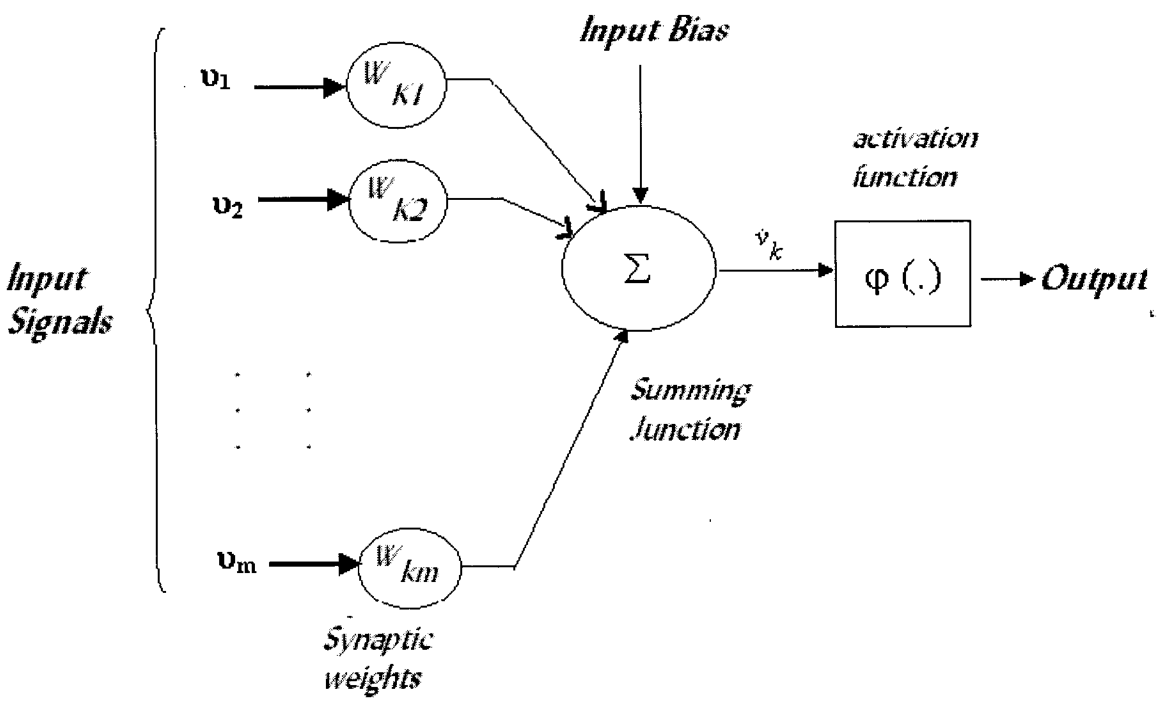


Figure 3.1 Nonlinear model of neuron

### 3.1.1 Types of Activation Function

The activation function denoted by $\varphi(v)$, defines the output of a neuron in terms of the induced local field $v$. Here we identify three basic types of activation functions [2].

1. Threshold function or unipolar binary function. For this type of activation function as illustrated in Fig. 3.2, we have [2]

$$\varphi(v) = \begin{cases} 1 & if\ v \geq 0 \\ 0 & if\ v < 0 \end{cases} \qquad (3.1)$$

Correspondingly, the output of neuron k employing such a threshold function is expressed as [2]

$$y_k = \begin{cases} 1 & if\ v_k \geq 0 \\ 0 & if\ v_k < 0 \end{cases} \qquad (3.2)$$

where $v_k$ is the induced local field of the neuron; that is [2]

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \qquad (3.3)$$

Such a neuron is referred to as the McCulloch-Pitts model. In this model, the output of a neuron takes on the value of 1 if the induced local field of that neuron is nonnegative, and 0 otherwise. This statement describes the all-or-none property of the McCulloch-Pitts model. This model is also called a hard limiting activation function or binary function. Networks such as this, whose output is either 0 or 1, are potentially suitable for pattern classification problems in which we wish to divide the patterns into two classes, labelled "0" and "1".

2. Piecewise-Linear Function or unipolar ramp function. For this type of activation function described in Fig. 3.3, expressed as [2]

$$\varphi(v) = \begin{cases} 1 & if\ v \geq .5 \\ v & if\ .5 > v > -.5 \\ 0 & if\ v \leq -.5 \end{cases} \qquad (3.4)$$

where the amplification factor inside the linear region of operation is assumed to be unity. This form of the activation function may be viewed as an approximation to a non-linear amplifier.

3. Sigmoid Function. The sigmoid function, whose graph is s-shaped, is by far the most common form of activation function used in the construction of artificial neural networks. It is defined as a strictly increasing function that exhibits a graceful balance between linear and non-linear behaviour. The output of the network is a real number and is not simply either 0 or 1 as for the binary function.

An example of the sigmoid function is the logistic function or unipolar continuous activation function, defined by [2]

$$\varphi(v) = \frac{1}{1+\exp(-av)} \qquad (3.5)$$

in which a is the slope parameter of the sigmoid function. By varying the parameter a, we obtain sigmoid functions of different slopes, as illustrated in Fig. 3.4. In fact, the slope at the origin is equal to a/4. In the limit, as the slope parameter approaches infinity, the sigmoid function becomes simply a threshold function. Whereas a threshold function assumes the values of 0 or 1, a sigmoid function assumes a continuous range of values from 0 to 1. Note also that the sigmoid function is differentiable, whereas the threshold function is not.

The activation functions defined in Eq. (3.5), range from 0 to +1. It is sometimes desirable to have the activation function range from –1 to +1, in which case the activation function assumes an antisymmetric form with respect to the origin; that is, the activation function is an odd function of the induced local field.

Specifically, the threshold function of Eq. (3.1) is now defined as [2]

$$\varphi(v) = \begin{cases} 1 & if\ v > 0 \\ 0 & if\ v = 0 \\ -1 & if\ v < 0 \end{cases} \qquad (3.6)$$

which is commonly referred to as the signum function. For the corresponding form of a sigmoid function we may use the hyperbolic tangent function (Fig. 3.5) or bipolar continuous activation function, defined by [2]

$$\varphi(v) = \tanh(v/2) = \frac{1-\exp(-v)}{1+\exp(-v)} = \frac{2}{1+\exp(-v)} - 1 \qquad (3.7)$$

The hyperbolic tangent activation function is a logistic activation function biased and rescaled.

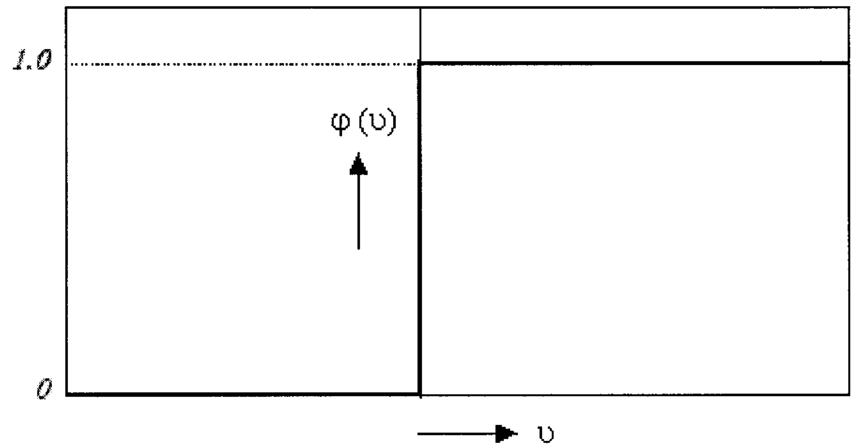


Figure 3.2 Threshold activation function

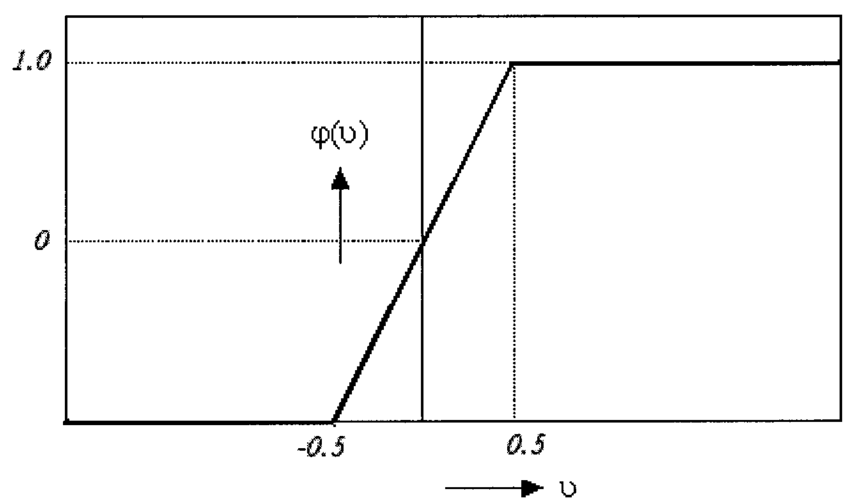


Figure 3.3 Piecewise-Linear activation Function

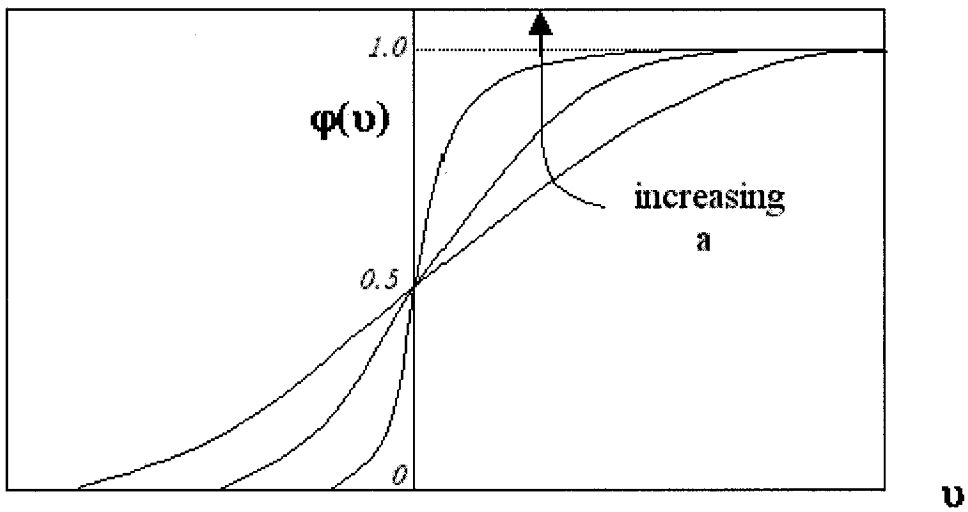


Figure 3.4 Logistic activation function

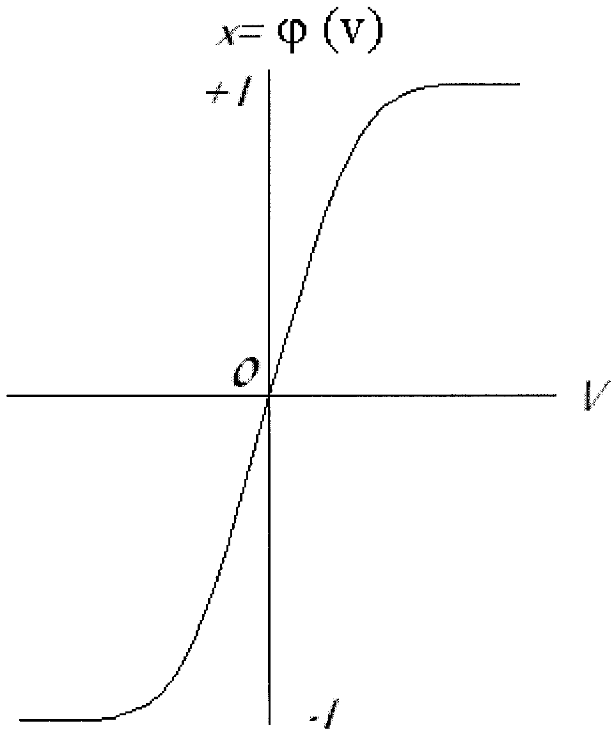


Figure 3.5 Hyperbolic tangent activation function

The hard limiting activation function as illustrated in Eq. (3.1) and Eq. (3.6) describes the discrete neuron model. The soft-limiting activation functions are often called sigmoidal characteristics.

Essentially, any function $f(net)$ that is monotonically increasing and continuous such that $net \in \Re$ and $f(net) \in (-1, 1)$ can be used instead of the soft limiting activation function in neural modeling. A few neural models that often involve some form of feedback require the use of another type of nonlinearity than that defined in (3.5) and (3.7).

**3.2 Back Propagation NN**

The usage of the term "back-propagation" appears to have evolved after 1985 when its use was popularized through the publication of the seminal book entitled *Parallel Distributed Processing* [42].

The back Propagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual outputs of multilayer feed-forward preceptrons and the desired output. It requires continuous differentiable non-linearities. The error signal at the output of neuron j at iteration $n$ (i.e., presentation of the nth training example) is defined by [2]

$$e_j(n) = \hat{Y}_i(n) - Y_i(n) \qquad (3.8)$$

where

$\hat{Y}_i(n)$ = actual output of $i$ th. neuron

$Y_i(n)$ = target output of $i$ th. neuron for a training set

The neuron j is an output node.

Haykin [2] defines the instantaneous value of the error energy for neuron j as $\frac{1}{2}e_j^2(n)$. Correspondingly, the instantaneous value $E(n)$ of the total error energy is obtained by summing $\frac{1}{2}e_j^2(n)$ over *all neurons in the output layer;* these are the only 'visible" neurons for which error signals can be calculated directly. Thus

$$E(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \qquad (3.9)$$

where the set $C$ includes all the neurons in the output layer of the network. Let $N$ denote the total number of patterns (examples) contained in the training set. The *average squared error energy* is obtained by summing *E(n)* over all n and then normalizing with respect to the set size $N$, as shown by [2]

$$E_{av} = \frac{1}{N}\sum_{n=1}^{N} E(n) \qquad (3.10)$$

The instantaneous error energy *E (n),* and therefore the average error energy $E_{av}$ is a function of all the free parameters (i.e., synaptic weights and bias levels) of the network. For a given training set, $E_{av}$ represents the *cost function* as a measure of learning performance. The objective of the learning process is to adjust the free parameters of the network to minimize $E_{av}$. Specifically, we consider a simple method of training in which the weights are updated on a *pattern-by-pattern* basis until one *epoch,* that is one complete presentation of the entire training set has been dealt with. The adjustments to the weights are made in accordance with the respective errors computed for each pattern presented to the network. The model of each neuron in the network includes a *nonlinear activation function.* Learning is accomplished by changing the value of the weights to achieve the desired result, i.e., the correct classification. The learning process adopted is supervised learning in which the desired output is known.

Rumelhart et al. [42] demonstrated a feed forward layered machine of the perceptron type, that could train itself autonomously as desired if it is used for activation at the *neurons* and if a backward propagation of error algorithm is used to change the interconnection weights and activation function threshold until proper recognition capability had been attained.

The back propagation-training algorithm uses a gradient search technique to minimize a cost function equal to the mean square difference between the desired output and the actual net outputs of a multilayer feed-forward perceptron [55].

$$E = \sum_{n=1}^{N} E(n) = \sum_{n=1}^{N} \sum_{i=1}^{k} \frac{1}{2}\left(\hat{Y}_i(n) - Y_i(n)\right)^2 \qquad (3.11)$$

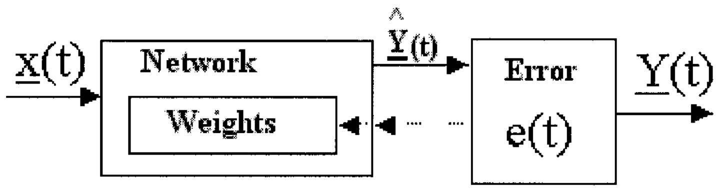


Figure 3.6 Back-propagation training

The process begins with selecting small random weights and internal thresholds and then presenting all training data repeatedly to train the net. Weights are adjusted after every trial using side information specifying the correct class until weights converge and the cost function is reduced to an acceptable value. An essential component of the algorithm is the iterative method described below that propagates error terms required to adapt weights back from nodes in the output layer to nodes in lower layers.

### 3.2.1 The back-propagation training algorithm

Within this type of algorithm, several processes are executed. The first step is to choose small random number values for the hidden layer weights V and the output weights W. This process is called initialize weights.

The second step is to present the inputs and desired outputs. In the learning process, the network is presented with a pair of patterns, an input pattern and a corresponding desired output pattern. The input and desired output patterns are [52]:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_I \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_k \end{bmatrix}$$

The third step is to calculate the actual outputs $\hat{Y}(n)$, and the error $E(n)$ for that set of weights.

Assuming the use of a logistic function for the sigmoidal nonlinearity as expressed in (3.5).

The output vector matrix is [52]:

$$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \cdot \\ \cdot \\ \hat{y}_k \end{bmatrix}$$

$z_j = f(v_j^t x)$ j=1,2,..............J

$\hat{y}_k = f(w_k^t z)$ k=1,2,.............K

where

$v_j$ is the $j$'th row of V.

$w_k$ is the k'th row of W.

Calculate the average squared error as expressed in (3.11).

Using its weights and thresholds, the network produces its own output pattern, which is compared with the desired output pattern.

The next step is to update the weights. In this process we calculate the derivatives of E with respect to all of the weights. If increasing a given weight would lead to more error, we adjust that weight downwards. If increasing a weight leads to less error, we adjust it upwards.

Learning comprises changing the weights and thresholds so as to minimize this error function in a gradient decent manner. Assume that the gradient descent search is performed to reduce the error E through the adjustment of weights [52].

$$\Delta w_{kj} = \eta \delta_k z_j \qquad (3.12)$$

where $\delta_k$ is the error signal at an output unit k is given by [52] :

$$\delta_k = \left(Y_k - \hat{Y}_k\right)\hat{Y}_k\left(1 - \hat{Y}_k\right) \qquad \text{for } k=1,2,\ldots\ldots\ldots K \qquad (3.13)$$

$\eta$ = learning rate parameter

The weight adjustment in the hidden layer is [52]:

$$\Delta v_{ji} = \eta \delta_j x_i \qquad (3.14)$$

where $\delta_j$ is the error signal at an output unit $j$ is given by [52]:

$$\delta_j = z_j(1 - z_j)\sum_k \delta_k w_{kj} \qquad \text{for } j=1,2\ldots\ldots\ldots\ldots J \qquad (3.15)$$

In practice, one way to increase the learning rate without causing oscillations is to modify the expression of $\Delta W_{kj}$ above to include a momentum term, i.e. [52]:

$$\Delta w_{kj}(n+1) = \eta \delta_k z_j + \alpha \Delta w_{kj}(n) \qquad (3.16)$$

where

n = the presentation number

$\alpha$ = constant which determines the effect of past weights changes on the current direction of movement in the weight space.

After adjusting all the weights up or down, we start all over, and keep on going through this process until the weights settle down or the average squared error computed over the entire training set is at a minimum or acceptably small value.

In this thesis method, a tangent hyperbolic function or bipolar continuous function is used for the sigmoidal nonlinearity. The error signal terms will become [52]

$$\delta_k = \frac{1}{2}\left(Y_k - \hat{Y}_k\right)\left(1 - \hat{Y}_k^2\right) \qquad \text{for } k=1,2,\ldots\ldots\ldots K$$

$$\delta_j = \frac{1}{2}(1 - z_j^2)\sum_k \delta_k w_{kj} \qquad \text{for } j=1,2\ldots\ldots\ldots\ldots J.$$

The process is the same as that when logistic sigmoidal function is used.

### 3.3 Hopfield Neural Network

Neural Networks have been used to solve a variety of constrained optimization problems. Due to their fast network convergence to optimal solutions, they eliminate the time bottlenecks that usually arise in most sequential algorithms. In the problem of optimization, the Hopfield neural network has a well-demonstrated capability of finding solutions to difficult optimization problems. Since Hopfield and Tank proposed an artificial neural network in 1985 for solving difficult optimization problems like the Travelling Salesman Problem (TSP), many researchers have applied this approach to a number of constrained optimization problems.

Solving optimization problems requires minimization of some cost function subject to a set of constraints. These cost functions are known as energy functions, and the neural network will produce optimal solutions by minimizing an energy function.

In power systems, the Hopfield networks have been applied to optimal power flow and economic load dispatch problems [14, 16, 17, 18, 19, 20, 21, 23]. The Hopfield network consists of a set of neurons and a corresponding set of unit delays, forming a multiple-loop feedback system as shown in Fig. 3.7. It is such that each neuron contains two op amps. The output of neuron j is connected to the input of neuron $i$ through a conductance $W_{ij}$. The Hopfield network may be operated in continuous mode or discrete mode, depending on the model adopted for describing the neurons. The continuous mode of operation is based on an additive model as shown in Fig. 3.8. On the other hand, the discrete mode of operation is based on the McCulloch-Pitts model. In formulating the energy function E for a continuous Hopfield model, the neurons are permitted to have fully connected. A discrete Hopfield model, on the other hand, need not have fully-connected.

The standard Hopfield Neural Network can be described as follows.

Neuron i has an input $u_i$, and an output $v_i$, and is connected to neuron j with a weight $W_{ij}$. A connection from a positive output is known as excitatory, and a connection from a negative output is called an inhibitory connection. Associated with each neuron is also an input bias terms $I_i$. The node equation for the continuous-time network with n neurons is given by [25]:

$$u_i = \sum_{j=1}^{n} W_{ij} v_j + I_i \qquad (3.17)$$

where

$u_i$ : the total input to neuron i.

$W_{ij}$ : the synaptic interconnection strength from neuron j to neuron i.

$I_i$ : the external input to neuron i.

$v_j$ : the output of neuron j.

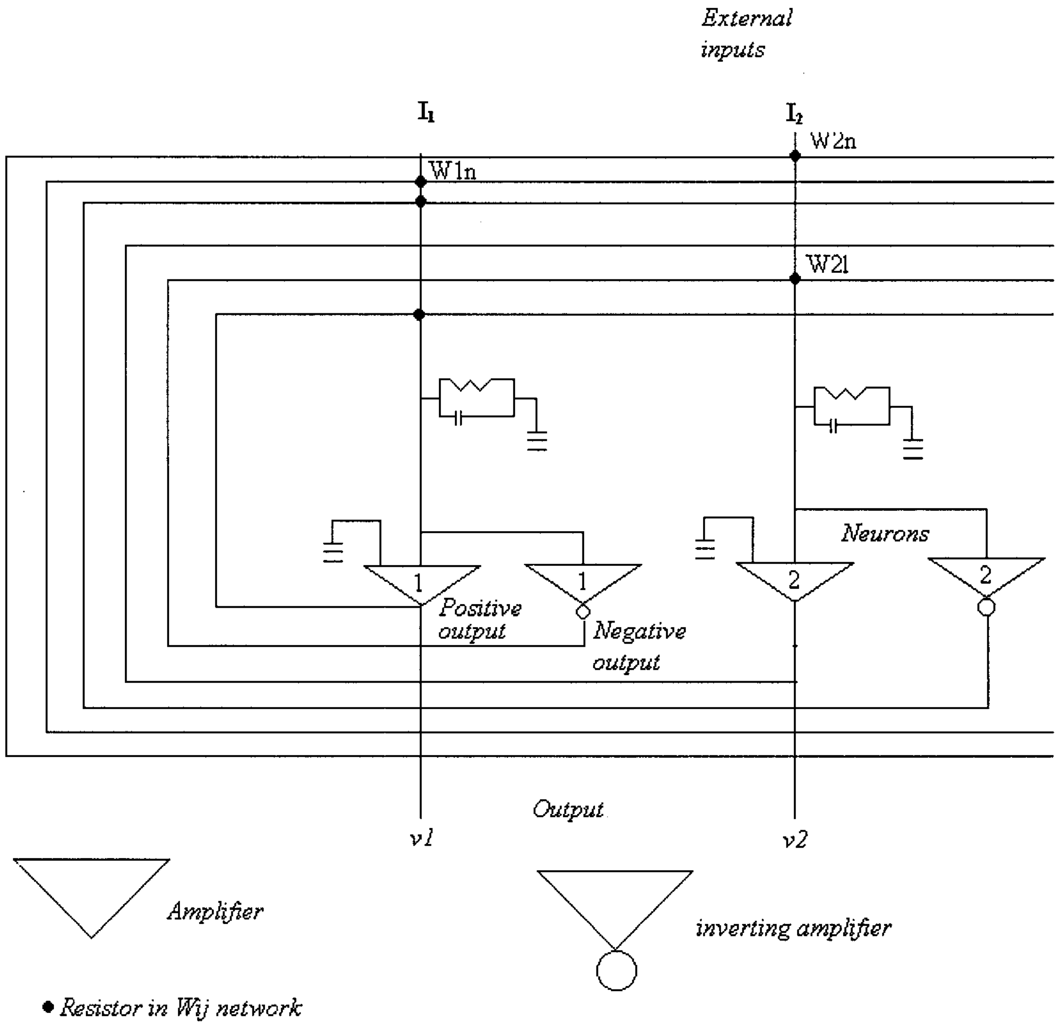


Figure 3.7 Hopfield Neural Network.

The output of the neuron is determined by the network input and the activation function of the neuron [2].

$$v_i = f(u_i) \qquad (3.18)$$

$v_i$ is a continuous variable in the interval 0 to 1, and $f(u_i)$ is a monotonically increasing function which constrains $v_i$ to this interval, is usually a hyperbolic tangent of the form as given by :

$$f(u_i) = \frac{1}{1+\exp(-\lambda u_i)} \qquad (3.19)$$

where $\lambda$ is a constant called the gain parameter.

The equation represents a continuous, non-decreasing, and differentiable function called a sigmoid function.

As shown in Figure (3.7) and (3.8), each neuron receives an external current $I_i$, which could represent actual data provided by the user of the neural network. From Figure 3.8, the dynamics of the Hopfield network can be described by [2]

$$C_i \frac{du_i}{dt} = \sum_{j=1}^{N} W_{ij} v_j - \frac{u_i}{R} + I_i \qquad (3.20)$$

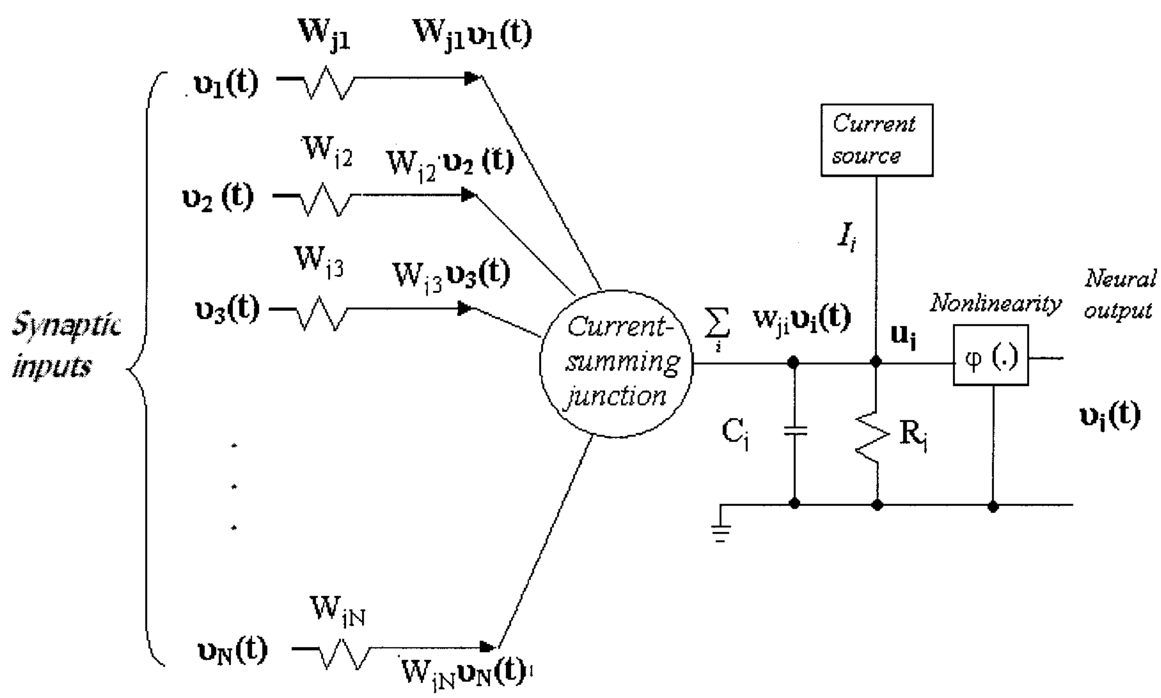


Figure 3.8 Input node of $i^{th}$ neuron

Dividing by C and redefining $\frac{W_{ij}}{C}$ and $\frac{I_i}{C}$ as $W_{ij}$ and $I_i$, the dynamics of the network become [2]:

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} + \sum_{j=1}^{N} W_{ij} v_j + I_i \qquad (3.21)$$

where $\tau = RC$ is the circuit's time constant, and N is the number of neurons in the network.

Hopfield has shown that the dynamics of the neurons follow a gradient decent of the quadratic energy function [2,14]

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} v_i v_j - \sum_{i=1}^{N} I_i v_i \qquad (3.22)$$

Hopfield has also shown that while the state of the neural network evolves inside the N-dimensional hypercube defining by $v_i \in \{0,1\}$, the minima of the energy function occur at $2^N$ corners of this space, only if $\lambda_i = \infty$.

In terms of the energy function, the dynamics of the $i^{th}$ neuron are described by [2]

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} - \frac{\partial E}{\partial v_i} \qquad (3.23)$$

### 3.3.1 Optimization by using Hopfield Neural Network

The general idea is discussed by taking the non-linear programming problem [20]:

Minimize $\quad f(x) = \frac{1}{2} x^t P x + q^t x \qquad (3.24)$

Subject to $\quad g_i^t x = s_i \qquad i = 1 \ldots\ldots n \qquad (3.25)$

$$w_i^t x \le d_i$$
or (3.26)
$$w_i^t x \ge d_i \qquad i = 1.....m$$

The mapping of the optimization problem to the Hopfield Neural Network is illustrated by first ignoring the inequality constraints. Let us now relate the variable x to the neuron output $v$ and the following energy function converges to its minimum [20]:

$$E = \alpha F(v) + \sum_i \beta_i [G_i(v)]^2 \qquad (3.27)$$

Here the functions $F(v)$ and $G(v)$ correspond to the objective and equality constraints and must satisfy some conditions required to be an energy function. The equality constraints are taken into account by adding terms $[\,G(v)\,]^2$ to E in order to minimize the mismatch to zero, otherwise the solution produced may have insufficient generation. Positive coefficients $\alpha$ and $\beta$ are used to determine the relative importance of each constraint and objective. Note that the energy function will contain the m equality constraint terms in addition to the objective, and that the converged solution may not be the global optimum.

There are some methods for handling inequality constraints. Tank & Hopfield solve the inequality constraints based on a Linear Programming problem.

The linear Optimization formulation may be presented as [25] :

min $C = \vec{A} \cdot \vec{V}$

Subject to

$\vec{D}_j \cdot \vec{V} \ge \vec{B}_j \qquad$ j = 1,..........,M

$$\vec{D}_j = \begin{bmatrix} D_{j1} \\ D_{j2} \\ \cdot \\ \cdot \\ \cdot \\ D_{jN} \end{bmatrix}$$

where

$\vec{A}$ : an N-dimensional vector of coefficients for the N variables which are the component of $\vec{V}$.

M : the number of constraint

$\vec{D}_j$ : variable coefficients in a constraint equation.

$B_j$ : the bounds

If the equality constraints are considered as well, they each can be replaced by two inequality constraints, so that the following discussion still holds.

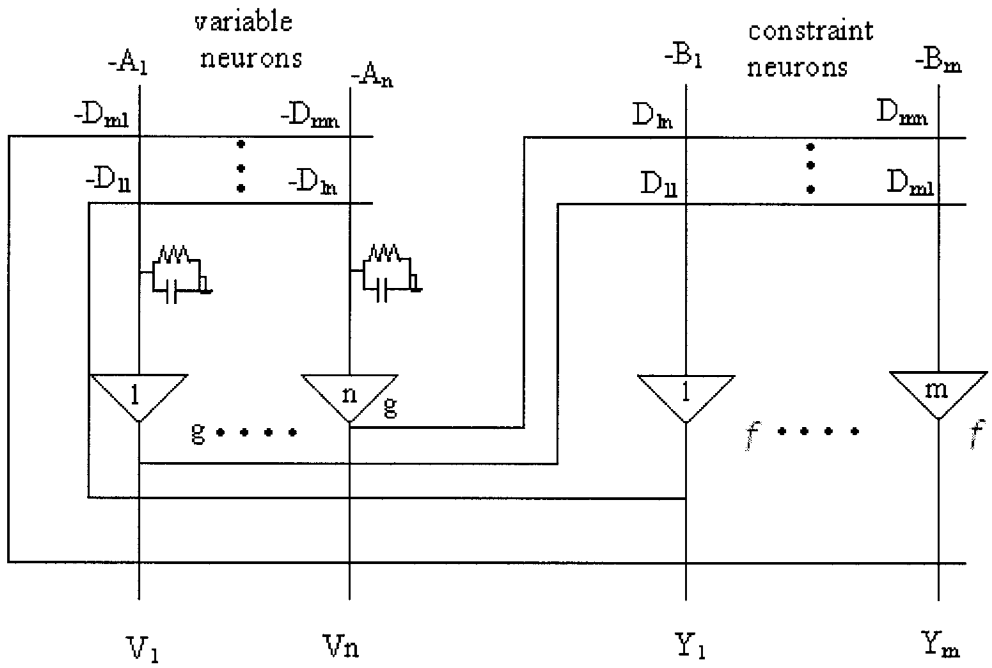


Figure 3.9 Neural Network to solve a linear programming problem

A network can be organized from these objectives and constraints as seen in Fig. 3.9 as follows:

In the circuit of Fig. 3.9:

- The N outputs $V_i$ represent the values of the variables in the linear programming problem.
- The components of $\vec{A}$ are proportional to input currents fed into these amplifiers.
- The M outputs $\psi_j$ represent constraint satisfaction. This output injects current into the input lines of the $V_i$ variable amplifiers by an amount proportional to $-D_{ji}$
- The input-output relations of $V_i$ amplifiers are linear: $V_i = g\ (u_i)$
- The $\psi_j$ amplifiers have the nonlinear input-output relation characteristic by the function [25] :

  $$\psi_j = f(u_j),$$

$$u_j = \vec{D}_j \cdot \vec{V} - B_j$$

$$f(z) = 0, \qquad z \geq 0$$

$$f(z) = -z, \qquad z < 0$$

The circuit equation for the variable amplifiers can be written as [25]:

$$C_i \frac{du_i}{dt} = -A_i - \frac{u_i}{R} - \sum_j D_{ji} f(u_j)$$
$$= -A_i - \frac{u_i}{R} - \sum_j D_{ji} f(\vec{D}_j \cdot \vec{V} - B_j) \qquad (3.28)$$

Consider an energy function of the form [25]

$$E = (\vec{A} \cdot \vec{V}) + \sum_j F(\vec{D}_j \cdot \vec{V} - B_j) + \sum_i \frac{1}{R} \int_0^{V_i} g^{-1}(V) dV \qquad (3.29)$$

where

$$f(z) = \frac{dF(z)}{dz}$$

The time derivative of E is [25]:

$$\frac{dE}{dt} = \sum_i \frac{dV_i}{dt} \left[ \frac{u_i}{R} + A_i + \sum_j D_{ji} f(\vec{D}_j \cdot \vec{V} - B_j \right] \qquad (3.30)$$

By substituting for the bracketed expression from equation (3.28) gives [25]:

$$\frac{dE}{dt} = -\sum_i C_i \frac{dV_i}{dt} \frac{du_i}{dt} = -\sum_i C_i g^{-1}(V_i) \left( \frac{dV_i}{dt} \right)^2 \qquad (3.31)$$

Since $C_i$ is positive and $g^{-1}(V_i)$ is monotone increasing function, this sum is nonnegative and

$$\frac{dE}{dt} \leq 0 \; ; \quad \frac{dE}{dt} = 0 \rightarrow \frac{dV_i}{dt} = 0 \; , \text{ for all i.} \qquad (3.32)$$

Thus, the time evolution of the system is a motion in state space, which seeks out a minimum to E and stops.

Abe et al. [13] introduced a slack variable to convert inequality constraints into equality constraints.

The energy function $E$ can be formulated as follows [13].

$$E = AE_1 + BE_2 + CE_3 \qquad (3.33)$$

where

$E_1$ : The objective function form Eq. 3.24

$E_2$ : the energy corresponding to the equality constraints (3.25)

$E_3$ : the energy corresponding to the inequality constraints (3.26)

$A, B\ and\ C > 0$ : the weights in the energy function

To obtain the energy function $E_2$, they used the same approach as Hopfield & Tank. For handling inequality constraints, they introduced a variable $y_i$ to convert Eq. 3.26 into equality constraints as follows [13].

$$\begin{aligned} &d_i y_i - w_i^t x = 0 \quad where \quad 1 \geq y_i \\ &and \\ &d_i y_i - w_i^t x = 0 \quad where \quad y_i \geq 1 \end{aligned} \qquad (3.34)$$

The energy function of $E_3$ can be calculated by squaring Eq. 3.34 divided by two.

By introducing the internal variable vectors $u\ and\ \zeta$, and extending the range of $x_i$ to $[0,1]$, the model for energy function can be written as [13]:

$$x_i = .5(1 + \tanh u_i / \tau_i) \quad for \quad 1 \geq x_i \geq 0 \qquad (3.35)$$

$$y_i = \begin{cases} 1 - \exp(-\zeta_i / \rho_i) & if \quad y_i \leq 1 \\ 1 + \exp(\zeta_i / \rho_i) & if \quad y_i \geq 1 \end{cases} \qquad (3.36)$$

$$\begin{pmatrix} \frac{du}{dt} \\ \frac{d\zeta}{dt} \end{pmatrix} = -\begin{pmatrix} \frac{\partial E}{\partial x} \\ \frac{\partial E}{\partial y} \end{pmatrix} = -(J)\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} b \\ 0 \end{pmatrix} \tag{3.37}$$

where

$u = (u_1, \ldots\ldots\ldots\ldots\ldots, u_n)^t, \quad \infty > u_i > -\infty$

$\zeta = (\zeta_i, \ldots\ldots\ldots\ldots\ldots, \zeta_n)^t, \quad \infty > \zeta_i > -\infty$

$\tau_i (> 0) \quad for \quad i = 1, \ldots\ldots, n;$ constant, and

$\rho_i (> 0) \quad for \quad i = 1, \ldots\ldots\ldots, n;$ constant.

Thus, integrating (3.35) to (3.37) for arbitrary initial values of $u$ *and* $\zeta$ we obtain a local minimum solution in the sense that the energy function $E$ is locally minimized.

In this thesis, to convert the constrained optimization problem into unconstrained problem by considering just equality constrained, the same approach as Hopfield & Tank is used. To handle inequality constraints, the exterior penalty function method is applied [4,30,31].

### 3.3.2 Hopfield Network Algorithm

Step 1 Assign Connection Weights

This step is to map the objective function to energy expressions of the Hopfield Neural Network. In this process we obtain the weights $W$ and the input bias $I$.

Step 2 Initialize with unknown input pattern

First, we choose the initial guess for the output $v_i$. Calculate $u_i$ by using the inverse of equation (3.18). This process is called initialize inputs.

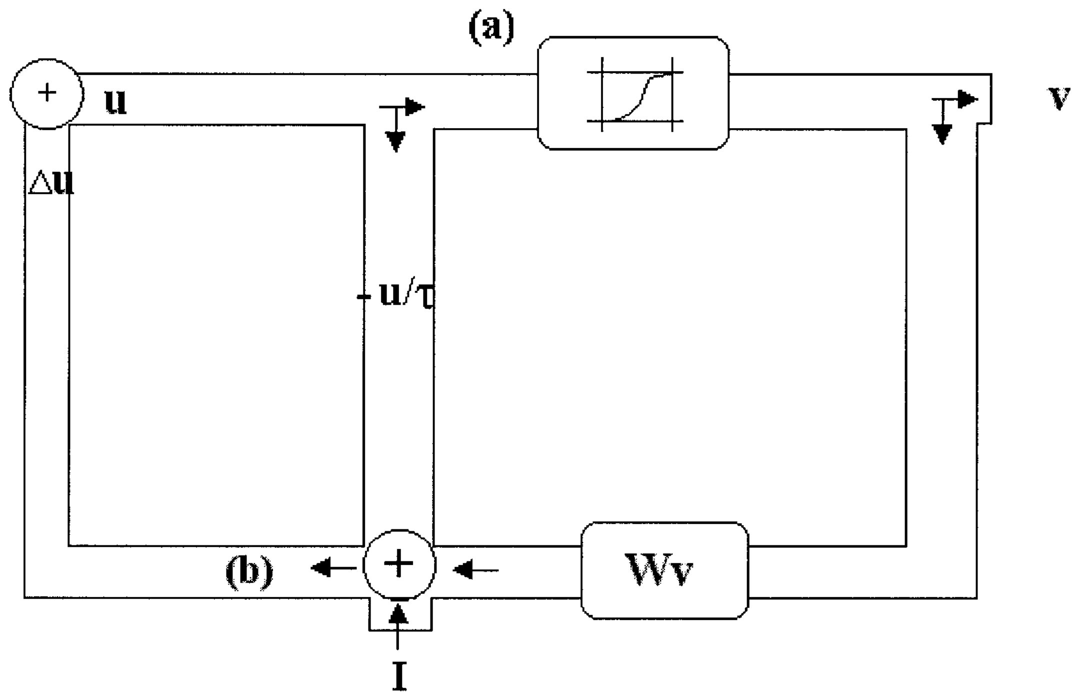


Figure 3.10 Schematic diagram of the continuous Hopfield network

(a) Non-linear threshold functions constraining $v$ to the unit hypercube.

(b) The change in $u$ is specified by the differential equation (3.21)

Step 3 Iterate until convergence

This process is illustrated as in Fig. 3.10.

In (a) the process begins by expressing the output $v$ as a function of the input $u$ by applying the activation function as Eq. (3.5) or (3.7).

The input $u$ is update by implement dynamical transients in the network as expressed in Eq. (3.21) as illustrated in process (b).

The process is repeated until node outputs remain unchanged with further iterations.

Step 4 Print out the output.

This model is used in this thesis to handle the problems of both economic and security-constrained dispatch.

# Chapter 4

# Economic Power Dispatch with Hopfield Neural Network

## 4.1 Background

The accomplishment of economic dispatch in power system operation consists of minimizing the operating cost depending on demand and subject to certain constraints, i.e. how to allocate the required load demand between the available generation units. Many methods have been applied to Economic Dispatch to achieve better solutions. Many of them suffer from slow convergence or even oscillating around the optimal solution. Therefore, there is a need to develop a more stable technique for handling the problem mentioned above.

For the combinatorial optimization problem with equality constraints, the Hopfield model is considered to be appropriate. Many cases in combinatorial optimization problems involve a linear combination of variables that are upper and lower bounded. In such cases, the standard Hopfield model requires some modifications.

Abe et al.[13], solved inequality constrained combinatorial optimization by introducing slack variables. The number of slack variables is always proportional to the number of constraints. Therefore, we have to assume many slack variables for solving the optimization problem that have many constraints. The problem would be very complicated, especially with the large-scale systems.

In this thesis, the Hopfield model is modified to handle inequality constraints by using the exterior penalty function method. The algorithm determines the weights and input bias in energy function is determined in this chapter.

**4.2 Hopfield Neural Network Based Economic Power Dispatch**

The economic power dispatch problem is formulated to find the optimal condition of power generation to minimize the total fuel cost, represented as [14]:

$$Cost = \sum_{i=1}^{Ng} \left( a_i + b_i P_{Gi} + c_i P_{Gi}^2 \right) \qquad (4.1)$$

The problem constraints include the power balance equation of meeting the power demand and power losses [14].

$$P_D + P_L = \sum_{i=1}^{Ng} P_{Gi} \qquad (4.2)$$

where

$P_D$ is the total load demand, and

$P_L$ is the total of power losses, that can be expressed as [3,7]:

$$P_L = \sum_{i=1}^{Ng} B_0 P_{Gi} + \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} B_{ij} P_{Gi} P_{Gj} + B_{00} \qquad (4.3)$$

In some cases, losses are assumed constant. The active power scheduling is required to satisfy the upper and lower bound of power generations.

$$P_{Gi\min} \le P_{Gi} \le P_{Gi\max} \qquad (4.4)$$

By applying Hopfield Neural Network based on the following neuron dynamic equation:

$$\frac{dU_i}{dt} = \sum_j T_{ij} V_j + I_i \quad (4.5)$$

$U_i$ is the total input of the neuron $i$, can be expressed as:

$$U_i = \sum_{i \neq j} T_{ij} V_j + I_i \quad (4.6)$$

where

$T_{ij}$ = the weight connection between neuron $j$ and neuron $i$

$V_i$ is the output of neuron $i$ , can be represented as a function of $U_i$ as

$$V_i = g(U_i)$$

where $g(U_i)$ is a non linear function at neuron $i$ , is typically a sigmoid function.

In this thesis, we use the two forms of sigmoid function, i.e. the tangent hyperbolic given by [2]:

$$g(U_i) = \{1 + \tanh(\lambda U_i)\}/2 \quad (4.7)$$

and the logistic activation function as

$$g(U_i) = \frac{1}{1 + \exp(-\lambda U_i)} \quad (4.8)$$

The energy $E$ of the network is defined as follows [14]:

$$E(V) = -.5 \sum_i \sum_j T_{ij} V_i V_j - \sum_i I_i V_i \quad (4.9)$$

The reason for trying both logistic sigmoid and tangent hyperbolic sigmoid is only to show that they both are valid and also lead to convergence.

The main idea behind solving the optimization problem is to formulate a suitable computational energy function $E(V)$ so that the lowest energy state would correspond to the required solution of $V$.

To map the economic dispatch problem by applying exterior penalty function for converting constrained optimization problems into unconstrained problems, we write the energy function as:

$$E = (A/2)\left(P_D + P_L - \sum_i P_i\right)^2 + (B/2)\sum_i \left(a_i + b_i P_i + c_i P_i^2\right)$$
$$+ (C/2)\sum_i h_i (P_i - P_{i\max})^2 + (D/2)\sum_i g_i (P_i - P_{i\min})^2 \qquad (4.10)$$

Where A, B, C and D are weighting factors.

$$h_i = \begin{cases} 0 & if\ P_i - P_{i\max} \le 0 \\ 1 & if\ P_i - P_{i\max} > 0 \end{cases}$$

$$g_i = \begin{cases} 0 & if\ P_i - P_{i\min} \ge 0 \\ 1 & if\ P_i - P_{i\min} < 0 \end{cases}$$

The power output value $P_i$ can be represented as

$$P_i = g_i(U_i)$$

We then find the mapping from Economic Dispatch to Hopfield Neural Network by comparing the coefficients of equations (4.9) and (4.10), and by representing $V_i$ *as* $P_i$. As a result one obtains :

$$T_{ii} = -A - Bc_i - Ch_i - Dg_i$$
$$T_{ij} = -A \qquad (4.11)$$
$$I_i = A(P_D + P_L) - .5Bb_i + Ch_i P_{i\max} + Dg_i P_{i\min}$$

## 4.3 Computational Result

In the following section, the simulation results of the economic power dispatch for some standard test systems are documented and analysed.

The application of the proposed algorithm was tested on Standard IEEE test system varying in size from 5-bus, 14-bus, and 57 -bus, and modified 30-bus. The basic characteristics of selected test systems are given in Table 4.1.

Table 4.1. Characteristics of the four IEEE standard test systems

| | | | | |
|---|---|---|---|---|
| Number of buses | 5 | 14 | 30 | 57 |
| Number of lines | 7 | 20 | 41 | 78 |
| Number of thermal generators | 3 | 2 | 6 | 4 |

The standard 5-bus system is taken from Saadat [3]. This system consists of three generators. These three generators are thermal and located on buses 1,2,and 3 respectively. The fuel cost function models for the three generators are assumed to be quadratic and expressed as

$$
\begin{aligned}
C_1 &= 200 + 7.0P_1 + 0.008P_1^2 \\
C_2 &= 180 + 6.3P_2 + 0.009P_2^2 \\
C_3 &= 140 + 6.8P_3 + 0.007P_3^2
\end{aligned}
$$

The limits of each generation is

$$
\begin{aligned}
10.0MW \le P_1 \le 85.0MW \\
10.0MW \le P_2 \le 80.0MW \\
10.0MW \le P_3 \le 70.0MW
\end{aligned}
$$

Power losses can be formulated by using Kron loss formula as follows

$$
P_L = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix} \begin{bmatrix} 0.0218 & 0.0093 & 0.0028 \\ 0.0093 & 0.0228 & 0.0017 \\ 0.0028 & 0.0017 & 0.0179 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}
+ \begin{bmatrix} 0.0003 & 0.0031 & 0.0015 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} + 0.00030523
$$

The line parameters of the 5-bus system and power load are shown in Appendix A.

The 14-bus system is an IEEE-AEP standard system [32]. Unlike the 5-bus system, the 14-system has two generators. They are located on bus 1 and 2 respectively. They are also thermal and the fuel cost models are given by

$$
\begin{aligned}
C_1 &= 50.607 + 10.662P_1 + 0.01165P_1^2 \\
C_2 &= 50.607 + 10.662P_2 + 0.01165P_2^2
\end{aligned}
$$

where $P_1$ *and* $P_2$ are in MW

The lower and upper bounds of each generator is expressed as

$$10.0MW \le P_1 \le 300.0MW$$
$$10.0MW \le P_2 \le 100.0MW$$

The power losses are:

$$P_L = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} 0.0245 & 0.0105 \\ 0.0105 & 0.0135 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} + \begin{bmatrix} 0.0018 & 0.0006 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} + 8.8422e-004$$

The 30-bus system is a modified IEEE 30-busbar system [27]. The system has six thermal generators. The six generators are located on buses 1, 2, 5, 8, 11 and 13 respectively. The line parameters are given in Appendix A as well.

The fuel cost models are expressed as follows;

$$C_1 = 2.00\,P_1 + 0.00375\,P_1^2$$
$$C_2 = 1.75\,P_2 + 0.0175\,P_2^2$$
$$C_3 = 1.00\,P_5 + 0.0625\,P_5^2$$
$$C_4 = 3.25\,P_8 + 0.00834\,P_8^2$$
$$C_5 = 3.00\,P_{11} + 0.025\,P_{11}^2$$
$$C_1 = 3.00\,P_{13} + 0.025\,P_{13}^2$$

The limits of generations are

$$50.0MW \le P_1 \le 200.0MW$$
$$20.0MW \le P_2 \le 80.0MW$$
$$15.0MW \le P_5 \le 50.0MW$$
$$10.0MW \le P_8 \le 35.0MW$$
$$10.0MW \le P_{11} \le 30.0MW$$
$$12.0MW \le P_{13} \le 40.0MW$$

The power losses for this system are expressed as

$$P_L = \begin{bmatrix} P_1 & P_2 & P_5 & P_8 & P_{11} & P_{13} \end{bmatrix} \begin{bmatrix} 0.0218 & 0.0102 & 0.0010 & -0.0010 & 0.0001 & 0.0027 \\ 0.0102 & 0.0187 & 0.0004 & -0.0015 & 0.0003 & 0.0031 \\ 0.0010 & 0.0004 & 0.0430 & 0.0134 & -0.0160 & -0.0108 \\ -0.0010 & -0.0015 & -0.0134 & 0.0224 & 0.0097 & 0.0051 \\ 0.0001 & 0.0003 & -0.0160 & 0.0097 & 0.0256 & -0.000 \\ 0.0027 & 0.0031 & -0.0108 & 0.0051 & -0.0000 & 0.0358 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_5 \\ P_8 \\ P_{11} \\ P_{13} \end{bmatrix}$$

$$+ \begin{bmatrix} -0.0003 & 0.0022 & -0.0057 & 0.0034 & 0.0016 & 0.0078 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_5 \\ P_8 \\ P_{11} \\ P_{13} \end{bmatrix} + 0.0014$$

In the 57-bus system, is an IEEE-AEP standard system [32]. There are four generators located on buses 1, 3, 8, and 12 respectively. The line parameters and load powers are given in Appendix A as well.

The fuels cost are expressed as follows;

$$
\begin{aligned}
C_1 &= 50.607 + 10.662P_1 + 0.01155P_1^2 \\
C_2 &= 50.607 + 10.662P_3 + 0.01155P_3^2 \\
C_3 &= 50.607 + 10.662P_8 + 0.01155P_8^2 \\
C_4 &= 50.607 + 10.662P_{12} + 0.01155P_{12}^2
\end{aligned}
$$

The power losses are expressed as

$$
P_L = \begin{bmatrix} P_1 & P_3 & P_8 & P_{12} \end{bmatrix}
\begin{bmatrix}
0.0113 & 0.0035 & -0.0035 & 0.0000 \\
0.0035 & 0.0115 & -0.0013 & -0.0015 \\
-0.0035 & -0.0013 & 0.0097 & -0.0036 \\
0.0000 & -0.0015 & -0.0036 & 0.0057
\end{bmatrix}
\begin{bmatrix} P_1 \\ P_3 \\ P_8 \\ P_{12} \end{bmatrix} +
$$

$$
\begin{bmatrix} 0.0028 & 0.0008 & -0.0014 & -0.0016 \end{bmatrix}
\begin{bmatrix} P_1 \\ P_3 \\ P_8 \\ P_{12} \end{bmatrix} + 0.0077
$$

The generation limits for this system are

$$
\begin{aligned}
20.0MW &\le P_1 \le 600.0MW \\
20.0MW &\le P_3 \le 600.0MW \\
20.0MW &\le P_8 \le 600.0MW \\
20.0MW &\le P_{12} \le 600.0MW
\end{aligned}
$$

The results are the optimal solution of each generator, and the obtained minimal cost for the different constant shape parameter ( $\lambda$ ) of the activation function built to the neuron. Moreover, the number of iteration for solving this problem is also recorded.

**5 bus test system.**

Table 4.2. The optimal conditions for each unit and the minimal cost obtained, for 5-bus test system.

| Unit | Optimal Condition (MW) | | |
|---|---|---|---|
| | Conventional Method (Lagrangian Multiplier) | HNN | |
| | | Logsig | Tansig |
| P1(MW) | 33.4701 | 33.2592 | 33.2592 |
| P2(MW) | 64.0974 | 64.1729 | 64.1729 |
| P3(MW) | 55.1011 | 54.9097 | 54.9097 |
| Cost ($/hr) | 1599.98 | 1597.50 | 1597.50 |
| Losses(MW) | 2.6686 | 2.3418 | 2.3418 |

Note :

Tansig : Tangent hyperbolic sigmoid activation function

Logsig : Logistic sigmoid activation function

The accuracy of the solution obtained by the algorithm when applying this proposed method to the 5-bus test system, is shown in Table 4.2. These results have been done by using the hyperbolic tangent function and logistic function with the same shape constant. In this case, we use the same initial guess for each neuron for these two types of activation function. The minimal cost for both activation function is exactly the same, but the number of iteration to reach convergence is different. The load demand for this system is 150 MW. The minimal cost function obtained by using a hyperbolic tangent activation function is the same as that obtained by using logistic activation function. In addition, can be seen that by using different activation functions, the number of iterations to obtain convergence when the tangent hyperbolic function is applied is about half of that required by applying logistic function (see Table 4.4). This is because of the form of these two functions are different as shown in the Fig. 4.1. By comparing with the conventional method (employing the Lagrangian Multipliers), this result is relatively similar. The losses obtained are slightly smaller than obtained by using conventional method

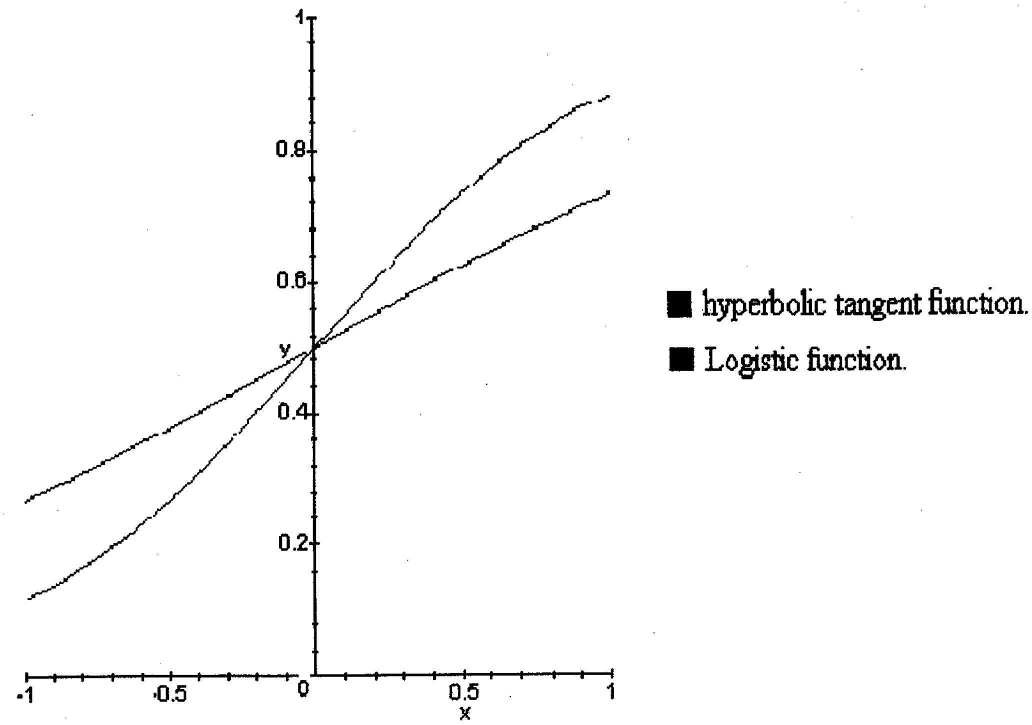


Figure 4.1 Two types of activation function

The choice of the initial guess for each neuron is a difficult task. Once we choose the initial guess of the neuron, whatever the value of the shape constant would be, the minimal conditions obtained are the same for two types of activation function. The difference is in the time consumed to reach convergence. By choosing different values of the initial guess, the results obtained shown in Table 4.3. In this table, two different starting point are chosen, and the optimal solution can be seen in case 1 and case 2.

Table 4.3. The optimal conditions obtained by using different initial guess for the neurons, for 5-bus test system.

| Unit | Optimal Condition (MW) | |
|---|---|---|
| | Case 1 | Case 2 |
| P1(MW) | 46.7763 | 46.7306 |
| P2(MW) | 57.4135 | 63.0069 |
| P3(MW) | 48.1765 | 42.7241 |
| Cost ($/hr) | 1600.20 | 1600.60 |
| Losses(MW) | 2.3663 | 2.4616 |

Table 4.4. Number of iterations required to reach optimal solution by varying the shape constant of sigmoid and by using different types of activation function for 5-bus test system.

| Lambda | No. of iteration | |
|---|---|---|
| | Logsig | Tansig |
| 0.0004 | 94 | 44 |
| 0.0005 | 74 | 34 |
| 0.0008 | 44 | 19 |
| 0.001 | 34 | 13 |

| | | |
|---|---|---|
| 0.0012 | 27 | 9 |
| 0.0015 | 20 | 10 |
| 0.0017 | 17 | 12 |
| 0.0018 | 16 | 16 |
| 0.0019 | 14 | 20 |
| 0.002 | 13 | 24 |
| 0.0024 | 9 | 76 |
| 0.0025 | 8 | 160 |
| 0.003 | 10 | N/A |
| 0.004 | 24 | |
| 0.005 | 160 | |
| 0.006 | N/A | |

From Table 4.4. can be seen that the number of iterations decreases from 44 to 9 when the value of lambda is increased from 0.0004 to 0.0012 by applying the tangent hyperbolic function. However, when the value of lambda is increased more beyond 0.0012, the number of iterations is also increased and the result is not applicable when the value of lambda is 0.003. At that value of lambda, the solution values undergo oscillations, and eventually give an unacceptable result. The same situation happened with the logistic activation function, where the number of iterations decreased from 94 to 8 when the value of lambda is increased from 0.0004 to 0.0025.

Table 4.4 shows the relation between the number of iterations and the value of lambda by using different types of activation function. It seems that the number of iterations for the two kinds of activation function will be the same when the value of lambda for logistic function is twice the value of lambda for hyperbolic tangent function. For instance, when the value of lambda is 0.001, by using tangent hyperbolic function the number of iterations is 13. On the other hand, by using logistic sigmoid function, the value of lambda 0.002 gives the number of iterations is equal to 13 also.

Generally, in this case the value of 0.0012 for lambda seems to give a good accuracy and high speed of convergence, especially with using tangent hyperbolic activation function.

**14-bus test system**

The result obtained is shown as :

Table 4.5. The optimal conditions for each unit and the minimal cost obtained, for 14-bus test system.

| Unit | Optimal Condition (MW) | | |
|---|---|---|---|
| | Conventional method | HNN | |
| | | Logsig | Tansig |
| P1(MW) | 171.6240 | 171.9580 | 171.9580 |
| P2(MW) | 100.0000 | 99.2010 | 99.2010 |
| Cost ($/hr) | 3456.92 | 3451.40 | 3451.40 |
| Losses(MW) | 12.6240 | 12.1590 | 12.1590 |

By using the same treatment as 5-bus system test above, the accuracy of the minimum cost obtained in the 14-bus system test is as shown in Table 4.5. The minimum cost obtained by this approach is little bit smaller than that obtained by conventional method. With the power demand at 259 MW, the losses obtained by this method are slightly lower than the conventional method.

The type of activation function also affects the speed of convergence. With the same initial guess for each neuron and the same shape constant, the number of iterations required when using the hyperbolic tangent is lower than that by using logistic activation function. On the other hand, the minimal cost for both activation functions are the same, as shown in Table 4.5. Table 4.6 shows the values for the other cases when the different initial guesses to the neuron are given. In these cases, the optimal conditions and the costs obtained are different from that shown in Table 4.5.

From Table 4.7, can be seen that the tangent hyperbolic activation function gives the fastest solution when the value of lambda is 0.0008; and the number of iterations is 6. Whereas, the logistic activation function gives the fastest solution with the number of iterations is 7 and the value of lambda is 0.0015.

Table 4.6. The optimal conditions obtained by using different initial guess for the neurons, for 14-bus test system.

| Unit | Optimal Condition (MW) | |
|---|---|---|
| | Case 1 | Case 2 |
| P1(MW) | 175.3956 | 178.4473 |
| P2(MW) | 95.9204 | 93.0115 |
| Cost ($/hr) | 3459.6 | 3467.3 |
| Losses(MW) | 12.3160 | 12.4588 |

Table 4.7. Number of iterations required to reach the optimal solution by varying the shape constant of sigmoid and by using different types of activation function for 14-bus test system.

| Lambda | No. of iteration | |
|---|---|---|
| | Logsig | Tansig |
| 0.0004 | 60 | 26 |
| 0.0005 | 46 | 19 |
| 0.0006 | 37 | 14 |
| 0.0007 | 31 | 10 |
| 0.00075 | 28 | 7 |
| 0.0008 | 26 | 6 |

| | | |
|---|---|---|
| 0.0009 | 22 | 11 |
| 0.001 | 19 | 13 |
| 0.0012 | 14 | 26 |
| 0.0014 | 10 | 76 |
| 0.0015 | 7 | 252 |
| 0.002 | 13 | N/A |
| 0.0025 | 34 | |
| 0.003 | 252 | |
| 0.004 | N/A | |

For the value of lambda greater then 0.0015 with tansig activation function, and lambda greater than 0.003 with logsig activation function do not give feasible result. The result given at those values will alternate and end up at the upper and lower limits.

**30-bus system test**

Table 4.8 shows the performance for the 30-bus test system. The optimal solution by using tangent hyperbolic function and logistic function is reached. The proposed method gives an amount of the minimal cost which is little bit lower than that obtained using the conventional method. With the power demand 283.4 MW; the minimal cost obtained is $801.9642 /hour and the losses are 9.3305 MW. The other cases shown in Table 4.9 by changing the initial guess for the neurons.

Table 4.8. The optimal conditions for each unit and the minimal cost obtained, for 30-bus test system.

| Unit | Optimal Condition (MW) | | |
|---|---|---|---|
| | Conventional method | HNN | |
| | | Logsig | Tansig |
| P1(MW) | 176.9754 | 176.2093 | 176.2093 |
| P2(MW) | 48.2544 | 48.7806 | 48.7806 |
| P5(MW) | 20.9630 | 19.8327 | 19.8327 |
| P8(MW) | 22.4100 | 21.5560 | 21.5560 |
| P11(MW) | 12.3962 | 12.5991 | 12.5991 |
| P13(MW) | 12.0000 | 13.7528 | 13.7528 |
| Cost ($/hr) | 802.68 | 801.9642 | 801.9642 |
| Losses(MW) | 9.5990 | 9.3305 | 9.3305 |

Table 4.9. The optimal conditions obtained by using different initial guess for the neurons, for 30-bus test system.

| Unit | Optimal Condition (MW) | |
|---|---|---|
| | Case 1 | Case 2 |
| P1(MW) | 158.3727 | 155.1983 |
| P2(MW) | 49.1785 | 55.0919 |
| P5(MW) | 29.5324 | 25.0451 |

| | | |
|---|---|---|
| P8(MW) | 24.6696 | 25.0378 |
| P11(MW) | 14.7974 | 16.0265 |
| P13(MW) | 14.8564 | 15.0169 |
| Cost ($/hr) | 808.4366 | 806.2851 |
| Losses(MW) | 8.0069 | 8.0165 |

Table 4.10. Number of iterations required to reach the optimal solution by varying the shape constant of sigmoid and by using different types of activation function for 30-bus test system.

| Lambda | No. of iteration | |
|---|---|---|
| | Logsig | Tansig |
| 0.0001 | 289 | 142 |
| 0.0004 | 68 | 31 |
| 0.0005 | 53 | 24 |
| 0.0006 | 44 | 19 |
| 0.0007 | 37 | 15 |
| 0.00075 | 34 | 13 |
| 0.0008 | 31 | 12 |
| 0.0009 | 27 | 10 |
| 0.001 | 24 | 7 |
| 0.0012 | 19 | 8 |

| | | |
|---|---|---|
| 0.0014 | 15 | 11 |
| 0.0015 | 13 | 15 |
| 0.0017 | 11 | 25 |
| 0.002 | 7 | 93 |
| 0.0025 | 10 | N/A |
| 0.003 | 17 | |
| 0.004 | 99 | |
| 0.005 | N/A | |

For the 30-bus test system, in order to reduce the number of iteration, lambda is increased from 0.0001 until 0.005 as shown in Table 4.10. Lower than 0.0001, the same result could be reached, but it needs too much time to get convergence. The fastest solution could be achieved when the value of lambda is equal to 0.002 by using the logistic transfer function, and lambda equal to 0.001 for the tangent hyperbolic function. In this case, by using tansig activation function, we could get a lower number of iteration to reach the optimal solution than that by using logsig. When we set lambda greater then 0.002 for tangent hyperbolic function and greater than 0.004 for logistic function, the result will an unfeasible solution.

**57-bus test system**

Table 4.11. The optimal conditions for each unit and the minimal cost obtained, for 57-bus test system.

| Unit | Optimal Condition (MW) | | |
|---|---|---|---|
| | Conventional method | HNN | |
| | | Logsig | Tansig |
| P1(MW) | 295.5308 | 295.6114 | 295.6114 |
| P3(MW) | 292.6571 | 293.0090 | 293.0090 |
| P8(MW) | 339.9919 | 338.5716 | 338.5716 |
| P12(MW) | 346.0974 | 346.3031 | 346.3031 |
| Cost ($/hr) | 18505 | 18490 | 18490 |
| Losses(MW) | 23.4774 | 22.6951 | 22.6951 |

The minimal cost has been reached by applying this method on the 57-bus test system as shown in Table 4.11. Using the Hopfield method gives a slightly lower amount of minimal cost than that obtained by using the conventional method. The power demand given is 1250.8 MW. Using the tangent hyperbolic function gives faster convergence than that by using logistic function. The same problem as when applying this method to the other test system is getting trap to the local minimal when choosing the different initial guess for the neurons, as shown in Table 4.12.

Table 4.12. The optimal conditions obtained by using different initial guess for the neurons, for 57-bus test system.

| Unit | Optimal Condition (MW) | |
|---|---|---|
| | Case 1 | Case 2 |
| P1(MW) | 292.8473 | 290.8870 |
| P3(MW) | 267.9581 | 265.9594 |
| P8(MW) | 342.9450 | 343.5210 |
| P12(MW) | 368.1548 | 371.3325 |
| Cost ($/hr) | 18,507.00 | 18,511.00 |
| Losses(MW) | 21.1052 | 20.8999 |

Table 4.13. Number of iterations required to reach optimal solution by varying the shape constant of sigmoid and by using different type of activation function for 57-bus test system.

| Lambda | No. of iteration | |
|---|---|---|
| | Logsig | Tansig |
| 0.0004 | 573 | 299 |
| 0.0008 | 284 | 147 |
| 0.0012 | 189 | 96 |
| 0.0016 | 147 | 71 |

| | | |
|---|---|---|
| 0.002 | 116 | 55 |
| 0.0025 | 92 | 43 |
| 0.003 | 76 | 35 |
| 0.0035 | 64 | 29 |
| 0.004 | 55 | 25 |
| 0.0045 | 48 | 21 |
| 0.005 | 43 | 18 |
| 0.0055 | 39 | 16 |
| 0.006 | 35 | 14 |
| 0.0065 | 32 | 12 |
| 0.007 | 29 | 11 |
| 0.0075 | 27 | 9 |
| 0.008 | 25 | 8 |
| 0.0085 | 23 | 6 |
| 0.009 | 21 | 4 |
| 0.01 | 18 | 8 |
| 0.02 | 8 | N/A |
| 0.03 | 34 | |
| 0.035 | 226 | |
| 0.04 | N/A | |

The system finds a good solution for the minimal cost when setting the value of lambda within the range 0.0004 to 0.01 for the hyperbolic tangent function, and within the range 0.0004 to 0.035 for the logistic function.

From Table 4.13, it is seen that using the value of lambda equal to 0.009 for applying the tangent hyperbolic function and lambda equal to 0.02 for logistic function, the fastest solution can be reached. This Table shows that for the same lambda, the number of iterations required to get an optimal solution using the tansig function is always almost half of that number reached using logsig activation function.

### 4.3.1 Comparison between the proposed method and the conventional method.

Comparing our results with those obtained by using conventional optimization method is shown in Table 4.14. It can be seen that the results have been obtained by using four test system are slightly different with the results when we use the conventional method (Lagrangian Multiplier). This means that the result obtained by this thesis method is consistent with the result obtained by the conventional method (Lagrange Multiplier method).

Table 4.14. The comparison of the optimal solution by using the proposed method and Lagrange multiplier method.

| NUMBER OF BUSES | COST ($/HR) | | | ERROR (%) |
|---|---|---|---|---|
| | PROPOSED | | CONVEN-TIONAL | |
| | LOGSIG | TANSIG | | |
| 5 | 1597.50 | 1597.50 | 1599.9 | 0.15 |
| 14 | 3451.40 | 3451.40 | 3456.92 | 0.16 |
| 30 | 801.9642 | 801.9642 | 802.68 | 0.089 |
| 57 | 18490 | 18490 | 18505 | 0.08 |

From Table 4.14 can be concluded that the proposed technique performs well with these standard systems i.e. our results are almost the same as those obtained by means of known conventional methods (Lagrangian Multiplier method). On the other hand this proposed method still less significant than the other exist method because of its dependency on the initial guess. Therefore, this proposed method needs to be improved in order to overcome that difficulty mentioned above. In fact, the proposed method, using the hyperbolic tangent and logistic activation function gives the same results when the same values of initial guess for the neurons is applied. The time required to get convergence whatever the value of lambda used by applying the logistic function is twice that needed when applying tangent hyperbolic function.

# Chapter 5

# ACTIVE SECURITY-CONSTRAINED DISPATCH USING HOPFIELD NEURAL NETWORK

## 5.1 BACKGROUND

Security Constrained Dispatch is defined as an Optimal Power Flow problem, in which the objective function is the total cost of generations and the security constraints are placed on the bus voltage magnitudes, phase angles and the generated reactive powers. The aim of using Security Constrained Dispatch is to find the optimal solution that will minimize the production cost that satisfies all of the constraints as fast as possible [46].

In a Security Constrained Dispatch developed by Salgado et al. [12], the dependent variable constraints are on the bus voltage magnitude and the generated reactive powers. The results show that the effect of the bus voltage magnitude and the generated reactive powers on the real power generated is excessively weak.

Considering the fact that there are strong relations between the power generated, P, and the bus voltage angle, $\delta$, and also between the generated reactive power, Q, and the bus voltage magnitude, V, and that the couplings between P-V and Q-$\delta$ are very weak. Therefore, the Optimal Power Flow solution may be divided into a P-$\delta$ optimization module and a Q-V optimization module [46].

According to these principles, Sjoholm and Boye [46], have proposed a new method involving modifying the constraints, i.e., by replacing the bus voltage magnitude and generated reactive power constraints with the constraints on the bus voltage angles.

When the generating units are located at a great distance from load centres, the production cost minimization may result in the transfer of a large amount of active power through the transmission lines, resulting in a low voltage values at load buses.

In order to improve the shortcomings mentioned above, Kirschen and Van Meeteren [24] suggested the MW/Voltage control, by introducing voltage constraints directly into the active optimization process.

In this thesis, the method proposed by Sjoholm and Boye [46] is modified, by taking into account a generated active power increment on the swing bus for the inequality and equality constraints. In addition, the maximum and minimum values of load bus voltage magnitudes are considered as suggested by [24]. These modification could result in more accurate solution.

## 5.2 ACTIVE SECURITY CONSTRAINED DISPATCH ALGORITHM

For active power generation cost minimization, the cost function is given by the total sum of generator fuel costs [12, 27, 46].

$$C(P_{Gi}) = \sum_{i=1}^{N_G} \left(a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) \qquad (5.1)$$

where

$N_G$ = set of indices of generator buses including the swing busbar.

In order to have the objective function in terms decision variables as constraints, the equation (5.1) is approximated using the 2nd order Taylor series expansion as [12, 27, 46]

$$C\left(P_{Gi} + \Delta P_{Gi}\right) = \sum_{i=1}^{N_G} \left(a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) + \sum_{i=1}^{N_G} \left(b_i + 2c_i P_{Gi}\right)\Delta P_{Gi} + c_i \Delta P_{Gi}^2$$

The incremental generation cost function expanded in a second order Taylor series

$$F_C(\Delta P_i) = \sum_{i=1}^{N_G} (b_i + 2c_i P_i)\Delta P_i + c_i \Delta P_i^2 \qquad (5.2)$$

Or, in matrix form [12, 27, 46]:

$$F_C(\Delta P_{S,PV}) = [A_1]\Delta P_{S,PV} + \Delta P_{S,PV}^T [A_2]\Delta P_{S,PV} \qquad (5.3)$$

where

$[A_1]$ : $(1xN_G)$ row vector with components $(b_i + 2c_i P_i)$

$[A_2]$ : $(N_G x N_G)$ diagonal matrix with components $c_i$

$\Delta P_{S,PV}$ : $(N_G x1)$ column vector of the generated active power increments.

The optimization should satisfy many constraints. In this case, the equality and inequality constraints are included.

**Equality Constraint**

The equality constraint can be derived from the modification of the relation among variables in load flow algorithm as follows:

The nodal difference equation for the busbar powers can be expressed in matrix form as [3]:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \left|\dfrac{\Delta V}{V}\right| \end{bmatrix} \tag{5.4}$$

The slack-bus phase angle and voltage magnitudes, the PQ buses active and reactive powers and PV-buses voltage magnitude are assumed to remain constant

$$\Delta\delta_S = 0,\ \Delta V_S = 0,\ \Delta P_{PQ} = 0,\ \Delta Q_{PQ} = 0, \text{ and } \Delta V_{PV} = 0 \tag{5.5}$$

By using the condition of Eq. (5.5) in Eq. (5.4) yields the following mutual dependencies among real generating powers [27] :

$$\begin{bmatrix} \Delta P_S \\ \Delta P_{PV} \end{bmatrix} = [H] \begin{bmatrix} \Delta\delta_{PV} \\ \Delta\delta_{PQ} \end{bmatrix}$$

where

$\Delta P_S$ : the power generated increment at slack bus

$\Delta P_{PV}$ : the power generated increment at the $PV$ bus

or

$$\Delta P_S = [H_1] \begin{bmatrix} \Delta\delta_{PV} \\ \Delta\delta_{PQ} \end{bmatrix} \quad \text{and} \quad [\Delta P_{PV}] = [H_2] \begin{bmatrix} \Delta\delta_{PV} \\ \Delta\delta_{PQ} \end{bmatrix}$$

yields:

$$\begin{bmatrix} \Delta\delta_{PV} \\ \Delta\delta_{PQ} \end{bmatrix} = [H_2]^{-1}[\Delta P_{PV}]$$

Thus

$$\Delta P_S = [H_1][H_2]^{-1}[\Delta P_{PV}]$$

Finally we have

$$[1 \quad -S][\Delta P_{S.PV}] = 0 \qquad (5.6)$$

where

$$S = [H_1][H_2]^{-1}$$

The above equation represents the linearized active power balance equation.

**Inequality Constraints**

The first constraints are on the upper and lower bounds of the power generating units. The generating constraints give the maximum and minimum generating capacity, outside of which it is not feasible to generate due to technical or economic reasons. The generating limits are expressed as [12, 27, 46]

$$\Delta P_{Gi\min} \le \Delta P_{Gi} \le \Delta P_{Gi\max} \qquad (5.7)$$

The second sets of constraints are the upper and lower bounds of voltage phase angles. The voltage phase angles depend on the generating units.

From the power flow problem, the relation between the deviation of active and reactive power generating and the deviation of voltage phase angle can be reformulated as [55]:

$$\begin{bmatrix} \Delta P_S \\ \Delta P_{PV} \\ \Delta Q_S \\ \Delta Q_{PV} \end{bmatrix} = [J] \begin{bmatrix} \Delta\delta_{PV} \\ \Delta\delta_{PQ} \\ \Delta V_{PQ} \end{bmatrix} \qquad (5.7)$$

By considering the fast decoupled algorithm as explained in Chapter 2, the successive phase angle changes are [3]

$$\Delta\delta = -[B']^{-1} \frac{\Delta P}{V} \qquad (5.8)$$

So, the constraints on $\delta$ can be written as follows:

$$\Delta\delta_{\min} \le \Delta\delta_{PV,PQ} = -[B']^{-1} \frac{\Delta P_{S,PV}}{V} \le \Delta\delta_{\max} \qquad (5.9)$$

where

$\Delta P_{S,PV}$ : the power generated increments at slack and $PV$ bus

In this proposed method, the deviation of power of the generating units are included for the swing bus and voltage controlled buses, as shown in (5.9), and the deviation of voltage phase angles are included at the voltage control buses and load buses.

The third sets of constraints are the upper and lower bounds of the voltage magnitude. Consideration of voltage magnitude constraints in the active optimization requires knowledge of a linearized relation between the incremental voltage magnitude at bus $i$ and the incremental changes in the active controls [24]. According to [24], the effect of the active power injections on the voltage magnitude is described by:

$$\Delta V_{PQ} = \left[L^{-1}(-M)H^{-1}\right]\Delta P_{S,PV} \quad (5.10)$$

By replacing the matrices H and L with the matrices B' and B" of the fast-decoupled power flow, we have [24]

$$\Delta V_{PQ} = \left[(B'')^{-1}(-M)(B')^{-1}\right]\Delta P_{S,PV} \quad (5.11)$$

The Voltage constraint can be expressed as

$$\Delta V_{\min} \le \Delta V_{PQ} = \left[(B'')^{-1}(-M)(B')^{-1}\right]\Delta P_{S,PV} \le \Delta V_{\max} \quad (5.12)$$

## 5.3 ACTIVE SECURITY-CONSTRAINT DISPATCH USING HOPFIELD NEURAL NETWORK

The objective of the optimization problem is the incremental cost function as expressed in equation (3)

Min $\quad F_C(\Delta P_{S,PV}) = [A_1]\Delta P_{S,PV} + \Delta P_{S,PV}^T [A_2]\Delta P_{S,PV}$

Subject to

Equality constraint

$$[1 \quad -S]\Delta P_{S,PV} = 0$$

And Inequality constraints

$$\Delta P_{Gi\min} \le \Delta P_{Gi} \le \Delta P_{Gi\max}$$

$$\Delta\delta_{\min} \le \Delta\delta_{PV,PQ} = -\left[B'\right]^{-1} \frac{\Delta P_{S,PV}}{V} \le \Delta\delta_{\max}$$

$$\Delta V_{\min} \le \Delta V_{PQ} = \left[(B'')^{-1}(-M)(B')^{-1}\right]\Delta P_{S,PV} \le \Delta V_{\max}$$

The cost objective function and the demand requirement constraint are combined to produce an augmented objective function given by

$$\phi(x,K)=K_0\left([A_1]\Delta P_{S,PV}+\Delta P_{S,PV}^T[A_2]\Delta P_{S,PV}\right)+K_1\sum_{i=1}^{m}\left\{\max\left[0,h_i\left(\Delta P_{S,PV}\right)\right]\right\}^2 + K_2\sum_{j=1}^{n}\left\{g_j\left(\Delta P_{S,PV}\right)\right\}^2 \qquad (5.13)$$

where

$K_0, K_1$ *and* $K_2$ are weighting factors indicating the relative importance of each term.

$h_i(\Delta P_{S,PV})$ is the inequality constraints as a function of power generated increments.

$g_j(\Delta P_{S,PV})$ is the equality constraints as a function of power generated increments.

Then the augmented objective function is mapped into the Hopfield network.

The Hopfield network is created with N neurons. The output value of each neuron represents the generated increment of each unit.

The sigmoid function of each neuron is modified to limit the output value of each neuron to lie between the minimum and maximum generated increment of each unit. The first form used by Park et al. [14] is a hyperbolic tangent function given by

$$\Delta P_{S,PV}=\left(\Delta P_{S,PV}^{\max i}-\Delta P_{S,PV}^{\min i}\right)\left(1+\tanh\left(\lambda U_i\right)\right)/2+\Delta P_{S,PV}^{\min i} \qquad (5.14)$$

And logistic function given by

$$\Delta P_{S,PV}=\left(\Delta P_{S,PV}^{\max i}-\Delta P_{S,PV}^{\min i}\right)\left(\frac{1}{1+\exp\left(-\lambda U_i\right)}\right)+\Delta P_{S,PV}^{\min i} \qquad (5.15)$$

### 5.4. Numerical Results

A Matlab program has been written to implement the modified Hopfield neural network method to solve the four test systems described in this thesis. These systems are a 5-bus test system from Saadat [3], an IEEE-AEP Standard 14-busbar system [32], and a modified IEEE 30-busbar system [27], and IEEE-AEP 57-busbar test system [32].

The parameter $\lambda$ determines the shape of the sigmoid function and the rate at which the output, $V_i$, is updated with respect to a change in input, $u_i$. The proper value of $\lambda$ depends on the data being processed. The proper value has been chosen by trying different values of $\lambda$ for each problem instance.

The data used in this chapter are the same as the data that was used in chapter 4.
The result for both systems is presented in Tables 5.1 to 5.16.

**30-bus test system**

The results obtained by applying this method for the 30-bus system test are shown in Tables 5.1, 5.2, 5.3 and 5.4. This result has been compared with some other results by using the Gradient Projection method from [27], [12], [46] and [47]. They used the same objective function and data but different constraints. The inequality constraints used by the first reference [27] is the limitation of power generations. The second [12] used the upper and lower bounds of active power generating units, reactive power generating units and voltage magnitude of load buses as inequality constraints. The third [46], as the inequality constraints are the upper and lower bounds of power generation and voltage phase angles, and the last [47], they considered to be the active power flows along lines. The minimal costs have been reached by those methods with the load 283.4 MW are \$804.853/hr, \$806.88/hr, \$820.89/hr and \$821.57/hr respectively.

Using the proposed method, the constraints are the upper and lower bounds of power generating units, voltage phase angles and voltage magnitudes. The first step is a power flow unoptimizes calculation to determine the initial conditions from which the optimization will begin. Afterward, the initial guess for the generated power increment is chosen, and then run the program to get the optimal solutions, i.e. minimal cost of each generating unit, voltage phase angle and voltage magnitude on each bus.

One of the problems of this method for computer applications is the need of an initial feasible point. For the first case, a certain initial guess for increment of power generation is chosen, and the solutions achieved are written in Table 5.1 and 5.2. for a minimal cost obtained and voltage phase angles and voltage magnitude by applying the hyperbolic tangent and logistic activation function. With the same power demand as the other methods above, the minimal cost reached for these two types of activation function are the same, i.e. \$ 804/hour as shown in Table 5.1. The differences between them are on the time to convergence. This amount is lower than those obtained by using the three previous methods [12, 46, 47]. The power losses obtained by using this method are 8.8 MW, which is lower than that obtained by Lee's [27], i.e. 10.154 MW, and 1.3 MW higher than that obtained by Yalcinoz's [47], i.e. 7.532 MW. The active security dispatch method is also checked for the voltage magnitudes and voltage angles. The results are shown in Tables 5.2. The lowest value of voltage magnitude on the load bus is 0.994 p.u. on bus 3, and the highest is 1.0608 p.u. on bus 12. The bus voltage phase angles yielded are also good, i.e. vary between the minimum –11.266 degrees and the maximum 0.00 degrees. Both voltage magnitudes and phase angles are consistent with other results. The overall time needed to reach the convergence is 1.43 second. Therefore, the results are reasonable.

Table 5.3 and 5.4 show the results where a different initial guess is used. In this case, the minimal cost obtained is $ 809.645/hour, which is still consistent with those obtained by three previous methods In addition, this result gives the lower losses than in Lee's method. The voltage magnitudes are also inside the bounds. The lowest value is 0.991 p.u. on bus 3 and the highest is 1.0820 p.u. on bus 12. The bus voltage phase angles lie between the minimum of -11.266 degrees and the maximum of 0.00 degrees. From these results it is seen that this method is consistent with the other methods.

Table 5.1 The optimal conditions of power generated and minimal cost obtained by using hyperbolic tangent and logistic activation function in the first case for 30-bus test system.

| Variable | Limits | | Optimal Condition | |
|---|---|---|---|---|
| | Lower | Upper | Tansig | Logsig |
| P1 | 50.0 | 200.0 | 169.1185 | 169.1262 |
| P2 | 20.0 | 80.0 | 48.8350 | 48.8375 |
| P5 | 15.0 | 50.0 | 25.8993 | 25.8996 |
| P8 | 10.0 | 35.0 | 20.1401 | 20.1400 |
| P11 | 10.0 | 30.0 | 13.5199 | 13.5200 |
| P13 | 12.0 | 40.0 | 14.7000 | 14.7000 |
| Cost ($/hour) | | | 803.9797 | 804.0152 |
| Losses (MW) | | | 8.8128 | 8.8233 |

Table 5.2. The bus phase angles and voltage magnitudes obtained in the first case for 30-bus test system.

| BUS NO. | DELTA (degrees) | VOLTAGE (p.u.) |
|---|---|---|
| 1 | 0.0000 | 1.0600 |
| 2 | -2.8649 | 1.0430 |
| 3 | -7.3207 | 0.9940 |
| 4 | -6.0802 | 1.0417 |
| 5 | -7.8707 | 1.0100 |
| 6 | -6.2100 | 1.0170 |
| 7 | -6.1536 | 1.0186 |
| 8 | -6.2180 | 1.0100 |
| 9 | -6.2521 | 1.0535 |
| 10 | -8.4780 | 1.0480 |
| 11 | -4.8749 | 1.0820 |
| 12 | -7.4425 | 1.0608 |
| 13 | -6.4275 | 1.0710 |
| 14 | -8.5540 | 1.0460 |
| 15 | -8.6680 | 1.0410 |
| 16 | -8.2650 | 1.0480 |
| 17 | -8.6200 | 1.0430 |
| 18 | -9.2910 | 1.0310 |
| 19 | -9.4720 | 1.0290 |
| 20 | -9.2800 | 1.0330 |
| 21 | -8.9490 | 1.0350 |
| 22 | -8.9450 | 1.0360 |
| 23 | -9.1460 | 1.0310 |
| 24 | -9.4440 | 1.0250 |
| 25 | -9.4330 | 1.0210 |
| 26 | -9.8500 | 1.0030 |
| 27 | -9.1690 | 1.0270 |
| 28 | -5.5784 | 1.0096 |
| 29 | -10.3900 | 1.0070 |
| 30 | -11.2660 | 0.9960 |

Table 5.3 The optimal conditions of power generated and minimal cost obtained in the second case when the initial guess for the neurons is changed for 30-bus test system.

| Variable | Limits | | Optimal Condition | |
|---|---|---|---|---|
| | Lower (MW) | Upper (MW) | Tansig (MW) | Logsig (MW) |
| P1 | 50.0 | 200.0 | 150.6683 | 150.6762 |
| P2 | 20.0 | 80.0 | 59.9966 | 59.9983 |
| P5 | 15.0 | 50.0 | 22.9995 | 22.9998 |
| P8 | 10.0 | 35.0 | 22.0000 | 22.0000 |
| P11 | 10.0 | 30.0 | 15.0099 | 15.0100 |
| P13 | 12.0 | 40.0 | 20.7299 | 20.7299 |
| Cost ($/hour) | | | 809.6447 | 809.6769 |
| Losses (MW) | | | 8.0044 | 8.0142 |

Table 5.4. The bus phase angles and voltage magnitudes obtained in the second case when the initial guess for the neurons is changed for 30-bus test system

| BUS NO. | DELTA (degrees) | VOLTAGE (p.u.) |
|---|---|---|
| 1 | 0.0000 | 1.0600 |
| 2 | -2.5308 | 1.0430 |
| 3 | -6.3886 | 0.9991 |
| 4 | -5.6314 | 1.0048 |
| 5 | -7.7617 | 1.0100 |
| 6 | -5.9645 | 1.0120 |
| 7 | -5.9684 | 1.0049 |
| 8 | -5.9815 | 1.0100 |
| 9 | -6.3342 | 1.0540 |
| 10 | -8.4780 | 1.0480 |
| 11 | -4.7928 | 1.0820 |
| 12 | -7.6683 | 1.0610 |
| 13 | -6.2017 | 1.0710 |
| 14 | -8.5540 | 1.0460 |
| 15 | -8.6680 | 1.0410 |
| 16 | -8.2650 | 1.0480 |
| 17 | -8.6200 | 1.0430 |
| 18 | -9.2910 | 1.0310 |
| 19 | -9.4720 | 1.0290 |
| 20 | -9.2800 | 1.0330 |
| 21 | -8.9490 | 1.0350 |
| 22 | -8.9450 | 1.0360 |
| 23 | -9.1460 | 1.0310 |
| 24 | -9.4440 | 1.0250 |
| 25 | -9.4330 | 1.0210 |
| 26 | -9.8500 | 1.0030 |
| 27 | -9.1690 | 1.0270 |
| 28 | -5.6189 | 1.0160 |
| 29 | -10.3900 | 1.0070 |
| 30 | -11.2660 | 0.9960 |

Based on the previous comparison, the proposed method can be considered to be an alternative method for solving the optimization problem. In the rest of this chapter, the proposed method is applied to the 14-bus, 57-bus and 5-bus test systems.

**5-bus test system**

Tables 5.5 and 5.6 show the results by applying this method to the 5-bus test system. The minimal cost obtained using the two kinds of activation function are exactly the same amount, i.e. \$ 1597.50/hour for the power demand 150 MW. The lowest voltage magnitude at the load bus has been reached is 0.9798 p.u. at bus 5, and the highest is 1.0162 p.u. at bus 4. The voltage phase angles are within the maximum 0.00 degrees and –2.975 degrees. Tables 5.7 and 5.8 show the results for the optimal conditions and the voltage magnitudes and phase angles obtained when we start with the other initial guess for the neurons. The minimal cost is \$ 1600.50/hour. The different from the first case is only 0.19%. The new voltage magnitudes at the load buses within the maximum 1.0168 p.u. and minimum 0.9836 p.u. The voltage phase angles reached are within the range –3.351 degrees and 0.00 degree. The overall time needed for solving this problem is 0.22 second.

Table 5.5 The optimal conditions of power generated and minimal cost obtained by using hyperbolic tangent and logistic activation function for 5-bus test system.

| Variable (MW) | Limits | | Optimal Condition | |
|---|---|---|---|---|
| | Lower (MW) | Upper (MW) | Tansig (MW) | Logsig (MW) |
| P1 | 10 | 85 | 33.2529 | 33.2530 |
| P2 | 10 | 80 | 64.5999 | 64.6000 |
| P3 | 10 | 70 | 54.4999 | 54.5000 |
| Cost ($/hr) | | | 1597.50 | 1597.50 |
| Losses (MW) | | | 2.3528 | 2.3529 |

Table 5.6. The bus phase angles and voltage magnitudes obtained in the first case for 5-bus test system.

| BUS NO. | DELTA (degree) | VOLTAGE (p.u.) |
|---|---|---|
| 1 | 0.0000 | 1.0600 |
| 2 | -0.5227 | 1.0450 |
| 3 | -0.8053 | 1.0300 |
| 4 | -1.4808 | 1.0162 |
| 5 | -2.9750 | 0.9798 |

Table 5.7 The optimal conditions of power generated and minimal cost obtained in the second case when the initial guess for the neurons is changed for 5-bus test system.

| Variable (MW) | Limits | | Optimal Condition | |
|---|---|---|---|---|
| | Lower (MW) | Upper (MW) | Tansig (MW) | Logsig (MW) |
| P1 | 10 | 85 | 47.5029 | 47.5029 |
| P2 | 10 | 80 | 55.6899 | 55.6900 |
| P3 | 10 | 70 | 49.1599 | 49.1600 |
| Cost ($/hr) | | | 1600.5 | 1600.5 |
| Losses (MW) | | | 2.3527 | 2.3529 |

Table 5.8 The bus phase angles and voltage magnitudes obtained in the second case for 5-bus test system

| BUS NO. | PHASE ANGLE (degree) | VOLTAGE (p.u.) |
|---|---|---|
| 1 | 0.0000 | 1.0600 |
| 2 | -0.8944 | 1.0450 |
| 3 | -1.2913 | 1.0300 |
| 4 | -1.9536 | 1.0168 |
| 5 | -3.3510 | 0.9836 |

**14-bus test system**

When applying this method to the 14-bus test system with the power load 259 MW, the amount of minimal cost reached is $3451.20/hour by using the hyperbolic tangent activation function or the logistic activation function as shown in Table 5.9. The voltage magnitudes have the minimum value 0.9754 p.u. at bus 3 by using tangent hyperbolic function and maximum value 1.0900 p.u. at bus 8. The voltage phase angles obtained are also consistent with other results. The maximum has been reached is 0.00 degrees and the minimum is –15.994 degrees as given in Table 5.10. By giving different values of initial point for the increment of power generating, the results as shown in Tables 5.11 and 5.12. The minimal cost has been reached is $ 3515.20/hour for either by using hyperbolic tangent or logistic activation function. Both the voltage magnitudes and voltage phase angles are consistent with other results. The overall time needed to solve this problem is 0.77 second.

Table 5.9 The optimal conditions of power generated and minimal cost obtained by using hyperbolic tangent and logistic activation function in the first case for 14-bus test system.

| Variable | Limits | | Optimal Condition | |
|---|---|---|---|---|
| | Lower (MW) | Upper (MW) | Tansig (MW) | Logsig (MW) |
| P1 | 10.00 | 300.00 | 171.8159 | 171.8164 |
| P2 | 10.00 | 100.00 | 99.3400 | 99.3400 |
| Cost ($/hr) | | | 3451.20 | 3451.20 |
| Losses (MW) | | | 12.1559 | 12.1564 |

Table 5.10. The bus phase angles and voltage magnitudes obtained in the first case for 14-bus test system

| BUS NO. | DELTA (degree) | VOLTAGE (p.u.) |
|---|---|---|
| 1 | 0.0000 | 1.0600 |
| 2 | -3.3849 | 1.0450 |
| 3 | -12.1799 | 0.9754 |
| 4 | -9.7441 | 0.9810 |
| 5 | -6.8035 | 1.0482 |
| 6 | -14.2500 | 1.0700 |
| 7 | -13.3620 | 1.0610 |
| 8 | -13.3620 | 1.0900 |
| 9 | -14.9400 | 1.0550 |
| 10 | -15.1030 | 1.0500 |
| 11 | -14.8070 | 1.0570 |
| 12 | -15.0960 | 1.0550 |
| 13 | -15.1690 | 1.0490 |
| 14 | -15.9940 | 1.0310 |

Table 5.11 The optimal conditions of power generated and minimal cost obtained in the second case when the initial guess for the neurons is changed for 14-bus test system.

| Variable | Limits | | Optimal Condition | |
|---|---|---|---|---|
| | Lower (MW) | Upper (MW) | Tansig (MW) | Logsig (MW) |
| P1 | 10.00 | 300.00 | 194.4659 | 194.4665 |
| P2 | 10.00 | 100.00 | 77.7998 | 77.7999 |
| Cost ($/hr) | | | 3515.20 | 3515.20 |
| Losses (MW) | | | 13.2657 | 13.2664 |

Table 5.12. The bus phase angles and voltage magnitudes obtained in the second case for 14-bus test system

| BUS NO. | DELTA (degree) | VOLTAGE (p.u.) |
|---|---|---|
| 1 | 0.0000 | 1.0600 |
| 2 | -3.9763 | 1.0450 |
| 3 | -12.3894 | 0.9882 |
| 4 | -9.9682 | 0.9946 |
| 5 | -7.5630 | 1.0377 |
| 6 | -14.2500 | 1.0700 |
| 7 | -13.3620 | 1.0610 |
| 8 | -13.3620 | 1.0900 |
| 9 | -14.9400 | 1.0550 |
| 10 | -15.1030 | 1.0500 |
| 11 | -14.8070 | 1.0570 |
| 12 | -15.0960 | 1.0550 |
| 13 | -15.1690 | 1.0490 |
| 14 | -15.9940 | 1.0310 |

**57-bus test system**

By applying the proposed method to the 57-bus system test also gives a consistent solutions. There is a slight difference for the minimal cost when we apply this method for the system by using the hyperbolic tangent function and using the logistic function. When the hyperbolic tangent function is built in the network, and applied for power demand 1250.8 MW, the minimal cost obtained is \$ 1.8492 x $10^4$ /hour, and logistic function obtained \$ 1.8493 x $10^4$ /hour as shown in Table 5.13. This might be because of the size of the system and having too many constraints with the small value of variables (i.e. power generating increment), as happened also in 30-bus system test. Table 5.14 shows that there are no violations for voltage phase angles and voltage magnitudes obtained. The lowest voltage magnitude value is 0.9630 p.u. at bus 31, and the highest is 1.0890 at bus 46. The bus voltage phase angles have been obtained within the maximum 1.4631 degrees and minimum -18.62 degrees. By choosing different value of the initial guess for the power generating increment, the result can be seen in Table 5.15 and 5.16. The minimal costs obtained are \$ 18,506/hour and \$ 18,507/hour by using hyperbolic tangent and logistic activation function respectively. All of the voltage magnitudes and voltage phase angles are consistent with other results. The overall time needed to get convergence is 1.92 second.

Table 5.13 The optimal conditions of power generated and minimal cost obtained by using hyperbolic tangent and logistic activation function in the first case for 57-bus test system.

| Variable (MW) | Limits | | Optimal Condition | |
|---|---|---|---|---|
| | Lower (MW) | Upper (MW) | Tansig (MW) | Logsig (MW) |
| P1 | 20.00 | 600.00 | 297.3230 | 297.3150 |
| P3 | 20.00 | 600.00 | 285.9509 | 285.9754 |
| P8 | 20.00 | 600.00 | 339.9574 | 339.9787 |
| P12 | 20.00 | 600.00 | 349.9543 | 349.9771 |
| Cost ($/hr) | | | 18,492 | 18,493 |
| Losses (MW) | | | 22.3856 | 22.4462 |

Table 5.14. The bus phase angles and voltage magnitudes obtained by using hyperbolic tangent and logistic activation function in the first case for 57-bus test system

| BUS NO. | Phase Angle (degrees) | VOLTAGE (p.u.) |
|---|---|---|
| 1 | 0.0000 | 1.0600 |
| 2 | 1.4631 | 1.0499 |
| 3 | -3.2996 | 1.0200 |
| 4 | -8.3614 | 0.9854 |
| 5 | -8.1350 | 1.0030 |
| 6 | -8.0231 | 1.0055 |
| 7 | -6.6952 | 1.0077 |
| 8 | -5.2197 | 1.0200 |
| 9 | -8.4781 | 1.0048 |
| 10 | -11.0015 | 1.0067 |
| 11 | -9.5960 | 1.0000 |
| 12 | -9.3795 | 1.0400 |
| 13 | -9.5771 | 0.9981 |
| 14 | -8.9670 | 0.9970 |
| 15 | -6.7761 | 1.0058 |
| 16 | -8.1881 | 1.0358 |
| 17 | -4.3910 | 1.0451 |
| 18 | -11.2810 | 1.0300 |
| 19 | -12.7010 | 0.9990 |
| 20 | -12.9120 | 0.9920 |
| 21 | -12.4720 | 1.0360 |
| 22 | -12.4220 | 1.0370 |
| 23 | -12.4800 | 1.0360 |
| 24 | -12.7490 | 1.0260 |
| 25 | -17.4290 | 1.0100 |
| 26 | -12.4460 | 0.9840 |
| 27 | -10.9350 | 1.0040 |
| 28 | -9.9070 | 1.0180 |
| 29 | -9.2010 | 1.0310 |
| 30 | -17.9590 | 0.9900 |
| 31 | -18.6200 | 0.9630 |
| 32 | -17.8470 | 0.9760 |
| 33 | -17.8850 | 0.9730 |

| | | |
|---|---|---|
| 34 | -13.7950 | 0.9820 |
| 35 | -13.5660 | 0.9890 |
| 36 | -13.3080 | 0.9980 |
| 37 | -13.1160 | 1.0080 |
| 38 | -12.2970 | 1.0400 |
| 39 | -13.1950 | 1.0060 |
| 40 | -13.3550 | 0.9930 |
| 41 | -12.5240 | 1.0330 |
| 42 | -13.6830 | 1.0140 |
| 43 | -10.5020 | 1.0390 |
| 44 | -11.4460 | 1.0450 |
| 45 | -8.9590 | 1.0650 |
| 46 | -10.6940 | 1.0890 |
| 47 | -12.0520 | 1.0620 |
| 48 | -12.1520 | 1.0560 |
| 49 | -12.4300 | 1.0650 |
| 50 | -12.8400 | 1.0520 |
| 51 | -11.9340 | 1.0790 |
| 52 | -10.8720 | 1.0030 |
| 53 | -11.6050 | 0.9940 |
| 54 | -11.0730 | 1.0190 |
| 55 | -10.1920 | 1.0530 |
| 56 | -13.9850 | 1.0250 |
| 57 | -14.2230 | 1.0230 |

Table 5.15 The optimal conditions of power generated and minimal cost obtained in the second case when the initial guess for the neurons is changed for 57-bus test system.

| Variable (MW) | Limits | | Optimal Condition | |
|---|---|---|---|---|
| | Lower (MW) | Upper (MW) | Tansig (MW) | Logsig (MW) |
| P1 | 20.00 | 600.00 | 310.0098 | 310.0034 |
| P3 | 20.00 | 600.00 | 313.5293 | 313.5547 |
| P8 | 20.00 | 600.00 | 327.8578 | 327.8789 |
| P12 | 20.00 | 600.00 | 324.3351 | 324.3575 |
| Cost ($/hr) | | | 18,506 | 18,507 |
| Losses (MW) | | | 24.932 | 24.994 |

Table 5.16. The bus phase angles and voltage magnitudes obtained in the second case for 57-bus test system

| BUS NO. | DELTA (degrees) | VOLTAGE (p.u.) |
|---|---|---|
| 1 | 0.0000 | 1.0600 |
| 2 | 1.2830 | 1.0490 |
| 3 | -3.1170 | 1.0200 |
| 4 | -8.4481 | 0.9832 |
| 5 | -8.1350 | 1.0030 |
| 6 | -8.0055 | 1.0056 |
| 7 | -6.6524 | 1.0078 |
| 8 | -5.3403 | 1.0200 |
| 9 | -8.3982 | 1.0055 |
| 10 | -10.9558 | 1.0078 |
| 11 | -9.5960 | 1.0000 |
| 12 | -9.6427 | 1.0400 |
| 13 | -9.4817 | 1.0004 |
| 14 | -8.9670 | 0.9970 |
| 15 | -6.8823 | 1.0043 |
| 16 | -8.1374 | 1.0374 |
| 17 | -4.3974 | 1.0458 |
| 18 | -11.2810 | 1.0300 |
| 19 | -12.7010 | 0.9990 |
| 20 | -12.9120 | 0.9920 |
| 21 | -12.4720 | 1.0360 |
| 22 | -12.4220 | 1.0370 |
| 23 | -12.4800 | 1.0360 |
| 24 | -12.7490 | 1.0260 |
| 25 | -17.4290 | 1.0100 |
| 26 | -12.4460 | 0.9840 |
| 27 | -10.9350 | 1.0040 |
| 28 | -9.9070 | 1.0180 |
| 29 | -9.2010 | 1.0310 |
| 30 | -17.9590 | 0.9900 |
| 31 | -18.6200 | 0.9630 |
| 32 | -17.8470 | 0.9760 |
| 33 | -17.8850 | 0.9730 |

| | | |
|---|---|---|
| 34 | -13.7950 | 0.9820 |
| 35 | -13.5660 | 0.9890 |
| 36 | -13.3080 | 0.9980 |
| 37 | -13.1160 | 1.0080 |
| 38 | -12.2970 | 1.0400 |
| 39 | -13.1950 | 1.0060 |
| 40 | -13.3550 | 0.9930 |
| 41 | -12.5240 | 1.0330 |
| 42 | -13.6830 | 1.0140 |
| 43 | -10.5020 | 1.0390 |
| 44 | -11.4460 | 1.0450 |
| 45 | -8.9590 | 1.0650 |
| 46 | -10.6940 | 1.0890 |
| 47 | -12.0520 | 1.0620 |
| 48 | -12.1520 | 1.0560 |
| 49 | -12.4300 | 1.0650 |
| 50 | -12.8400 | 1.0520 |
| 51 | -11.9340 | 1.0790 |
| 52 | -10.8720 | 1.0030 |
| 53 | -11.6050 | 0.9940 |
| 54 | -11.0730 | 1.0190 |
| 55 | -10.1920 | 1.0530 |
| 56 | -13.9850 | 1.0250 |
| 57 | -14.2230 | 1.0230 |

In this chapter, we have applied the proposed method on the security-constrained dispatch problems for standard IEEE 5, 14, 30 and 57-bus systems. From the results obtained, we can conclude that the exterior penalty term added to penalize the objective function in case of constraint violation is quite applicable for handling the problem of security-constrained dispatch. From the comparison held between the proposed Hopfield neural network and many other methods we also discovered that the accuracy of the solutions obtained are almost the same. Moreover, the proposed algorithm consistently converges to the optimal solution within a reasonable amount of time.

# Chapter 6
# Conclusion and Future Research

A solution to Economic Power Dispatch and Security-Constrained Dispatch problem using a modified Hopfield Neural Network has been demonstrated. This proposed strategy is applied to four standard test systems.

## 6.1 Conclusion

The results obtained by the proposed Hopfield Network are relatively good in terms of accuracy and speed compared to the results obtained by the conventional method and those obtained by other techniques. The minimal costs obtained by using this method for the standard IEEE is slightly different than that obtained by conventional method. The errors for all test systems obtained are not more than 0.2 %. Therefore, the proposed technique can be used as an alternative method for solving optimization problems. By applying hyperbolic tangent and logistic activation functions reached the same optimal conditions but with different speed to obtain convergence. Using the hyperbolic tangent function gives to get convergence than that by using the logistic function with the same control factor $\lambda$. From the results shown in chapter 4, we conclude that the number of iterations to obtain convergence when the tangent hyperbolic function is applied is about a half of that required by applying logistic function.

When choosing different initial conditions, the solution can be trapped into a local minimum, as shown in Chapter 4. The results obtained are still consistant, because the error is less than 1%. Convergence to the optimal solution is guaranteed with an acceptable percentage of error.

Applying the method used in this thesis to active security-constrained dispatch for 30-bus test system gives good results, relatively similar to the solution obtained by employing gradient projection [27, 12, 46] and also similar to other Hopfield Neural Network techniques [47]. The bus voltage magnitudes and the voltage phase angles obtained are also consistent with other results.

One of the problems of the proposed technique is the need of a good starting point. When choosing a certain initial guess for the generated power increment, the results obtained and the computation time are consistent with other results as shown in Chapter 5. Because of the fast computation and good results obtained, the method shows promise for on-line applications. However a method of choosing proper initial conditions will have to be developed.

The advantage of this method is in the ease of formalization of the problem. The method requires modest memory resources and is efficient in terms of computation time. This representation is applicable to many problems other than the economic load-dispatching problem.

A disadvantage of this method is that we cannot guarantee the global optimum, because the accuracy of result depends on tuning many parameters. There are not enough guidelines to choose the parameters that give the best solution. In addition, the results depend on the initial guess, supplied to the program. Improvement in these two areas will be necessary in order to have confidence that the global optimum solution has been achieved.

It is still difficult to explain why a solution sometimes converges not to a global optimal, but to a local optimal. Moreover, the solution obtained by Hopfield Neural Network is not guaranteed to be optimal, but just near optimal. Nevertheless, for most engineering problems, suboptimal solutions are sufficient.

## 6.2 Future Research

Due to the fact that it is normally difficult to choose appropriate guidelines to tune the parameters associated with the proposed algorithm, we see that an extended study has to be carried out to test the sensitivity of the obtained solution to these parameters and to specify a criterion to choose  suitable values for these parameters.

The Hopfield neural network in continuous-time guarantees convergence to a stable equilibrium solution but suffers from the possibility of local minimum reaching a problems. One such method is described in reference [49], which uses a chaotic simulated annealing theory to have higher ability of searching for globally optimal.

The speed of convergence in our method is highly depending on the initial feasible point and how far that point from the optimal solution. Therefore, using population-based method could be helpful, because of having the ability of starting from different initial feasible point.

Only constant penalty factors have been used for weighting the constraints in the energy function. Further research can be conducted for choosing the best penalty functions and whether they must be constant or adaptive penalties.

## References


1) Wood, A.J. and Wollenberg, B.F, *Power generation, operation, and control,* Second edition, John Wiley & Sons New York, 1996.

2) Simon Haykin, *Neural Networks, a comprehensive foundation,* Second edition, Prentice Hall, Upper Saddle River, New Jersey, 1999.

3) Hadi Saadat, *Power System Analysis,* WCB McGraw-Hill Companies, New York, 1999.

4) Eiselt, H.A., G. Pederzoli, C.L.Sandblom, *Continuous Optimization Models,* Walter de Gruyter, Berlin, New York, 1987.

5) Arthur R. Bergen, *Power Systems Analysis,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1986.

6) Olle I. Elgerd, *Electric Energy Systems Theory, An Introduction,* Second Edition, McGraw-Hill Book Company, New York, 1982.

7) El-Hawary, M.E., *Electrical Power Systems: Design and Analysis,* Reston Publishing Co., Reston, Va., 1983.

8) El-Abiad, A. H., *Power Systems Analysis and Planning*, McGraw-Hill Book Company, 1983.

9) J. Duncan Glover and Mulukutla Sarma, *Power System Analysis and Design: With Personal Computer Application,* PWS-KENT Publishing Co., Boston, 1989.

10) El-Hawary, M.E. and Christensen, G.S., *Optimal Economic Operation of Electric Power System,* Academic Press, Inc., New York 1979.

11) Kasangaki, V.B.A., Sendaula, H.M., Biswas, S.K., *An Artificial Neural Network for Optimal Power Flow,* international Journal of Power and Energy Systems, Vol. 18, no. 3, 1998, pp. 225-229.

12) R. Salgado, A. Bramaller, P. Aitchison, *Optimal Power Flow solutions using the gradient projection method, part 1 and 2,* IEE Proceedings, Vol. 137, Pt. C. No. 6, November 1990, pp 424-435.

13) Abe, Shigeo et al., *Solving Inequality Constrained Combinatorial Optimization Problems by the Hopfield Neural Networks,* Neural Networks, Vol.5, pp. 663-670, 1992.

14) Park, J.H., et al., E*conomic Load Dispatch for Piecewise Quadratic Cost Function Using Hopfield neural Networks,* IEEE Transactions on Power Systems, Vol. 8, No. 3, August 1993, pp. 1030-1038.

15) Gee, A.H., and Prager, R.W., *Polyhedral combinatorics and Neural Networks,* Neural comput., 1994, 6, pp. 161-180

16) Ching-Tzong Su, Gwo-Jen Chiou, *A Fast-Computation Hopfield Method to Economic Dispatch of Power System,* IEEE transactions on Power Systems, Vol. 12, No. 4, November 1997, pp. 1759-1764.

17) Yalcinoz, T., Short, M.J., *Large Scale Economic Dispatch using an improved Hopfield Neural Network,* IEE Proc.-Gener. Transm. Distrib. vol. 144, No. 2, March 1997, pp. 181-185.

18) Yalcinoz, T., Short, M.J., *Neural Network Approach for Solving Economic Dispatch Problem with Transmission Capacity Constraints, IEEE* Transactions on Power Systems, Vol. 13, No. 2, May 1998, pp. 307-313.

19) King, T. D., El-Hawary, M.E., El-Hawary, F., *Optimal Environmental Dispatching of Electric Power Systems via an Improved Hopfield Neural Network Model,* IEEE Transaction on Power Systems, Vol. 10, No. 3, August 1995, pp. 1559-1565.

20) El-Hawary, M.E., *Applications of artificial neural networks in electric power systems operational planning,* Engineering Intelligent Systems, Vol. 7, no. 1, March 1999, 49-61.

21) Rukmi Sari Hartati, El-Hawary, M.E., *New Approach for solving Optimization Problems in Economic Load Dispatch using Hopfield Neural Networks,* Electrical and Computer Engineering 2000 Canadian Conference on, Vol. 2, 2000, pp. 722-725.

22) Alex D. Papalexopoulos, Carl F. Imparato, Felix F. Wu, *Large-Scale Optimal Power Flow : Effects of Initialization, Decoupling & Discretization,* IEEE Transactions on Power Systems, Vol. 4, No. 2, may 1989, pp. 748-759.

23) S.Matuda, Y.Akimoto, *The Representation of large numbers in neural networks and its application to economical load dispatching of electric power,* ICNN, vol. 1, June, pp. 587-592, 1989.

24) Daniel S. Kirschen, Hans P. Van Meeteren, *MW/Voltage Control in a Linear Programming Based Optimal Power Flow,* IEEE Transactions on Power Systems, Vol. 3, May 1988, pp. 481-489.

25) Tank, D.W., Hopfield, J.J., *Simple "neural" optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit,* IEEE vol. CAS-33, no. 5, pp. 533-541, May 1986.

26) Hopfield, J.J., David W. Tank, *Neural computation of decisions in optimization problems,* Biological Cybernetics, vol. 52, pp. 141-152, 1985.

27) Lee, K.Y., Y.M Park, J.L.Ortiz, *Fuel-cost minimization for both real-and reactive-power dispatches,* IEE Proceedings, Vol. 131, Pt. C, No.3, May 1984, pp. 85-93.

28) Alsac, O., Stott, B., *Decoupled Algorithms in Optimal Load Flow Calculation,* IEEE Power Engineering Society 1975 Summer Power Meeting, Paper A 75 545-4.

29) Stott, B., Alsac, O., *Fast Decoupled Load Flow,* IEEE transactions on Power Apparatus and Systems, Vol. PAS-93, pp. 859-867, May/June 1974.

30) Vanderplasts, *Numerical Optimization Techniques for Engineering Design; with application,* New-York, McGraw Hill, 1984, chapter 5, 121-152.

31) David A. Wismer, and R. Chattergy, *Introduction to Nonlinear Optimization, A problem solving approach,* Nort-Holland Publishing Company, New York, 1978.

32) Freris, L.L., Sasson, A.M., *Investigation of the load-flow problem,* Proceedings of IEE, Vol. 115, No. 10, pp. 1459-1470, 1968.

33) Happ, H.H., *Optimal Power Dispatch. A Comprehensive survey,* IEEE Trans. on Power App. Systems, Vol. PAS-96, No.3, May/June 1997, pp. 841-854.

34) Sasson, A.M. and Merril, H.M., *Some Applications of Optimization Techniques to Power System Problems,* Proceeding of IEEE, Vol.62, No. 7, 1974, pp. 959-972.

35) Carpentier, J., *Optimal power flows*, International Journal of Electrical Power and Energy Systems, Vol.1, No.1, April 1979, pp. 23-25.

36) Burchett, R.C., Happ, H.H. and Vierath, D.R., *Development in Optimal Power flow,* IEEE Trans. on Power App. Systems, Vol.PAS-101, No.2, February 1982, pp. 406-414.

37) Talaq, J.H, El-Hawary, F. and El-Hawary, M.E, *A Summary of Environmental/Economy dispatch Algorithms,* IEEE Trans. on Power Systems, Vol.9, No.3, August 1994, pp. 1508-1516.

38) William D. Stevenson, *Jr. Elements of Power Systems Analysis.* McGraw-Hill Book Co., Inc., 1975.

39) El-Abiad, A.H and Stagg, G.W., *Computer Methods in Power System Analysis,* McGraw Hill, New York, 1962.

40) Dommel, H.W., and Tinney, W.F., *Optimal Power Flow Solutions,* IEEE Trans. on Power Systems, Vol.PAS-87, Oct.1968, pp.1866-1876.

41) Atif S. Debs, *Modern Power Systems, Control and Operation,* Kluwer academic Publishers, Boston, 1988.

42) Rumelhart, D. E., McClelland, J. L., *Parallel Distributed Processing; Explorations in the Micro structure of cognition, Vol. 1* : Foundations, MIT press, Cambridge, MA, 1986.

43) McClelland, J.L. & Rumelhart, D. E., *Parallel Distributed Processing ; Explorations in the Micro structure of cognition, Vol. 2 : Psychological and Biological Models,* MIT press, Cambridge, MA, 1986.

44) McClelland, J. L. & Rumelhart, D. E., *Explorations in Parallel Distributed Processing; A hand book of models, Programs, and Exercises,* MIT press, Cambridge, MA, 1986.

45) Carpentier, J.L., *"CRIC", A new Active-Reactive Decoupling Process in Load Flows, Optimal Power Flows and System Control,* IFAC Power Systems and Power Plant Control, Beijing, 1980.

46) Robert Sjoholm, A. John Boye, *Security Constrained Dispatch Solutions using the Gradient Projection Method of Optimization,* Industrial Electronics, control, and Instrumentation, 1993. Proceedings of the IECON '93, International Conference on, 1993, pp. 2055-2060, vol. 3.

47) Yalcinoz, T., Short, M.J. and Cory, B.J., *Security Dispatch using the Hopfield Neural Network,* IEE Proceeding, Generation, Transmission Distribution, Vol. 146, No.5, September 1999, pp. 465-470.

48) Soumen Ghosh, Badrul H. Chowdhury, *Security-Constrained Optimal Rescheduling of Real Power using Hopfield Neural Network,* IEEE Transactions on Power Systems vol. 11, No. 4, November 1996

49) Luonan Chen, Kazuyuki Aihara, *Chaotic Simulated Annealing by a Neural Network Model with Transient Chaos,* Neural Networks, Vol. 8, No. 6, pp. 915-930, 1995.

50) David S. Watkins, *Fundamentals of Matrix Computations,* John Willey & Sons, New York, 1991.

51) C.Ray Wylie, Louis C. Barrett, *Advanced Engineering Mathematics,* McGraw-Hill Book Company, New York, 1982.

52) Zurada, J. M., *Introduction to Artificial Neural Systems,* West Publishing Company, New York, 1992.

53) Aiyer, S.V.B, *Solving Combinatorial Optimization Problems using Neural Networks,* Cambridge University, Engineering Department, Technical Report, CUED/F-INFENG/TR-89, October 1991.

54) El-Ela, Adel Ali Abou, *A complete strategy for Optimal Reactive Load Flow*, journal of Electrical Power and Energy, vol. 15, no. 2 Apr 1993, pp. 71-77.

55) Lau, Clifford, Neural Networks; Theoretical Foundations and Analysis, IEEE Press, 1992.

**Appendix-A**
Research Data

In this Appendix, the data used in the study reported in this thesis will be given. All the value are in per unit using 100-MVA base. However, the cost coefficients a, b and c are in \$/hour, \$/MWhour, and \$/MW$^2$hour respectively

**Input data for modified IEEE 30-bus test system**

Table A-1 The Line Data for IEEE 30-bus test system

| Line Number | From busbar number | To busbar number | Line impedance | | | Tap setting |
|---|---|---|---|---|---|---|
| | | | R (p.u) | ½ B (p.u) | X (p.u) | |
| 1 | 1 | 2 | .0192 | 0.0264 | .0575 | 1 |
| 2 | 1 | 3 | .0452 | 0.0204 | .1852 | 1 |
| 3 | 2 | 4 | .0570 | 0.0184 | .1737 | 1 |
| 4 | 3 | 4 | .0132 | 0.0042 | .0379 | 1 |
| 5 | 2 | 5 | .0472 | 0.0209 | .1983 | 1 |
| 6 | 2 | 6 | .0581 | 0.0187 | .1763 | 1 |
| 7 | 4 | 6 | .0119 | 0.0045 | .0414 | 1 |
| 8 | 5 | 7 | .0460 | 0.0102 | .1160 | 1 |
| 9 | 6 | 7 | .0267 | 0.0085 | .0820 | 1 |
| 10 | 6 | 8 | .0120 | 0.0045 | .0420 | 1 |
| 11 | 6 | 9 | .0000 | 0.0000 | .2080 | 1.078 |
| 12 | 6 | 10 | .0000 | 0.0000 | .5560 | 1.069 |
| 13 | 9 | 11 | .0000 | 0.0000 | .2080 | 1 |
| 14 | 9 | 10 | .0000 | 0.0000 | .1100 | 1 |
| 15 | 4 | 12 | .0000 | 0.0000 | .2560 | 1.032 |
| 16 | 12 | 13 | .0000 | 0.0000 | .1400 | 1 |
| 17 | 12 | 14 | .1231 | 0.0000 | .2559 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 18 | 12 | 15 | .0662 | 0.0000 | .1304 | 1 |
| 19 | 12 | 16 | .0945 | 0.0000 | .1987 | 1 |
| 20 | 14 | 15 | .2210 | 0.0000 | .1997 | 1 |
| 21 | 16 | 17 | .0824 | 0.0000 | .1932 | 1 |
| 22 | 15 | 18 | .1070 | 0.0000 | .2185 | 1 |
| 23 | 18 | 19 | .0639 | 0.0000 | .1292 | 1 |
| 24 | 19 | 20 | .0340 | 0.0000 | .0680 | 1 |
| 25 | 10 | 20 | .0936 | 0.0000 | .2090 | 1 |
| 26 | 10 | 17 | .0324 | 0.0000 | .0845 | 1 |
| 27 | 10 | 21 | .0348 | 0.0000 | .0749 | 1 |
| 28 | 10 | 22 | .0727 | 0.0000 | .1499 | 1 |
| 29 | 21 | 22 | .0116 | 0.0000 | .0236 | 1 |
| 30 | 15 | 23 | .1000 | 0.0000 | .2020 | 1 |
| 31 | 22 | 24 | .1150 | 0.0000 | .1790 | 1 |
| 32 | 23 | 24 | .1320 | 0.0000 | .2700 | 1 |
| 33 | 24 | 25 | .1885 | 0.0000 | .3292 | 1 |
| 34 | 25 | 26 | .2544 | 0.0000 | .3800 | 1 |
| 35 | 25 | 27 | .1093 | 0.0000 | .2087 | 1 |
| 36 | 28 | 27 | .0000 | 0.0000 | .3960 | 1.068 |
| 37 | 27 | 29 | .2198 | 0.0000 | .4153 | 1 |
| 38 | 27 | 30 | .3202 | 0.0000 | .6027 | 1 |
| 39 | 29 | 30 | .2399 | 0.0000 | .4533 | 1 |
| 40 | 8 | 28 | .6360 | 0.0214 | .2000 | 1 |
| 41 | 6 | 28 | .0169 | 0.0650 | .0599 | 1 |

Table A-2 The Load Data for IEEE 30-bus test system

| Busbar number | Load | |
| --- | --- | --- |
| | P (p.u) | Q (p.u) |
| 1 | 0.000 | 0.000 |
| 2 | 0.217 | 0.127 |
| 3 | 0.024 | 0.012 |
| 4 | 0.076 | 0.016 |
| 5 | 0.942 | 0.190 |
| 6 | 0.000 | 0.000 |
| 7 | 0.228 | 0.109 |
| 8 | 0.300 | 0.300 |
| 9 | 0.000 | 0.000 |
| 10 | 0.058 | 0.020 |
| 11 | 0.000 | 0.000 |
| 12 | 0.112 | 0.075 |
| 13 | 0.000 | 0.000 |
| 14 | 0.062 | 0.016 |
| 15 | 0.082 | 0.025 |
| 16 | 0.035 | 0.018 |
| 17 | 0.090 | 0.058 |
| 18 | 0.032 | 0.009 |
| 19 | 0.095 | 0.034 |
| 20 | 0.022 | 0.007 |
| 21 | 0.175 | 0.112 |
| 22 | 0.000 | 0.000 |
| 23 | 0.032 | 0.016 |
| 24 | 0.087 | 0.067 |
| 25 | 0.000 | 0.000 |
| 26 | 0.035 | 0.023 |

| | | |
|---|---|---|
| 27 | 0.000 | 0.000 |
| 28 | 0.000 | 0.000 |
| 29 | 0.024 | 0.009 |
| 30 | 0.106 | 0.019 |

Table A-3 The Generator Data for IEEE 30-bus test system

| Busbar number | Cost coefficients | | |
|---|---|---|---|
| | a | b | c |
| 1 | 0.0 | 2.00 | 0.00375 |
| 2 | 0.0 | 1.75 | 0.01750 |
| 5 | 0.0 | 1.00 | 0.06250 |
| 8 | 0.0 | 3.25 | 0.00834 |
| 11 | 0.0 | 3.00 | 0.02500 |
| 13 | 0.0 | 3.00 | 0.02500 |

| Generation Data | | | |
|---|---|---|---|
| Bus number | Voltage Mag | MW limits | |
| | | Min. | Max. |
| 1 | 1.060 | 50 | 200 |
| 2 | 1.043 | 20 | 80 |
| 5 | 1.010 | 15 | 50 |
| 8 | 1.010 | 10 | 35 |
| 11 | 1.082 | 10 | 30 |
| 13 | 1.075 | 12 | 40 |

**Input Data for the standard IEEE 14-bus system**

Table A-4 The Line Data for standard IEEE 14-bus test system

| Line number | From busbar number | To busbar number | Line Impedance | | | Tap setting |
|---|---|---|---|---|---|---|
| | | | R (p.u) | ½ B (p.u) | X (p.u) | |
| 1 | 1 | 2 | 0.01938 | .0264 | 0.05917 | 1 |
| 2 | 2 | 3 | 0.04699 | .0219 | 0.19797 | 1 |
| 3 | 2 | 4 | 0.05811 | .0187 | 0.17632 | 1 |
| 4 | 1 | 5 | 0.05403 | .0264 | 0.22304 | 1 |
| 5 | 2 | 5 | 0.05695 | .0170 | 0.17388 | 1 |
| 6 | 3 | 4 | 0.06701 | .0173 | 0.17103 | 1 |
| 7 | 4 | 5 | 0.01335 | .0064 | 0.04211 | 1 |
| 8 | 5 | 6 | 0.00000 | .0000 | 0.25202 | 0.932 |
| 9 | 4 | 7 | 0.00000 | .0000 | 0.20912 | 0.978 |
| 10 | 7 | 8 | 0.00000 | .0000 | 0.17615 | 1 |
| 11 | 4 | 9 | 0.00000 | .0000 | 0.55618 | 0.969 |
| 12 | 7 | 9 | 0.00000 | .0000 | 0.11001 | 1 |
| 13 | 9 | 10 | 0.03181 | .0000 | 0.08450 | 1 |
| 14 | 6 | 11 | 0.09498 | .0000 | 0.19890 | 1 |
| 15 | 6 | 12 | 0.12291 | .0000 | 0.25581 | 1 |
| 16 | 6 | 13 | 0.06615 | .0000 | 0.13027 | 1 |
| 17 | 9 | 14 | 0.19711 | .0000 | 0.27038 | 1 |
| 18 | 10 | 11 | 0.08205 | .0000 | 0.19207 | 1 |
| 19 | 12 | 13 | 0.22092 | .0000 | 0.19988 | 1 |
| 20 | 13 | 14 | 0.17093 | .0000 | 0.34802 | 1 |

Table A-5 The Load Data for standard IEEE 14-bus test system

| Busbar number | Load | |
|---|---|---|
| | P (p.u) | Q (p.u) |
| 1 | 0.000 | 0.000 |
| 2 | 0.217 | 0.127 |
| 3 | 0.942 | 0.190 |
| 4 | 0.478 | -0.039 |
| 5 | 0.076 | 0.016 |
| 6 | 0.112 | 0.075 |
| 7 | 0.000 | 0.000 |
| 8 | 0.000 | 0.000 |
| 9 | 0.295 | 0.166 |
| 10 | 0.090 | 0.058 |
| 11 | 0.035 | 0.018 |
| 12 | 0.061 | 0.016 |
| 13 | 0.135 | 0.058 |
| 14 | 0.149 | 0.050 |

Table A-6 The Generator Data for standard IEEE 14-bus test system

| Busbar Number | Cost coefficients | | |
|---|---|---|---|
| | a | b | c |
| 1 | 50.607 | 10.662 | 0.01165 |
| 2 | 50.607 | 10.662 | 0.01165 |

| Generation Data | | | |
|---|---|---|---|
| Bus number | Voltage Mag | MW limits | |
| | | Min. | Max. |
| 1 | 1.060 | 10 | 300 |
| 2 | 1.045 | 10 | 100 |

**Input Data for the standard IEEE-57 bus system**

Table A-7 The Line Data for standard IEEE 57-bus test system

| Line Number | From busbar number | To busbar number | Line Impedance | | | Tap Setting |
|---|---|---|---|---|---|---|
| | | | R (p.u) | X (p.u) | ½ B (p.u) | |
| 1 | 1 | 2 | .0083 | .0280 | .0645 | 1 |
| 2 | 2 | 3 | .0298 | .0850 | .0409 | 1 |
| 3 | 3 | 4 | .0112 | .0366 | .0190 | 1 |
| 4 | 4 | 5 | .0625 | .1320 | .0129 | 1 |
| 5 | 4 | 6 | .0430 | .1480 | .0174 | 1 |
| 6 | 6 | 7 | .0200 | .1020 | .0138 | 1 |
| 7 | 6 | 8 | .0339 | .1730 | .0235 | 1 |
| 8 | 8 | 9 | .0099 | .0505 | .0274 | 1 |
| 9 | 9 | 10 | .0369 | .1679 | .0220 | 1 |
| 10 | 9 | 11 | .0258 | .0848 | .0109 | 1 |
| 11 | 9 | 12 | .0648 | .2950 | .0386 | 1 |
| 12 | 9 | 13 | .0481 | .1580 | .0203 | 1 |
| 13 | 13 | 14 | .0132 | .0434 | .0055 | 1 |
| 14 | 13 | 15 | .0269 | .0869 | .0115 | 1 |
| 15 | 1 | 15 | .0178 | .0910 | .0494 | 1 |
| 16 | 1 | 16 | .0454 | .2060 | .0273 | 1 |
| 17 | 1 | 17 | .0238 | .1080 | .0143 | 1 |
| 18 | 3 | 15 | .0162 | .0530 | .0272 | 1 |
| 19 | 4 | 18 | .0000 | .2423 | .0000 | 0.978 |
| 20 | 5 | 6 | .0302 | .0641 | .0062 | 1 |
| 21 | 7 | 8 | .0139 | .0712 | .0097 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 22 | 10 | 12 | .0277 | .1262 | .0164 | 1 |
| 23 | 11 | 13 | .0223 | .0732 | .0094 | 1 |
| 24 | 12 | 13 | .0178 | .0580 | .0302 | 1 |
| 25 | 12 | 16 | .0180 | .0813 | .0108 | 1 |
| 26 | 12 | 17 | .0397 | .1790 | .0238 | 1 |
| 27 | 14 | 15 | .0171 | .0547 | .0074 | 1 |
| 28 | 18 | 19 | .4610 | .6850 | .0000 | 1 |
| 29 | 19 | 20 | .2830 | .4340 | .0000 | 1 |
| 30 | 21 | 20 | .0000 | .7767 | .0000 | 1.043 |
| 31 | 21 | 22 | .0736 | .1170 | .0000 | 1 |
| 32 | 22 | 23 | .0099 | .0152 | .0000 | 1 |
| 33 | 23 | 24 | .1660 | .2560 | .0042 | 1 |
| 34 | 24 | 25 | .0000 | .60276 | .0000 | 1 |
| 35 | 24 | 26 | .0000 | .0473 | .0000 | 1.043 |
| 36 | 26 | 27 | .1650 | .2540 | .0000 | 1 |
| 37 | 27 | 28 | .0618 | .0954 | .0000 | 1 |
| 38 | 28 | 29 | .0418 | .0587 | .0000 | 1 |
| 39 | 7 | 29 | .0000 | .0648 | .0000 | 0.967 |
| 40 | 25 | 30 | .1350 | .2020 | .0000 | 1 |
| 41 | 30 | 31 | .3260 | .4970 | .0000 | 1 |
| 42 | 31 | 32 | .5070 | .7550 | .0000 | 1 |
| 43 | 32 | 33 | .0392 | .0360 | .0000 | 1 |
| 44 | 34 | 32 | .0000 | .9530 | .0000 | 0.975 |
| 45 | 34 | 35 | .0520 | .0380 | .0016 | 1 |
| 46 | 35 | 36 | .0430 | .0537 | .0008 | 1 |
| 47 | 36 | 37 | .0290 | .0366 | .0000 | 1 |
| 48 | 37 | 38 | .0651 | .1009 | .0010 | 1 |
| 49 | 37 | 39 | .0239 | .0379 | .0000 | 1 |
| 50 | 36 | 40 | .0300 | .0466 | .0000 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 51 | 22 | 38 | .0192 | .0295 | .0000 | 1 |
| 52 | 11 | 41 | .0000 | .7490 | .0000 | 0.955 |
| 53 | 41 | 42 | .2070 | .3520 | .0000 | 1 |
| 54 | 41 | 43 | .0000 | .4120 | .0000 | 1 |
| 55 | 38 | 44 | .0289 | .0585 | .0010 | 1 |
| 56 | 15 | 45 | .0000 | .1042 | .0000 | 0.955 |
| 57 | 14 | 46 | .0000 | .0735 | .0000 | 0.9 |
| 58 | 46 | 47 | .0230 | .0680 | .0016 | 1 |
| 59 | 47 | 48 | .0182 | .0233 | .0000 | 1 |
| 60 | 48 | 49 | .0834 | .1290 | .0024 | 1 |
| 61 | 49 | 50 | .0801 | .1280 | .0000 | 1 |
| 62 | 50 | 51 | .1386 | .2200 | .0000 | 1 |
| 63 | 10 | 51 | .0000 | .0712 | .0000 | 0.93 |
| 64 | 13 | 49 | .0000 | .1910 | .0000 | 0.895 |
| 65 | 29 | 52 | .1442 | .1870 | .0000 | 1 |
| 66 | 52 | 53 | .0762 | .0987 | .0000 | 1 |
| 67 | 53 | 54 | .1878 | .2320 | .0000 | 1 |
| 68 | 54 | 55 | .1732 | .2265 | .0000 | 1 |
| 69 | 11 | 43 | .0000 | .1530 | .0000 | 0.958 |
| 70 | 44 | 45 | .0624 | .1242 | .0020 | 1 |
| 71 | 40 | 56 | .0000 | .1950 | .0000 | 0.958 |
| 72 | 56 | 41 | .5530 | .5490 | .0000 | 1 |
| 73 | 56 | 42 | .2125 | .3540 | .0000 | 1 |
| 74 | 39 | 57 | .0000 | .3550 | .0000 | 0.975 |
| 75 | 57 | 56 | .1740 | .2600 | .0000 | 1 |
| 76 | 38 | 49 | .1150 | .1770 | .0030 | 1 |
| 77 | 38 | 48 | .0312 | .0482 | .0000 | 1 |
| 78 | 9 | 55 | .0000 | .1205 | .0000 | 0.94 |

Table A-8 The Load Data for standard IEEE 57-bus test system

| Busbar number | Load | |
|---|---|---|
| | P (p.u) | Q (p.u) |
| 1 | 0.550 | 0.170 |
| 2 | 0.030 | 0.880 |
| 3 | 0.410 | 0.210 |
| 4 | 0.000 | 0.000 |
| 5 | 0.130 | 0.040 |
| 6 | 0.750 | 0.020 |
| 7 | 0.000 | 0.000 |
| 8 | 1.500 | 0.220 |
| 9 | 1.210 | 0.260 |
| 10 | 0.050 | 0.020 |
| 11 | 0.000 | 0.000 |
| 12 | 3.770 | 0.240 |
| 13 | 0.180 | 0.023 |
| 14 | 0.105 | 0.053 |
| 15 | 0.220 | 0.050 |
| 16 | 0.430 | 0.030 |
| 17 | 0.420 | 0.080 |
| 18 | 0.272 | 0.098 |
| 19 | 0.033 | 0.006 |
| 20 | 0.023 | 0.010 |
| 21 | 0.000 | 0.000 |
| 22 | 0.000 | 0.000 |
| 23 | 0.063 | 0.021 |
| 24 | 0.000 | 0.000 |

| | | |
|---|---|---|
| 25 | 0.063 | 0.032 |
| 26 | 0.000 | 0.000 |
| 27 | 0.093 | 0.005 |
| 28 | 0.046 | 0.023 |
| 29 | 0.170 | 0.026 |
| 30 | 0.036 | 0.018 |
| 31 | 0.058 | 0.029 |
| 32 | 0.016 | 0.008 |
| 33 | 0.038 | 0.019 |
| 34 | 0.000 | 0.000 |
| 35 | 0.060 | 0.030 |
| 36 | 0.000 | 0.000 |
| 37 | 0.000 | 0.000 |
| 38 | 0.140 | 0.070 |
| 39 | 0.000 | 0.000 |
| 40 | 0.000 | 0.000 |
| 41 | 0.063 | 0.030 |
| 42 | 0.071 | 0.044 |
| 43 | 0.020 | 0.010 |
| 44 | 0.120 | 0.018 |
| 45 | 0.000 | 0.000 |
| 46 | 0.000 | 0.000 |
| 47 | 0.297 | 0.116 |
| 48 | 0.000 | 0.000 |
| 49 | 0.180 | 0.085 |
| 50 | 0.210 | 0.105 |
| 51 | 0.180 | 0.053 |
| 52 | 0.049 | 0.022 |
| 53 | 0.200 | 0.100 |

| | | |
|---|---|---|
| 54 | 0.041 | 0.014 |
| 55 | 0.068 | 0.034 |
| 56 | 0.076 | 0.022 |
| 57 | 0.067 | 0.020 |

Table A-9 The Line Data for standard IEEE 57-bus test system

| Busbar Number | Cost coefficients | | |
|---|---|---|---|
| | a | b | c |
| 1 | 50.607 | 10.662 | 0.01155 |
| 2 | 50.607 | 10.662 | 0.01155 |
| 3 | 50.607 | 10.662 | 0.01155 |
| 4 | 50.607 | 10.662 | 0.01155 |

| Generation Data | | | |
|---|---|---|---|
| Bus number | Voltage Mag | MW limits | |
| | | Min. | Max. |
| 1 | 1.060 | 20 | 600 |
| 3 | 1.040 | 20 | 600 |
| 8 | 1.020 | 20 | 600 |
| 12 | 1.040 | 20 | 600 |

**Input Data for IEEE 5-bus test system**

Table A-10 The Line Data for IEEE 5-bus test system

| Line number | From busbar number | To busbar number | Line impedance | | | Tap Setting |
|---|---|---|---|---|---|---|
| | | | R (p.u) | X (p.u) | ½ B (p.u) | |
| 1 | 1 | 2 | 0.02 | 0.06 | .030 | 1 |
| 2 | 1 | 3 | 0.08 | 0.24 | .025 | 1 |
| 3 | 2 | 3 | 0.06 | 0.18 | .020 | 1 |
| 4 | 2 | 4 | 0.06 | 0.18 | .020 | 1 |
| 5 | 2 | 5 | 0.04 | 0.12 | .015 | 1 |
| 6 | 3 | 4 | 0.01 | 0.03 | .010 | 1 |
| 7 | 4 | 5 | 0.08 | 0.24 | .025 | 1 |

Table A-11 The Load Data for IEEE 5-bus test system

| Busbar number | Load | |
|---|---|---|
| | P (p.u) | Q (p.u) |
| 1 | 0.00 | 0.00 |
| 2 | 0.20 | 0.10 |
| 3 | 0.20 | 0.15 |
| 4 | 0.50 | 0.30 |
| 5 | 0.60 | 0.40 |

Table A-12 The Generator Data for IEEE 5-bus test system

| Busbar Number | Cost coefficients | | |
|---|---|---|---|
| | a | b | c |
| 1 | 200 | 7.0 | 0.008 |
| 2 | 180 | 6.3 | 0.009 |
| 3 | 140 | 6.8 | 0.007 |

| Generation Data | | | |
|---|---|---|---|
| Bus number | Voltage Mag | MW limits | |
| | | Min. | Max. |
| 1 | 1.060 | 10 | 85 |
| 2 | 1.045 | 10 | 80 |
| 3 | 1.030 | 10 | 70 |

# Appendix-B

## Pseudoinverse of matrix

The pseudoinverse, known as the Moore-Penrose generalized inverse, is an interesting generalization of the ordinary inverse.

Consider a system of $n$ linear algebraic equations in $n$ unknowns

$a_{i1}x_1 + a_{i2}x_2 + .................. + a_{in}x_n = b_i$

whose matrix form

$Ax = b$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ . & . & . & . \\ . & . & . & . \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{bmatrix}$$

The coefficients $a_{ij}$ and $b_i$ are given, and we wish to find $x_1, x_2, ............x_n$ that satisfy the equations. A is a square matrix.

Only square and non-singular linear system matrices have inverses in the ordinary sense. It can be expressed in terms of $A^{-1}$ as $x = A^{-1}b$.

For the nonsquare coefficient matrix $A$ can be expressed in terms of the pseudoinverse $A^*$ as $x = A^*b$ [50].

The procedure used here is based on the classical method of least squares.

According to [50],

If $A \in \Re^{nxm}$, $n \leq m$, and rank ( $A$ ) = $n$, then the pseudoinverse of $A$ = $A^{*} = A^{T}\left(AA^{T}\right)^{-1}$

Where $A^{T}$ is the transpose of matrix $A$. This pseudoinverse will satisfy the condition $A^{*}A = I$ , where $I$ is the identity matrix.

**Appendix-C**

**Simulator Source Code**

**Economic Dispatch Problems**

```
%         IEEE 30-BUS TEST SYSTEM (American Electric Power)
%         Bus Bus  Voltage Angle   ---Load---- -------Generator--
%         No  code Mag.    Degree  MW     Mvar  MW   Mvar Qmin Qmax
busdata= 1   1    1.06    0.0      0.0    0.0   0.0  0.0    0    0
         2   2    1.043   0.0    21.70   12.7  80.0  0.0  -40   50
         3   0    1.0     0.0      2.4    1.2   0.0  0.0    0    0
         4   0    1.06    0.0      7.6    1.6   0.0  0.0    0    0
         5   2    1.01    0.0     94.2   19.0  50.0  0.0  -40   40
         6   0    1.0     0.0      0.0    0.0   0.0  0.0    0    0
         7   0    1.0     0.0     22.8   10.9   0.0  0.0    0    0
         8   2    1.01    0.0     30.0   30.0  20.0  0.0  -30   40
         9   0    1.0     0.0      0.0    0.0   0.0  0.0    0    0
        10   0    1.0     0.0      5.8    2.0   0.0  0.0   -6   24
        11   2    1.082   0.0      0.0    0.0  20.0  0.0    0    0
        12   0    1.0     0       11.2    7.5   0    0      0    0
        13   2    1.071   0        0      0.0  20    0     -6   24
        14   0    1       0        6.2    1.6   0    0      0    0
        15   0    1       0        8.2    2.5   0    0      0    0
        16   0    1       0        3.5    1.8   0    0      0    0
        17   0    1       0        9.0    5.8   0    0      0    0
        18   0    1       0        3.2    0.9   0    0      0    0
        19   0    1       0        9.5    3.4   0    0      0    0
        20   0    1       0        2.2    0.7   0    0      0    0
        21   0    1       0       17.5   11.2   0    0      0    0
        22   0    1       0       0       0.0   0    0      0    0
        23   0    1       0        3.2    1.6   0    0      0    0
        24   0    1       0        8.7    6.7   0    0      0    0
        25   0    1       0       0       0.0   0    0      0    0
        26   0    1       0        3.5    2.3   0    0      0    0
        27   0    1       0       0       0.0   0    0      0    0
        28   0    1       0       0       0.0   0    0      0    0
```

```
        29   0    1       0       2.4   0.9    0    0     0   0
        30   0    1       0      10.6   1.9    0    0     0   0
```

To calculate power demand.

```
PD=sum(busdata(:,5));
```

To define cost coefficienfs

```
p=[0 0 0 0 0 0];
q=[2.0 1.75 1.0 3.25 3.0 3.0];
r=[ 0.00375 0.0175 0.0625 0.00834 0.025 0.025];
```

To define B coefficient matrix.

```
Bij=[0.0218    0.0102    0.0010   -0.0010    0.0001    0.0027
     0.0102    0.0187    0.0004   -0.0015    0.0003    0.0031
     0.0010    0.0004    0.0430   -0.0134   -0.0160   -0.0108
    -0.0010   -0.0015   -0.0134    0.0224    0.0097    0.0051
     0.0001    0.0003   -0.0160    0.0097    0.0256   -0.0000
     0.0027    0.0031   -0.0108    0.0051   -0.0000    0.0358];

B0= [-0.0003;0.0022;-0.0057;0.0034;0.0016;0.0078];

B00 = 0.0014;

mwlimits=[ 50  200
           20  80
           15  50
           10  35
           10  30
           12  40];

Pmin = mwlimits(:,1);
Pmax = mwlimits(:,2);

ng=6;
```

```
lambda=0.001;
Alpha = 10^6;
Beta = 1;
Gama = 10^3;
dt=0.00001;

 sigP=ones(1,ng);
 sigPsq=sigP'*sigP;
 PP=eye(ng);
 constr=[PP;-PP];
 constrsq=constr'*constr;
 b=[Pmax;-Pmin];

 % To define cost function
 Tobj=diag(r);

 % Simulation of neural network
 % To get initial value of x

  x=[155.00
      55.00
      25.00
      25.00
      16.00
      15.00];

  To calculate input u by using inverse of sigmoid function.

  for i=1:ng
     temp1(i)=2*(x(i)-Pmin(i))/(Pmax(i)-Pmin(i));
     temp2(i)=atanh(temp1(i)-1);
     u(i)=temp2(i)/lambda;
  end

 %for i=1:ng
```

```
%   temp1(i)=(x(i)-Pmin(i))/(Pmax(i)-Pmin(i));
%   u(i)=(-1/lambda)*log((1/temp1(i))-1);
%end

 N=1;
 while N<100
 % To define x
  for i=1:ng
     x(i)=(Pmax(i)-Pmin(i))*(tansig(lambda*u(i))+1)/2+ Pmin(i);
  end

  x = [x(1);x(2);x(3);x(4);x(5);x(6)]


  To calculate power losses by using Kron's method
  PL=0.01*(x'*Bij*x+B0'*x+B00);


  To solve inequality constraints by using penalty function method

  W=zeros(ng);
  s=zeros(ng,1);
  for i=1:2*ng
     if constr(i,:)*x<=b(i)
        h(i)=0;
     else
        h(i)=1;
        W=W+(constr(i,:))'*constr(i,:);
        s=s-b(i)*(constr(i,:))';
     end
  end

 Mapping from the augmented objective function into Hopfield NN.
 To define weights and input bias.

 weight=-(Alpha*sigPsq+Beta*Tobj+Gama*W);
 ib=Alpha*(PD+PL)*sigP'-Beta*q'/2+Gama*s;
```

```
  % To update u
  temp = weight*x+ib;
  u = u + temp*dt;

  To calculate energy function.

  E=-.5*x'*weight*x-ib'*x;
  N=N+1;
end

% To calculate Power generated and minimal cost
   Power=x
   cost=0.0;
   for i=1:ng
      cost=cost+p(i)+q(i)*Power(i)+r(i)*(Power(i))^2;
   end
   cost


sigmaP=0.0;
for i=1:ng
   sigmaP=sigmaP+Power(i);
end
sigmaP
```

**Active Security-Constrained Dispatch**

```
%         IEEE 30-BUS TEST SYSTEM (American Electric Power)
%         Bus Bus  Voltage Angle   ---Load---- -------Generator--
%         No  code Mag.    Degree  MW     Mvar  MW   Mvar Qmin Qmax
busdata= 1   1    1.06    0.0      0.0    0.0    0.0 0.0   0    0
         2   2    1.043   0.0    21.70   12.7   80.0 0.0 -40   50
         3   0    1.0     0.0      2.4    1.2    0.0 0.0   0    0
         4   0    1.06    0.0      7.6    1.6    0.0 0.0   0    0
         5   2    1.01    0.0     94.2   19.0   50.0 0.0 -40   40
         6   0    1.0     0.0      0.0    0.0    0.0 0.0   0    0
         7   0    1.0     0.0     22.8   10.9    0.0 0.0   0    0
         8   2    1.01    0.0     30.0   30.0   20.0 0.0 -30   40
         9   0    1.0     0.0      0.0    0.0    0.0 0.0   0    0
        10   0    1.0     0.0      5.8    2.0    0.0 0.0  -6   24
        11   2    1.082   0.0      0.0    0.0   20.0 0.0   0    0
        12   0    1.0     0        11.2   7.5    0    0    0    0
        13   2    1.071   0        0      0.0   20    0   -6   24
        14   0    1       0        6.2    1.6    0    0    0    0
        15   0    1       0        8.2    2.5    0    0    0    0
        16   0    1       0        3.5    1.8    0    0    0    0
        17   0    1       0        9.0    5.8    0    0    0    0
        18   0    1       0        3.2    0.9    0    0    0    0
        19   0    1       0        9.5    3.4    0    0    0    0
        20   0    1       0        2.2    0.7    0    0    0    0
        21   0    1       0       17.5   11.2    0    0    0    0
        22   0    1       0       0       0.0    0    0    0    0
        23   0    1       0        3.2    1.6    0    0    0    0
        24   0    1       0        8.7    6.7    0    0    0    0
        25   0    1       0       0       0.0    0    0    0    0
        26   0    1       0        3.5    2.3    0    0    0    0
        27   0    1       0       0       0.0    0    0    0    0
        28   0    1       0       0       0.0    0    0    0    0
        29   0    1       0        2.4    0.9    0    0    0    0
        30   0    1       0       10.6    1.9    0    0    0    0
```

```
%                                                Line code
%          Bus bus   R       X      1/2 B    = 1 for lines
%          nl  nr  p.u.    p.u.   p.u.      > 1 or < 1 tr. tap
linedata= 1   2   0.0192   0.0575   0.02640    1
          1   3   0.0452   0.1852   0.02040    1
          2   4   0.0570   0.1737   0.01840    1
          3   4   0.0132   0.0379   0.00420    1
          2   5   0.0472   0.1983   0.02090    1
          2   6   0.0581   0.1763   0.01870    1
          4   6   0.0119   0.0414   0.00450    1
          5   7   0.0460   0.1160   0.01020    1
          6   7   0.0267   0.0820   0.00850    1
          6   8   0.0120   0.0420   0.00450    1
          6   9   0.0      0.2080   0.0     0.978
          6  10   0         .5560   0       0.969
          9  11   0         .2080   0           1
          9  10   0         .1100   0           1
          4  12   0         .2560   0       0.932
         12  13   0         .1400   0           1
         12  14    .1231    .2559   0           1
         12  15    .0662    .1304   0           1
         12  16    .0945    .1987   0           1
         14  15    .2210    .1997   0           1
         16  17    .0824    .1923   0           1
         15  18    .1073    .2185   0           1
         18  19    .0639    .1292   0           1
         19  20    .0340    .0680   0           1
         10  20    .0936    .2090   0           1
         10  17    .0324    .0845   0           1
         10  21    .0348    .0749   0           1
         10  22    .0727    .1499   0           1
         21  22    .0116    .0236   0           1
         15  23    .1000    .2020   0           1
         22  24    .1150    .1790   0           1
         23  24    .1320    .2700   0           1
         24  25    .1885    .3292   0           1
```

| | | | | | |
|---|---|---|---|---|---|
| 25 | 26 | .2544 | .3800 | 0 | 1 |
| 25 | 27 | .1093 | .2087 | 0 | 1 |
| 28 | 27 | 0 | .3960 | 0 | 0.968 |
| 27 | 29 | .2198 | .4153 | 0 | 1 |
| 27 | 30 | .3202 | .6027 | 0 | 1 |
| 29 | 30 | .2399 | .4533 | 0 | 1 |
| 8 | 28 | .0636 | .2000 | 0.0214 | 1 |
| 6 | 28 | .0169 | .0599 | 0.065 | 1 |

This is the initial values from basic load flow

| % Bus | Bus | Voltage | Angle | ---Load---- | -------Generator-- | | |
|---|---|---|---|---|---|---|---|
| % No | code | Mag. | Degree | MW | Mvar | MW | Mvar |
| init=1 | 1 | 1.060 | 0.000 | 0.000 | 0.000 | 98.684 | 17.583 |
| 2 | 2 | 1.043 | -1.611 | 21.700 | 12.700 | 80.000 | 10.010 |
| 3 | 0 | 1.027 | -3.699 | 2.400 | 1.200 | 0.000 | 0.000 |
| 4 | 0 | 1.019 | -4.411 | 7.600 | 1.600 | 0.000 | 0.000 |
| 5 | 2 | 1.010 | -6.280 | 94.200 | 19.000 | 50.000 | 15.142 |
| 6 | 0 | 1.016 | -5.224 | 0.000 | 0.000 | 0.000 | 0.000 |
| 7 | 0 | 1.006 | -6.189 | 22.800 | 10.900 | 0.000 | 0.000 |
| 8 | 2 | 1.010 | -5.426 | 30.000 | 30.000 | 20.000 | 13.843 |
| 9 | 0 | 1.054 | -6.609 | 0.000 | 0.000 | 0.000 | 0.000 |
| 10 | 0 | 1.048 | -8.478 | 5.800 | 2.000 | 0.000 | 0.000 |
| 11 | 2 | 1.082 | -4.518 | 0.000 | 0.000 | 20.000 | 15.090 |
| 12 | 0 | 1.061 | -7.641 | 11.200 | 7.500 | 0.000 | 0.000 |
| 13 | 2 | 1.071 | -6.229 | 0.000 | 0.000 | 20.000 | 7.799 |
| 14 | 0 | 1.046 | -8.554 | 6.200 | 1.600 | 0.000 | 0.000 |
| 15 | 0 | 1.041 | -8.668 | 8.200 | 2.500 | 0.000 | 0.000 |
| 16 | 0 | 1.048 | -8.265 | 3.500 | 1.800 | 0.000 | 0.000 |
| 17 | 0 | 1.043 | -8.620 | 9.000 | 5.800 | 0.000 | 0.000 |
| 18 | 0 | 1.031 | -9.291 | 3.200 | 0.900 | 0.000 | 0.000 |
| 19 | 0 | 1.029 | -9.472 | 9.500 | 3.400 | 0.000 | 0.000 |
| 20 | 0 | 1.033 | -9.280 | 2.200 | 0.700 | 0.000 | 0.000 |
| 21 | 0 | 1.035 | -8.949 | 17.500 | 11.200 | 0.000 | 0.000 |
| 22 | 0 | 1.036 | -8.945 | 0.000 | 0.000 | 0.000 | 0.000 |
| 23 | 0 | 1.031 | -9.146 | 3.200 | 1.600 | 0.000 | 0.000 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 24 | 0 | 1.025 | -9.444 | 8.700 | 6.700 | 0.000 | 0.000 |
| 25 | 0 | 1.021 | -9.433 | 0.000 | 0.000 | 0.000 | 0.000 |
| 26 | 0 | 1.003 | -9.850 | 3.500 | 2.300 | 0.000 | 0.000 |
| 27 | 0 | 1.027 | -9.169 | 0.000 | 0.000 | 0.000 | 0.000 |
| 28 | 0 | 1.014 | -5.714 | 0.000 | 0.000 | 0.000 | 0.000 |
| 29 | 0 | 1.007 | -10.390 | 2.400 | 0.900 | 0.000 | 0.000 |
| 30 | 0 | 0.996 | -11.266 | 10.600 | 1.900 | 0.000 | 0.000 |

```
To calculate power demand.
PD=sum(busdata(:,5));

To define cost coefficienfs
p=[0 0 0 0 0 0];
q=10^2*[2.0 1.75 1.0 3.25 3.0 3.0];
r=10^4*[ 0.00375 0.0175 0.0625 0.00834 0.025 0.025];

ng=6;

To set up the parameters.
lambda=0.6;
A = 10^-4;
B = 10;
C = 10^3;
Delta = 10^3;
dt=0.0001;

% To get the initial value of generated powers, voltage phase angles
and voltage magnitudes.

    initV1=init(:,3);
    initD1=init(:,4)*pi/180;
    initP1=init(:,7)/100;
    nbus = length(init(:,1));
    code=init(:,2);
```

```
ii=0;
for k=1:nbus
   if code(k)==0
      ii=ii+1;
      initV(ii)=initV1(k);
   else,end
end
initV=initV';


ii=0;
for k=1:nbus
   if code(k)==0 | code(k)==2
      ii=ii+1;
      initD(ii)=initD1(k);
   else,end
end
initD=initD';

ii=0;
for k=1:nbus
   if code(k)==1 | code(k)==2
      ii=ii+1;
      initP(ii)=initP1(k);
   else,end
end
initP=initP';
```

To define the limits of generated powers, voltage phase angles and voltage magnitudes.

```
Pmin= .01*[50 20 15 10 10 12]';

Pmax= .01*[200 80 50 35 30 40]';
```

```
for i=1:length(initD)
   Dmin(i)= -pi/9;
end

for i=1:length(initD)
   Dmax(i)= pi/12;
end

for i=1:length(initV)
   Vmin(i)= 0.90;
end

for i=1:length(initV)
   Vmax(i)= 1.10;
end
```

To define the initial value of the limits of increment generated powers, increments phase angles and increment voltage magnitudes.

```
dPmin=Pmin-initP;

dPmax=Pmax-initP;

dDmin=Dmin-initD;
dDmax=Dmax-initD;

dVmin=Vmin-initV;
dVmax=Vmax-initV;
```

To define the inequality constraints.

```
constrmin=[dPmin;dDmin;dVmin];
constrmax=[dPmax;dDmax;dVmax];
```

To calculate Ybus matrix.

```
nl = linedata(:,1); nr = linedata(:,2); R = linedata(:,3);
X = linedata(:,4); Bc = j*linedata(:,5); a = linedata(:, 6);
nbr=length(linedata(:,1)); nbus = max(max(nl), max(nr));
```

```
Z = R + j*X; y= ones(nbr,1)./Z;         %branch admittance
for n = 1:nbr
if a(n) <= 0  a(n) = 1; else end
Ybus=zeros(nbus,nbus);      % initialize Ybus to zero
               % formation of the off diagonal elements
for k=1:nbr;
       Ybus(nl(k),nr(k))=Ybus(nl(k),nr(k))-y(k)/a(k);
       Ybus(nr(k),nl(k))=Ybus(nl(k),nr(k));
end
end
              % formation of the diagonal elements
for  n=1:nbus
     for k=1:nbr
         if nl(k)==n
         Ybus(n,n) = Ybus(n,n)+y(k)/(a(k)^2) + Bc(k);
         elseif nr(k)==n
         Ybus(n,n) = Ybus(n,n)+y(k) +Bc(k);
         else, end
     end
  end
  Ybus;

ns=0;
nbus = length(busdata(:,1));
for k=1:nbus
n=busdata(k,1);
kb(n)=busdata(k,2);
if kb(n) == 1, ns = ns+1; else, end
nss(n) = ns;
end
Ym = abs(Ybus); t = angle(Ybus);
G=real(Ybus);

B1=imag(Ybus);
```

To calculate the coefficients of B′ and B″ matrices.

```
for i=1:nbus
   for j=1:nbus
      dPdD(i,j)=-initV1(i)*B1(i,j);
   end
end
dPdD;

ii=0;
for ib=1:nbus
     if kb(ib)==1 | kb(ib) == 2
     ii = ii+1;
      jj=0;
         for jb=1:nbus
             if kb(jb) == 0 | kb(jb) == 2
             jj = jj+1;
             dPdD1(ii,jj)=dPdD(ib,jb);
             else,end
         end
     else, end
end
rank(dPdD1);
%Anew=(dPdD1'*dPdD1)^-1*dPdD1';
Anew=dPdD1'*(dPdD1*dPdD1')^-1;

ii=0;
for ib=1:nbus
     if kb(ib)==0
     ii = ii+1;
      jj=0;
         for jb=1:nbus
             if kb(jb) == 0
             jj = jj+1;
             dQdV(ii,jj)=dPdD(ib,jb);
             else,end
         end
```

```
     else, end
end

rank(dQdV);
%AAnew=(dQdV'*dQdV)^-1*dQdV';
AAnew=dQdV'*(dQdV*dQdV')^-1;

ii=0;
for ib=1:nbus
     if  kb(ib) == 2
     ii = ii+1;
      jj=0;
         for jb=1:nbus
             if kb(jb) == 0 | kb(jb) == 2
             jj = jj+1;
             dPdD2(ii,jj)=dPdD(ib,jb);
             else,end
         end
     else, end
end

rank(dPdD2);
%Anew2=(dPdD2'*dPdD2)^-1*dPdD2';
Anew2=dPdD2'*(dPdD2*dPdD2')^-1;


for i=1:nbus
   dQdD(i,i)=0;
   for j=1:nbus
      if i==j
         dQdD(i,i)= dQdD(i,i)+initV1(i)*initV1(j)*G(i,j);
         dQdD(i,i)=dQdD(i,i)-initV1(i)*G(i,i);
      else
         dQdD(i,j)=-initV1(i)*G(i,j);
      end
   end
end
```

```
% To define the equality constraint
ii=0;
for ib=1:nbus
     if kb(ib) == 1
     ii = ii+1;
      jj=0;
          for jb=1:nbus
               if kb(jb) == 0 | kb(jb) == 2
               jj = jj+1;
               L(ii,jj)=dPdD(ib,jb);
               else,end
          end
     else, end
  end

EQ=L*Anew2;

EQ1=[1 -EQ];
Eq=EQ1;


% Inequality constraints
 Pin=eye(ng);
 ii=0;
for ib=1:nbus
     if kb(ib) == 0
     ii = ii+1;
      jj=0;
          for jb=1:nbus
               if kb(jb) == 0 | kb(jb) == 2
               jj = jj+1;
               dQdD1(ii,jj)=dQdD(ib,jb);
               else,end
          end
     else, end
  end
```

```
K=AAnew*(-dQdD1)*Anew;

% To define w
w=[Pin;Anew;K];
ww=w;

 % To define cost functions
 Tobj=diag(r);

 % To calculate matrix Eqsq and s

 Eqsq=Eq'*Eq;

 % Simulation of neural network

% To choose the initial conditions for the increment of generated
  powers

 x=[ 0.4061
    -0.3000
    -0.1750
     0.0250
     0.0000
     0.0600];

%for i=1:ng
%   temp1(i)=2*(x(i)-dPmin(i))/(dPmax(i)-dPmin(i));
%    temp2(i)=atanh(temp1(i)-1);
%    u(i)=temp2(i)/lambda;
%end

 for i=1:ng
   temp1(i)=(x(i)-dPmin(i))/(dPmax(i)-dPmin(i));
   u(i)=(-1/lambda)*log((1/temp1(i))-1);
 end


 initPnew=initP;
 initDnew=initD;
 initVnew=initV;
```

```
  N=1;
  while N < 10
% To define x
for i=1:ng
   x(i)=(dPmax(i)-dPmin(i))*logsig(lambda*u(i)) + dPmin(i);
end

      x=[x(1);x(2);x(3);x(4);x(5);x(6)]

for i=1:59
   if w(i,:)*x<=constrmax(i)
      g(i)=0;
   else
      g(i)=1;
   end
end
g=g';

for i=1:59
   wnew(i,:)=w(i,:)*g(i);
end


for i=1:59
   if ww(i,:)*x>=constrmin(i)
      h(i)=0;
   else
      h(i)=1;
   end
end
h=h';


for i=1:59
   wwnew(i,:)=ww(i,:)*h(i);
end
```

```
% To calculate W and V
W=wnew'*wnew;
WW=wwnew'*wwnew;
V1=wnew'*constrmax;
V2=wwnew'*constrmin;

To update generated powers, voltage phase angles and voltage
magnitudes.

initPnew=initPnew+x;
initDnew=initDnew+Anew*x;
initVnew=initVnew+K*x;

% To define ib
  for i=1:ng
     ib(i)=(q(i)+2*r(i)*initPnew(i));
  end
  ib=[ib(1);ib(2);ib(3);ib(4);ib(5);ib(6)];

To update the limits of dP, dD and dV.

dPmin=Pmin-initPnew;
dPmax=Pmax-initPnew;
dDmin=Dmin-initDnew;
dDmax=Dmax-initDnew;

dVmin=Vmin-initVnew;
dVmax=Vmax-initVnew;

constrmin=[dPmin;dDmin;dVmin];
constrmax=[dPmax;dDmax;dVmax];


% To update u
  temp = -(A*Tobj+B*Eqsq+C*W+Delta*WW)*x;
  temp = temp - .5*A*ib + V1*C + V2*Delta ;
```

```
  u = u + temp*dt;
  N=N+1;
end
```

```
% To calculate optimal power conditions and minimal cost obtained.
```

To print out the optimal condition of generated powers and minimal cost obtained.

```
   Power=100*(initPnew)
   cost=0.0;
   initcost=0.0;
   for i=1:ng
      cost=cost+p(i)+q(i)*10^-2*Power(i)+r(i)*10^-4*(Power(i))^2;
      initcost=initcost+p(i)+q(i)*initP(i)+r(i)*(initP(i))^2;
   end
   cost
   initcost

initsigmaP=100*sum(initP);


sigmaP=0.0;
for i=1:ng
   sigmaP=sigmaP+Power(i);
end
sigmaP
```

To print out the result of voltage phase angles in degrees.

```
D=initDnew;
D=D*180/pi
```

To print out the voltage magnitudes obtained in p.u.

```
V=initVnew
```