

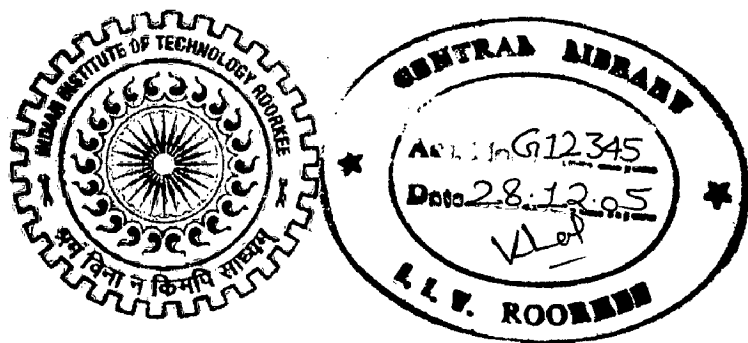
ON LINE POWER SYSTEM STATE ESTIMATION

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*
of
MASTER OF TECHNOLOGY
in
ELECTRICAL ENGINEERING
(With Specialization in Power System Engineering)

By

SATISH KUMAR SINGH



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
JUNE, 2005

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation entitled "**Online Power System State Estimation**" submitted in partial fulfillment of the requirements for the award of the degree of Master of Technology with specialization in Power System Engineering in the Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out from July 2004 to June 2005 under the guidance of **Dr. J. D. Sharma**, Professor, Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee.

I have not submitted the matter embodied in this report for the award of any other degree or diploma.

Date: 13th June 2005


Place: Roorkee

Satish K. Singh

(SATISH KUMAR SINGH)

CERTIFICATE

This is to certify that the above statement made by the candidate is true to the best of my knowledge and belief.


(J. D. Sharma,)

Professor,

Department Of Electrical Engineering,
Indian Institute of Technology Roorkee,
Roorkee-247667, India.

ACKNOWLEDGEMENTS

I express my foremost and deepest gratitude to **Dr. J. D. Sharma**, Professor, Department of Electrical Engineering, Indian Institute of Technology Roorkee, for his valuable guidance, support and motivation throughout this work. I have deep sense of admiration for his innate goodness and inexhaustible enthusiasm. The valuable hours of discussion and suggestions that I had with him have undoubtedly helped in supplementing my thoughts in the right direction for attaining the desired objective. I consider myself extremely fortunate for having got the opportunity to learn and work under his able supervision over the entire period of my association with him.

My sincere thanks to and all faculty members of Power System Engineering for their constant encouragement, caring words, constructive criticism and suggestions towards the completion of this work successfully.

My special thanks go to Mr. N. P. Patidar, Mr. Dheeraj Khatod, Ms. Damanjeet Kaur, Ph.D. Scholars, Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee for their valuable suggestion and comments. I'm grateful to my friends for their moral support and encouragements.

Last but not least, I'm highly indebted to my parents and family members, whose sincere prayers, best wishes, moral support and encouragement have a constant source of assurance, guidance, strength and inspiration to me.

(SATISH KUMAR SINGH)

ABSTRACT

The real time implementation of state estimation algorithms is always desirable for controlling the operation of a power system, but it is limited by computation speed when conventional methods are used. In this work a neural network based method is developed for solving state estimation problem, with constraints emanating due to physical restrictions on the network parameters. The neural network method for solving state estimation problem can be implemented on dedicated neural network processors or reconfigurable hardware to outperform their software implementation. The feasibility of hardware implementation is demonstrated by implementing the algorithm on dedicated neural network processor (NNP) architecture, to solve a simple nonlinear programming problem.

The prerequisite for state estimation is that the system under study must be observable i.e. there are sufficient measurements. A new method based on Hopfield neural network is developed to determine topological observability of power networks. The algorithm also determines where the meters should be placed in order to get an observable system.

FACTS devices, now days are becoming essential part of transmission systems, for optimal use of transmission capacities, which offer several advantages, in the system, so there is requirement of such estimators which not only estimate the voltage magnitude and phase angle but also FACTS device control parameters. Thus the state estimation algorithms must incorporate FACTS devices. In this work FACTS devices are also included in the neural network based, state estimation algorithm.

CONTENTS

	PAGE NO (S)
CANDIDATE'S DECLARATION	I
ACKNOWLEDGEMENTS	II
ABSTRACT	III
CONTENTS	IV
CHAPTER-1	1-8
INTRODUCTION	
1.1 GENERAL	1
1.2 OBJECTIVE OF THE WORK	5
1.3 LITERATURE REVIEW	5
1.4 REPORT ORGANISATION	7
CHAPTER-2	9-22
STATIC STATE ESTIMATION OF POWER SYSTEMS: STATE OF ART	
2.1 INTRODUCTION	9
2.1.1 ERRORS	11
2.1.2 MEASUREMENTS	11
2.2 WEIGHTED LEAST SQUARE ESTIMATION (WLSE) THEORY	12
2.3 STATE ESTIMATION OF AN AC NETWORK	14
2.4 EXISTING STATE ESTIMATION METHODS	18
CHAPTER-3	23-30
HOPFIELD NEURAL NETWORK	
3.1 INTRODUCTION	23
3.2 HOPFIELD NETWORKS	23
3.3 THE MODIFIED HOPFIELD NEURAL NETWORK APPROACH FOR SOLVING OPTIMIZATION PROBLEMS	28
CHAPTER-4	31-46
A MODIFIED HOPFIELD NEURAL NETWORK FOR DETERMINATION OF TOPOLOGICAL OBSERVABILITY IN POWER SYSTEMS	
4.1 INTRODUCTION	31
4.2 NETWORK OBSERVABILITY	33
4.2.1 FORMULATION OF TOPOLOGICAL OBSERVABILITY PROBLEM	35
4.3 NEURAL NETWORK METHODOLOGY	36
4.3.1 NEURAL NETWORK FOR HANDLING INEQUALITY	

CONSTRAINTS IN TOPOLOGICAL OBSERVABILITY PROBLEM	38
4.4 ALGORITHM	41
4.5 RESULTS AND DISCUSSIONS	43
CHAPTER-5	47-64
A MODIFIED HOPFIELD NEURAL NETWORK METHOD FOR EQUALITY CONSTRAINED STATE ESTIMATION	
5.1 INTRODUCTION	47
5.2 STATE ESTIMATION WITH CONSTRAINTS	48
5.3 THE MODIFIED HOPFIELD NEURAL NETWORK	51
5.4 FORMULATION OF STATE ESTIMATION PROBLEM	54
5.4.1 ESTIMATION ALGORITHM	56
5.5 TEST RESULTS	57
CHAPTER-6	65-88
STATE ESTIMATION OF POWER SYSTEMS EMBEDDED WITH FACTS DEVICES	
6.1 INTRODUCTION	65
6.2 FLEXIBLE AC TRANSMISSION SYSTEMS (FACTS)	65
6.2.1 ROLE OF FACTS DEVICES IN POWER SYSTEMS	66
6.2.2 BASIC TYPES OF FACTS CONTROLLERS	67
6.3 UNIFIED POWER FLOW CONTROLLERS	71
6.3.1 STATIC MODELLING OF UPFC	73
6.4 LOAD FLOW CALCULATION OF POWER SYSTEM WITH UPFC	75
6.4.1 MODIFIED NONLINEAR POWER FLOW EQUATIONS	77
6.4.2 ALGORITHM FOR SOLVING LOAD FLOW EQUATIONS	80
6.4.3 RESULTS OF LOAD FLOW	80
6.5 FORMULATION OF STATE ESTIMATION PROBLEM FOR POWER SYSTEM EMBEDDED WITH UPFC	84
6.6 RESULTS AND DISCUSSIONS	85
CHAPTER-7	89-96
IMPLEMENTATION OF MODIFIED HOPFIELD NEURAL NETWORK METHOD ON NEURAL NETWORK PROCESSOR FOR SOLVING NONLINEAR PROGRAMMING PROBLEM.	
7.1 INTRODUCTION	89
7.2 THE MODIFIED HOPFIELD NETWORK	90
7.3 FORMULATION OF THE NONLINEAR OPTIMIZATION PROBLEM	90
7.4 COMPUTATIONAL PARADIGM	91
7.4.1 COMPUTATIONAL PARADIGM AND LANGUAGE	92

7.5 NEURAL NETWORK PROCESSOR (NNP)	92
7.6 EXAMPLE	
CHAPTER-8	97
CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK	
REFERENCES	101
APPENDIX	108

CHAPTER-1

INTRODUCTION

1.1 GENERAL

State estimation is the process of assigning values to unknown system state variables, based on measurements from that system. Usually, the process involves imperfect measurements that are redundant and the process of estimating the system states is based on a statistical criterion that estimates the true value of the state variables to minimize or maximize the selected criteria. A commonly used and familiar criterion is that of minimizing the sum of the squares of the differences between the estimated and “true” (i.e. measured) values of the system parameters.

In a power system, state variables are the voltage magnitudes and relative phase angles at the system buses. Measurements are required in order to estimate the system performance in real time for both system security control and constraints on economic dispatch. The inputs to an estimator are imperfect (noisy) power system measurements of voltage magnitudes, power, VAR, or ampere flow quantities. The estimator is designed to estimate the “best estimate” of the system voltage and phase angles recognizing that there are errors in the measured quantities and there may be redundant measurements. The output data is then used in system control centres in the implementation of the security constrained dispatch and control of the system.

Load dispatcher in power system control centre is required to know at all times the value of voltages, currents and power throughout the network. Some of the values such as bus voltage magnitude and power line flows can be measured within a certain degree of variance. Difficulties are further encountered when some of the data is missing either due to meter being out of order or missing transmission.

State Estimation (SE) utilizes the available redundancy, for systematic cross checking of the measurements, to approximate the states as well as generate information in respect of missing observations or gross measurement errors called bad data. The prerequisite for state estimation is that the system must be observable with the available measurements.

States of a power system can also be computed with the load flow calculations, based on equal number of measurements, assuming them to be accurate. However, the implicit error will lead to imperfect data base and prejudice the security monitoring, whereas ,the state estimator is a data processing algorithm for use on a digital computer to transform meter readings (measurement vector) into an estimate of the system’s states (state vector), which is not only accurate but best reliable also. A comparison between Load Flow Calculation and State Estimation has been shown in Fig.1.1.

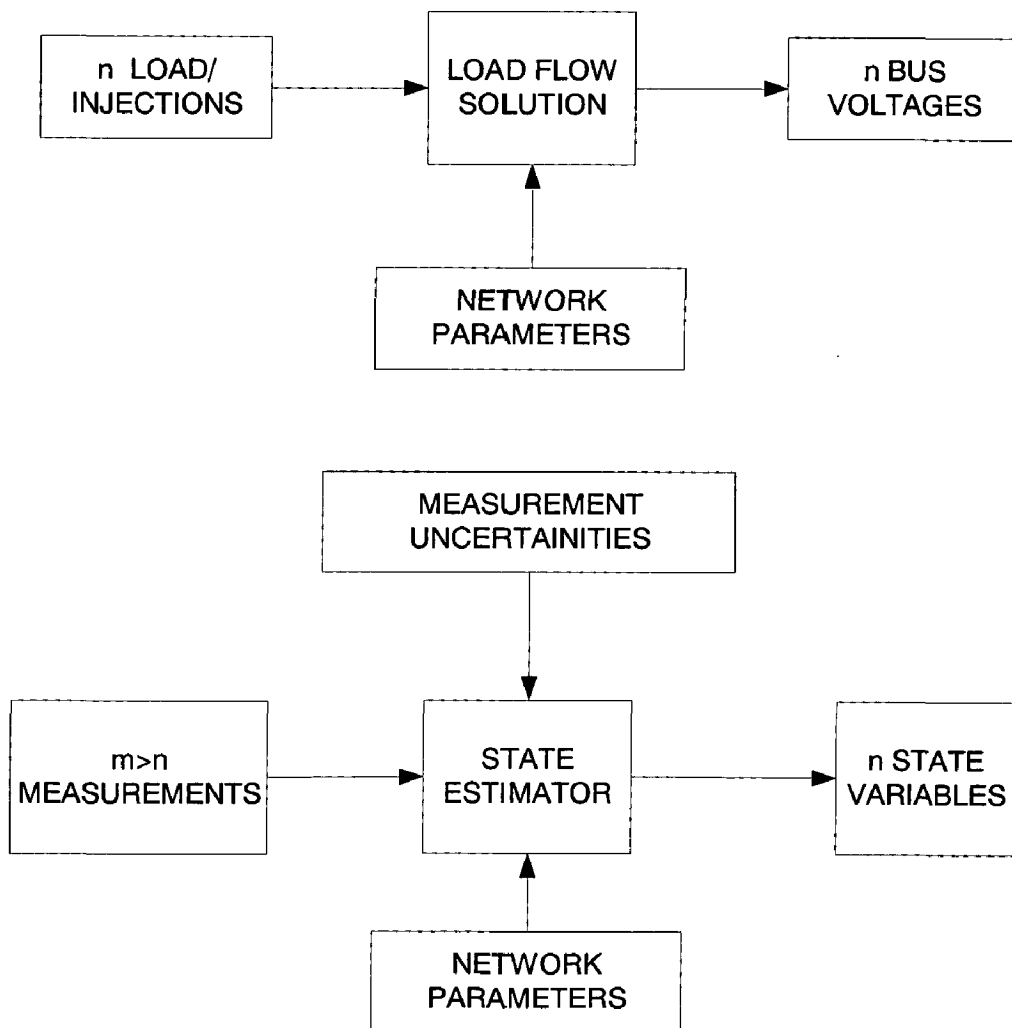


Fig 1.1: Comparison between load flow calculations and state estimation.

As, the state estimator, is required to cater for the needs of online application, computation speed plays a vital role especially when systems are large. Alternate

methods of state estimation are being reported to optimize on (i) numerical stability, (ii) computation efficiency, and (iii) implementation complexity [1].

APPLICATION OF POWER SYSTEMS STATE ESTIMATION

In real time environment the state estimator consists of different modules such as network topology processor, observability analyzer, state estimator and bad data processor.

Fig.1.2 is a schematic diagram showing the information flow between the various functions to be performed in an operation control centre computer system. The system gets its information about the power system from remote terminal units that encode measurement transducer output and opened/closed status information into digital signals that are transmitted to the operations centre over communication circuits. In addition, the control centre can transmit control information such as raise/lower commands to generators and open/close commands to circuit breakers and switches. The analog measurements of generator output must be used directly by the AGC program whereas all the other data will be processed by the state estimator before being used by other programs.

In order to run the state estimator, we must know how the transmission lines are connected to the load and generator buses, this information is called network topology. Since the breakers and switches in any substation can cause the network topology to change, a program must be provided that reads the telemeter breaker/switch status indications and restructures the electrical model of the system.

As seen in the Fig.1.2, the electric model of the power system's transmission system is sent to the state estimator program together with the analog measurements. The output of the state estimator consists of all bus voltage magnitudes and phase angles, transmission line MW and MVAR flows calculated from the bus voltage magnitude and phase angles, and bus loads and generations calculated from the line flows. These quantities, together with the electric load model developed by the network topology program provide the basis for the economic dispatch program.

The output of the state estimator i.e. $|V|, \delta, P_{ij}, Q_{ij}$ together with latest model form the basis for the economic load dispatch or minimum emission dispatch,

contingency analysis program etc.

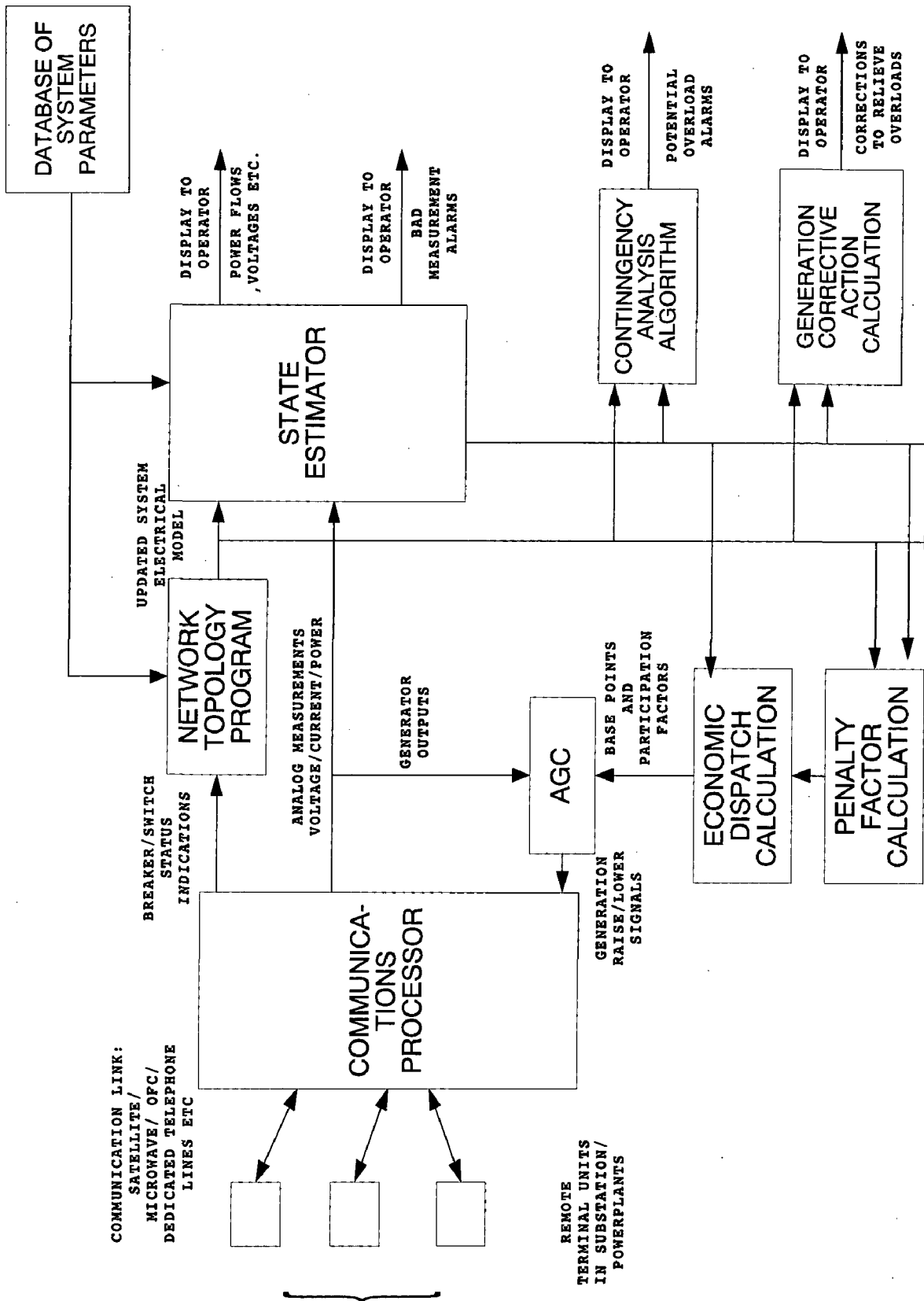


Fig. 1.2 Energy control centre system security schematic

1.2 OBJECTIVE OF THE WORK

From the literature review it is found that state estimation computations are reaching a limit as far as conventional computer based algorithms are concerned. It is therefore required to find out newer methods for state estimation, which can be implemented on hardware to outperform their software counterpart. The algorithms based on neural networks can easily be implemented on dedicated hardware.

The objective of this work is:

- To formulate the power system topological observability problem as an integer programming problem and to develop a new methodology based on Hopfield neural network for determination of topological observability in power system networks.
- To formulate power system state estimation problem as a constrained nonlinear programming problem.
- To apply a method based on modified Hopfield network, where the objective function and constraints are handled in two stages to solve constrained state estimation problem.
- To select a suitable mathematical model of a UPFC for studying the static performance of power systems, and to modify the N-R load flow method to incorporate UPFCs.
- To carry out state estimation of power system embedded with FACTS devices, with modified Hopfield neural network.
- To implement modified Hopfield neural network methodology on neural network processor (NNP) to solve nonlinear programming problems.

1.3: LITERATURE REVIEW:

1.3.1: State Estimation:

Power system state estimation was introduced by Schweppe et al. [2] in 1969. The state estimation problem is usually mathematically formulated as a weighted least squares (WLS) problem [2]. Traditionally, it is solved iteratively by the normal equations

(NE) approach. The great advantage of the NE method is that the gain matrix can be easily factorized using well-known sparsity techniques. The implementation of robust state estimation methods has received significant attention [3]–[4]. The NE approach can handle zero injections, by assigning much larger weights for the zero injection equations. The artificially large weights may cause ill conditioning problems, thus degrading the convergence. Orthogonal transformation based methods [5]–[7] have been used to alleviate the ill conditioning problems. The formulation of WLS with equality constraints has been suggested to handle zero-injections in [8]. A method based on direct elimination of variables using the equalities is suggested in [9]. In [10], the so-called Hachtel’s augmented matrix method for solving least squares problems has been applied. In [11], it is shown analytically that the Hachtel’s method is numerically more robust than the method of NE with equality constraints. An extension of the Hachtel’s method, based on a convenient blocked sparse formulation and the use of block arithmetic in matrix operations, is presented in [12] and [13]. Although NE/C and Hachtel’s augmented methods are robust, their weakness is that the corresponding gain matrices are symmetric but indefinite and the factorization routines must be modified to handle zero pivots during Gaussian elimination. In [14], a method for enforcing equality and limit constraints in WLS state estimation is presented.

Estimating the state is only a part of the larger function of the state estimation. Other related functions include observability analysis, bad data processing, and external network modeling. In 1975, Clements and Wollenberg undertook the studies on topological observability [15]. Since then, a lot of the studies have been made. Quintana, et al. developed a graph-theory based method [16]. Monticello and Wu presented an approach with the Gauss Elimination [17]. After that, Bargiela, et al., contributed to developing a technique using the maximum flow [18]. Recently, Mori and Tsuzuki proposed artificial neural networks-based approaches [19], [20].

1.3.2 Hopfield Neural network:

In 1985, Hopfield and Tank [21] presented Hopfield networks to solve optimization problem. Wilson and Pawley [22] first published the shortcomings of the Hopfield network nearly three years after Hopfield and Tank’s original paper was published. In the original Hopfield networks a number of penalty parameters need to be

fixed before each simulation of the network, yet the values of these parameters that will enable the network to generate valid solutions are unknown. The problem of optimally selecting these parameters is not unimportant and much work has been done to try to facilitate this process [23]. Perhaps the most important breakthrough in the field, however, came from the valid subspace approach of Aiyer et al. [24]. By representing all of the constraints by a single term, feasibility of the Hopfield network can now be essentially guaranteed. Recently a modified Hopfield network has been proposed to solve nonlinear optimization problems [25]. The constraints of the problem are not included in the network energy function rather they are handled by valid subspace technique.

1.3.3. FACTS devices:

The basic idea about the FACTS devices have been given in the books by N.G. Hingorani et al [26], Verma et al [27] and Y. H. Song et al [28]. A Detailed Explanation has been given for all the FACTS devices. A. Edris [29] gives the history of emergence of FACTS and the recent developments in the field. L. Gyugyi [30] gives the concept of unified power flow control (UPFC) and its versatility to control almost all the parameters in the system. L. Gyugyi et al [31] presents the control behavior of unified power flow controllers (UPFC) and relation between real and reactive power flowing between the lines in which UPFC is connected.

1.3.4. Load flow with Facts devices

The induction of FACTS devices increases the complexity of system and hence the problem arises in the analysis of the system. L.L. Freris et al [32] review the problem arising in load flow of the system with the induction of FACTS devices. Y. H. Song, et al [33] proposes the Power injection modeling model of the UPFC for the load flow studies. In [34] Hatchel method is applied for solving state estimation of power systems in presence of UPFC.

1.4 REPORT ORGANIZATION

In chapter 2, a review of state estimation is presented along with brief description of the existing algorithms.

Chapter 3 presents the concept of Hopfield neural network together with its dynamic behavior.

In chapter 4, the problem of topological observability of power system is formulated as an integer programming problem and a new algorithm based on Hopfield neural network is presented to solve it.

Chapter 5 presents a modified Hopfield neural network methodology for solving equality constrained state estimation problem.

In chapter 6, a review of various FACTS controllers, which exist around the world and detailed model structure of UPFC, has been given. The algorithm for modified N-R to incorporate UPFC has been presented, and finally state estimation of the power system in presence of UPFC has been carried out by the modified Hopfield neural network methodology.

Chapter 7 presents implementation of the modified Hopfield neural network algorithm on a dedicated neural network processor.

In chapter 8, conclusions and the scope for future research are given.

CHAPTER-2

STATIC STATE ESTIMATION OF POWER SYSTEMS: STATE OF ART

Load flow calculations indeed are an inevitable tool for off-line studies and planning exercises, but incomplete and erroneous measurement is a real time proposition. Solution for such a situation is provided by static state estimator, which ignores the slow changes in the system and utilizes redundant set of measurements for cross checking and approximating the most reliable estimates of the system state.

2.1 INTRODUCTION

Modern electric power systems are enormous in scale, typically spanning continents, and provide services for hundreds of millions of customers. These systems consist of numerous interconnected networks that contain various types of generators and consumers of electrical energy, which are also interconnected by high voltage equipment such as transmission lines and transformers. Each of these networks is operated and maintained by companies who are responsible for the consistently secure and economic operation of their network, as well as for the reliability of the larger power system. The famous 1965 blackout in the Northeastern United States awakened the power community to the fact that their means of providing this service would require implementation of more thorough and advanced techniques.

Prior to this, power system operators working at the control centers had only a minimal amount of information and controls available to achieve these objectives. The only information received was that which is essential to control the real-time network's basic operation, such as system frequency, breaker statuses, and a minimal set of active power measurements.

In 1969 Fred Schweppe introduced the idea of using the redundant number of measurements, made available by the supervisory control and data acquisition (SCADA) system to statistically determine the state of the network. His proposition, the state estimator [2], was eventually accepted and serves as a basis for static state estimation.

The goal of control center design is security control under the three states of power system operation: the normal, emergency, and restorative states. There are several functions performed by a control center in the classical energy management system (EMS) environment, and the most difficult of these to implement are those that run in a real-time environment. The key functions include state estimation, security monitoring, on-line load flow, security analysis, supervisory control, automatic generation control, automatic voltage/VAR control, and economic dispatch control [35]. These functions often interact in a complex manner, but all are aimed at providing the system operator with a coherent view of the system and/or carrying out the operator's decisions. Since all of these functions are directly dependent on state estimation, it is essential that the system operator trust the result. The classical model as discussed above assumes steady-state functions; an introductory overview of dynamic-state functions is provided in [36].

Though power systems are dynamic, real-time systems, dynamic state estimation is generally not employed, reasons being: static state estimation presently fills the control center needs; there are many difficulties in determining the dynamic system model; and dynamic state estimation is computationally intensive[37]. Of these, the difficulty of defining a tractable, reliable model of the dynamic power system is the biggest inhibitor, due to the highly unpredictable and nonlinear nature of power systems.

The state estimator plays the essential role of a purifier, creating a complete and reliable database for security monitoring, security analysis and the various controls of a power system. The state estimator thus employs statistical methods to act as a tunable filter between the field data measurements and security and control functions.

The fundamental equation for the problem of power system state estimation (SE) can be formulated as

$$z = h(x) + e \quad (2.1)$$

Where z represents all measurements, including power injection, power flow and bus voltage magnitude measurements, e is the measurement noise vector, x is the state vector composed of the phase angles and magnitudes of the voltages at network buses, and $h(\cdot)$ stands for the nonlinear measurement functions in terms of state variables. It is always assumed that the parameters and observability of the systems are already determined in advance.

2.1.1: ERRORS

2.1.1.1 Measurement Error:

The process of telemetering the measurements to the control centre is disposed to the following discrepancy:

- (i) Error in transducer calibration
- (ii) Noise in communication channel
- (iii) Non-simultaneity of the data

And defaults-

- (i) Failure of the communication channel
- (ii) Meter being defective or out of order.

2.1.1.2 Modeling Error

The model of the network constitutes its topology and parameters. The topological errors can be caused by missing information in respect of disconnected or reconnected line, while parameter errors are due to its wrong initial estimation. Moreover, the system representation considered in such studies is single phase, while unbalance conditions cause significant error.

2.1.2 MEASUREMENTS

Non-availability of measurements may create condition of unobservability and therefore it is important to maintain sufficient redundancy. This leads to the following classification of measurements.

- **Telemetered measurements:** On-line telemetered bus voltage magnitudes, active and reactive power flows, active and reactive injections, subject to noise or error in metering, communication system etc. They are assigned weightings in inverse proportion of their variance.
- **Pseudo measurements** are the guesses in respect of generation or substation loads based on historical data and are assigned least weightings. It is used in the event of missing data or bad data.
- **Virtual Measurements** also known as zero injections, which contain no error. They are associated with network buses that have neither load nor generation and thus their injected power is zero, and therefore do not require measurement;

however they are used to create redundancy. These measurements are assigned highest weighting.

The measurements are never simultaneous, they are sequential, however at a very close interval and therefore the static state estimator assumes it to be snap-shot measurement [2], i.e. all measurements are assumed to be taken simultaneously.

2.2: WEIGHTED LEAST SQUARE ESTIMATION (WLSE) THEORY:

It is often desirable to put different weightings on the different components of measurements since some of the measurements may be more reliable and accurate than the others and should be given more importance. If a single parameter x , is estimated using N_m measurements, the WLSE problem can be described as:

$$\min J(x) = \sum_{i=1}^{N_m} \frac{[z_i^{\text{meas}} - h_i(x)]^2}{\sigma_i^2} \quad (2.2)$$

where

σ_i =variance for the i^{th} measurement

$J(x)$ =measurement residual

N_m =number of independent measurements

z_i^{meas} = i^{th} measured quantity

If N_s unknown parameters are to be estimated using N_m measurements, the estimation problem can be described as:

$$\min J(x_1, x_2, \dots, x_{N_s}) = \sum_{i=1}^{N_m} \frac{[z_i^{\text{meas}} - h_i(x_1, x_2, \dots, x_{N_s})]^2}{\sigma_i^2} \quad (2.3)$$

Matrix Formulation

If functions $h_i(x_1, x_2, \dots, x_{N_s})$ are *linear functions* then

$$h_i(x_1, x_2, \dots, x_{N_s}) = h_i(x) = h_{i1}x_1 + h_{i2}x_2 + \dots + h_{iN_s}x_{N_s} \quad (2.4)$$

$$h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_{N_s}(x) \end{bmatrix} = [H]x \quad (2.5)$$

Where, $[H]$ is an $(N_m \times N_s)$ matrix containing the coefficient of the linear functions $h(x)$.
Placing the measurements in a vector form:

$$\mathbf{Z}^{\text{meas}} = \begin{bmatrix} Z_1^{\text{meas}} \\ Z_2^{\text{meas}} \\ \vdots \\ Z_{N_m}^{\text{meas}} \end{bmatrix} \quad (2.6)$$

Eqn. (2.3) may be written in a very compact form as:

$$\min J(x) = [\mathbf{z}^{\text{meas}} - f(x)]^T [\mathbf{R}^{-1}] [\mathbf{z}^{\text{meas}} - f(x)] \quad (2.7)$$

$$\text{Where } [\mathbf{R}] = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_{N_s}^2 \end{bmatrix}$$

$[\mathbf{R}]$ is called covariance matrix of measurement errors.

To obtain the general expression for the minimum in Eqn. (2.7), expand the expression and substitute $[H]x$ for $f(x)$ from Eqn. (2.5):

$$\min J(x) = \left\{ \mathbf{z}^{\text{meas}T} [\mathbf{R}^{-1}] \mathbf{z}^{\text{meas}} - \mathbf{x}^T [\mathbf{H}]^T [\mathbf{R}^{-1}] \mathbf{z}^{\text{meas}} - \mathbf{z}^{\text{meas}} [\mathbf{R}^{-1}] [\mathbf{H}] \mathbf{x} + \mathbf{x}^T [\mathbf{H}]^T [\mathbf{R}^{-1}] [\mathbf{H}] \mathbf{x} \right\}$$

The minimum of $J(x)$ is found when $\frac{\partial J(x)}{\partial x_i} = 0$, for $i=1 \dots N_s$; this is identical to the stating that the gradient of $J(x)$, $\nabla J(x)$ is exactly zero.

The gradient of $J(x)$ is

$$\nabla J(x) = -2[\mathbf{H}]^T [\mathbf{R}^{-1}] \mathbf{z}^{\text{meas}} + 2[\mathbf{H}]^T [\mathbf{R}^{-1}] [\mathbf{H}] x$$

Then, $\nabla J(x) = 0$ gives.

$$\mathbf{x}^{\text{est}} = \left[[\mathbf{H}]^T [\mathbf{R}^{-1}] [\mathbf{H}] \right]^{-1} [\mathbf{H}]^T [\mathbf{R}^{-1}] \mathbf{z}^{\text{meas}} \quad (2.8)$$

It is to be noted here that Eqn. (2.8) holds when $N_s < N_m$, that is, when the number of parameters being estimated is less than the number of measurements being made.

When $N_s = N_m$, the estimation problem reduces to

$$x^{est} = [H]^{-1} z^{meas} \quad (2.9)$$

In power system state estimation, underdetermined problems (i.e., where $N_s > N_m$) are solved by adding pseudo measurements to the measurement set to give a completely determined ($N_s = N_m$) or over determined ($N_s < N_m$) problem,

2.3. STATE ESTIMATION OF AN AC NETWORK

In the Ac network, the measured quantities are MW, MVAR, MVA, amperes, transformer tap position, and voltage magnitudes. The state variables are the voltage magnitude at each bus and the phase angles at all but the reference bus. The equation for power flowing over a transmission line is clearly not a linear function of the voltage magnitude and phase angle at each end of line. Therefore, the $h(x)$ functions will be nonlinear functions; except for a voltage magnitude measurement where $h(x)$ is simply unity times the particular x_i that corresponds to the voltage magnitude being measured. For MW and MVAR measurements on a transmission line from bus i to bus j , $J(x)$ will contain the following terms:

$$\frac{[MW_{ij}^{meas} - (|E_i|^2 (G_{ij}) - |E_i| |E_j| (\cos(\theta_i - \theta_j) G_{ij} + \sin(\theta_i - \theta_j) B_{ij}))]^2}{\sigma_{MW_{ij}}^2}$$

$$\frac{[MVAR_{ij}^{meas} - (-|E_i|^2 (B_{ij}) - |E_i| |E_j| (\sin(\theta_i - \theta_j) G_{ij} - \cos(\theta_i - \theta_j) B_{ij}))]^2}{\sigma_{MVAR_{ij}}^2}$$

A voltage magnitude measurement would result in the following term in $J(x)$.

$$\frac{(|E_i|^{meas} - |E_i|)^2}{\sigma_{|E_i|}^2}$$

Similar functions can be derived for MVA or ampere measurements.

Since the relationship between the states ($|E|$'s and θ 's) and the power flow in a network is nonlinear, some iterative technique is required to minimize $J(x)$. A commonly used technique for power system state estimation is to calculate the gradient of $J(x)$ and then force it to zero using Newton's method (briefly reviewed below)

Given the functions $g_i(x)$, $i=1, \dots, n$, It is desired to find out x^{ans} that gives $g_i(x^{ans}) = g_i^{des}$, for $i=1, \dots, n$.

Arranging the g_i functions in a vector form,

$$\mathbf{g}^{\text{des}} - \mathbf{g}(\mathbf{x}) = 0 \text{ for } \mathbf{x} = \mathbf{x}^{\text{ans}} \quad (2.10)$$

By perturbing \mathbf{x} , Eqn. (2.10) can be written as

$$\mathbf{g}^{\text{des}} - \mathbf{g}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{g}^{\text{des}} - \mathbf{g}(\mathbf{x}) - [\mathbf{g}'(\mathbf{x})]\Delta\mathbf{x} = 0 \quad (2.11)$$

Where $\mathbf{g}(\mathbf{x} + \Delta\mathbf{x})$ have been expanded in a Taylor's series about \mathbf{x} , and all higher order terms are ignored. The $[\mathbf{g}'(\mathbf{x})]$ term is the Jacobian matrix of first derivatives of $\mathbf{g}(\mathbf{x})$.

Then from Eqn. (2.11)

$$\Delta\mathbf{x} = [\mathbf{g}'(\mathbf{x})]^{-1}(\mathbf{g}^{\text{des}} - \mathbf{g}(\mathbf{x})) \quad (2.12)$$

To solve for \mathbf{g}^{des} , the value of $\Delta\mathbf{x}$ is obtained using Eqn. (2.12) and then $\mathbf{x}^{\text{new}} = \mathbf{x} + \Delta\mathbf{x}$ is calculated. Eqn. (2.12) is reapplied until either $\Delta\mathbf{x}$ becomes very small or $\mathbf{g}(\mathbf{x})$ comes close to \mathbf{g}^{des} .

Now returning to the state estimation problem as given in Eqn. (2.2)

$$\min J(\mathbf{x}) = \sum_{i=1}^{Nm} \frac{[z_i^{\text{meas}} - h_i(\mathbf{x})]^2}{\sigma_i^2} ; \text{ The gradient of } J(\mathbf{x}) \text{ is formulated as follows:}$$

$$\begin{aligned} \nabla_{\mathbf{x}} J(\mathbf{x}) &= \begin{bmatrix} \frac{\partial J(\mathbf{x})}{\partial x_1} \\ \frac{\partial J(\mathbf{x})}{\partial x_2} \\ \vdots \\ \vdots \end{bmatrix} \\ &= -2 \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_3}{\partial x_1} & \dots \\ \frac{\partial h_1}{\partial x_2} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_3}{\partial x_2} & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2} \\ \frac{1}{\sigma_2^2} \\ \vdots \\ \vdots \end{bmatrix} \begin{bmatrix} z_1 - h_1(\mathbf{x}) \\ z_2 - h_2(\mathbf{x}) \\ \vdots \\ \vdots \end{bmatrix} \quad (2.13) \end{aligned}$$

The Jacobian of $h(\mathbf{x})$, is given by

$$[H] = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_3}{\partial x_1} & \dots \\ \frac{\partial h_1}{\partial x_2} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_3}{\partial x_2} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (2.14)$$

Eqn. (2.13) can be written as

$$\nabla_x J(x) = \left\{ -2[H]^T [R]^{-1} \begin{bmatrix} z_1 - h_1(x) \\ z_2 - h_2(x) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \right\} \quad (2.15)$$

To make $\nabla_x J(x)$ equal zero, applying Newton's method as in Eqn. (2.12),

$$\Delta x = \left[\frac{\partial \nabla_x J(x)}{\partial x} \right]^{-1} [-\nabla_x J(x)] \quad (2.16)$$

The Jacobian of $\nabla_x J(x)$ is calculated as follows:

$$\begin{aligned} \frac{\partial \nabla_x J(x)}{\partial x} &= \frac{\partial}{\partial x} \left\{ -2[H]^T [R]^{-1} \begin{bmatrix} z_1 - f_1(x) \\ z_2 - f_2(x) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \right\} \\ &= -2[H]^T [R]^{-1} [-H] \\ &= 2[H]^T [R]^{-1} [H] \end{aligned} \quad (2.17)$$

then

$$\Delta x = \frac{1}{2} \left[[H]^T [R]^{-1} [H] \right]^{-1} \left[2[H]^T [R]^{-1} \begin{bmatrix} z_1 - f_1(x) \\ z_2 - f_2(x) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \right] \quad (2.18)$$

$$= \left[[H]^{-1} [R]^{-1} [H] \right]^{-1} [H]^T [R]^{-1} \begin{bmatrix} z_1 - f_1(x) \\ z_2 - f_2(x) \\ \vdots \\ \vdots \end{bmatrix} \quad (2.19)$$

Eqn. (2.19) is applied iteratively to solve the AC state estimation problem.

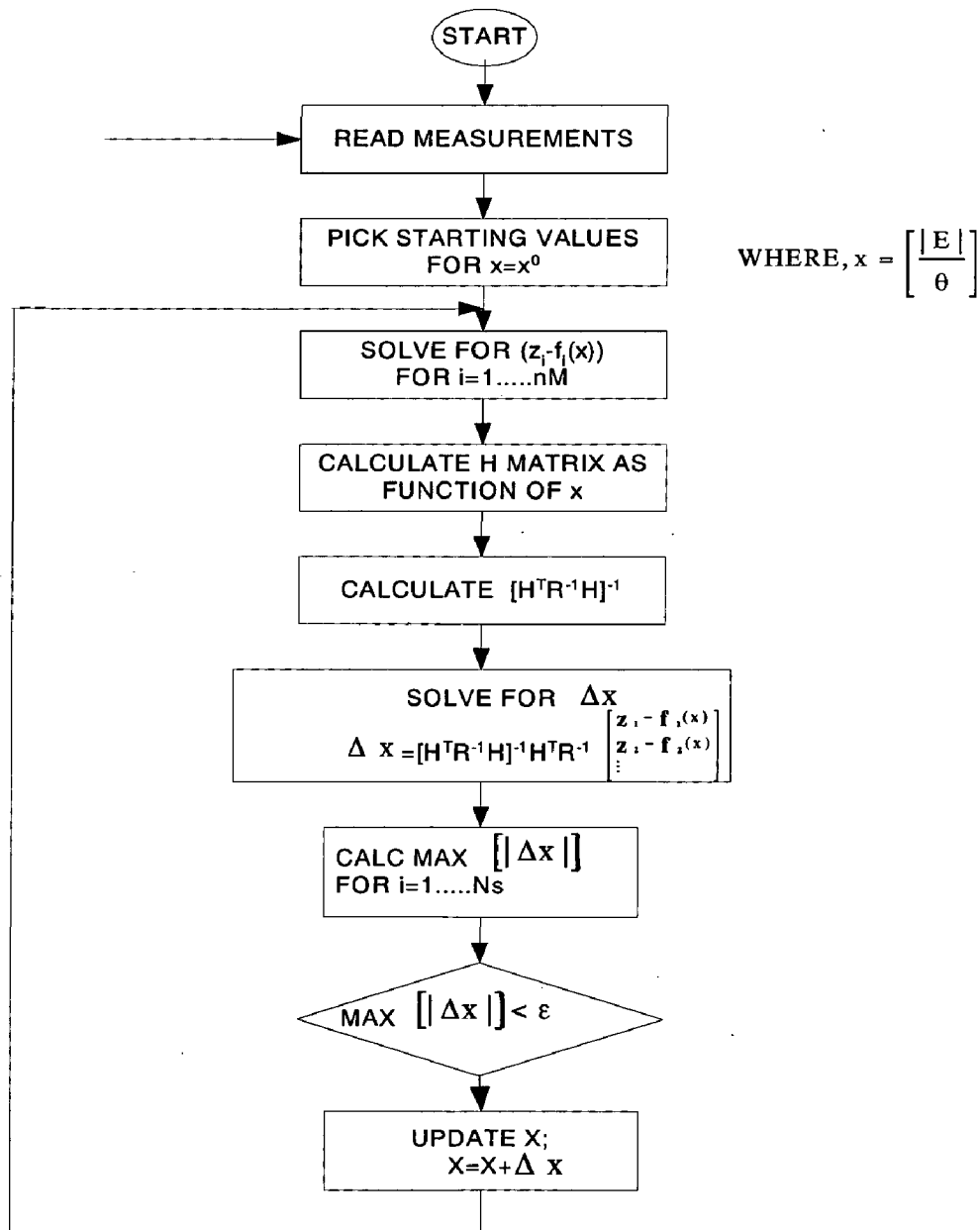


Figure 2.1: State - estimation solution algorithm

2.4: EXISTING STATE ESTIMATION METHODS

1. Normal Equations

The estimate of the state vector x is obtained by minimizing the weighted least square function

$$J(x) = [z - h(x)]^T W [z - h(x)] \tag{2.20}$$

Where W is a diagonal ($N_m \times N_m$) matrix whose elements are the measurement weighting factors. The measurement weighting factor is taken as the reciprocal of the error variance. The estimate is solved by an iterative scheme as discussed in the previous section which computes the corrections Δx at each iteration by solving

$$G(x) \Delta x = H^T(x) W \Delta z \tag{2.21}$$

where

$$\Delta z = z - h(x)$$

$$H(x) = \frac{\partial h(x)}{\partial x} = \text{Jacobian matrix}$$

$$G(x) = H^T(x) W H(x)$$

$$x = x^k \text{ at the } k^{\text{th}} \text{ iteration.}$$

Eqn. (2.21) is the so-called normal equations of the linear weighted least square problem and is solved by first performing sparse matrix triangular factorization of the gain matrix:

$$G = U^T U \dots \dots \dots \tag{2.22}$$

The classical approach for state estimation is the so-called normal equations approach. In this method, different types of measurements are differentiated by the use of different weighting factors in the formulation. In power network there are some nodes with zero injections i.e. switching substations as constant load. Such buses are called constrained buses and can be included in the cost function by assigning large weighting factors. It has been observed empirically that such disparity in weighting factors may cause ill-conditioning. The need to represent equality constraints as pseudo measurements with relatively high weights has caused a search for an alternative to the normal equations approach.

2. Orthogonal transformation [5, 6] (ORTHO)

The objective function of the linear weighted least squares problem is

$$\begin{aligned}
J(\Delta x) &= [\Delta z - H\Delta x]^T W [\Delta z - H\Delta x] \\
&= [\tilde{\Delta z} - \tilde{H}\Delta x]^T [\tilde{\Delta z} - \tilde{H}\Delta x] \\
&= \|\tilde{\Delta z} - \tilde{H}\Delta x\|^2
\end{aligned} \tag{2.23}$$

where, $\tilde{H} = W^{1/2}H$ and $\tilde{\Delta z} = W^{1/2}\Delta z$

The orthogonal transformation method avoids squaring the gain matrix by using the following decomposition of the Jacobian matrix:

Let Q be an orthogonal matrix, i.e. $Q^T Q = I$, such that

$$Q\tilde{H} = \begin{bmatrix} U \\ 0 \end{bmatrix} \tag{2.24}$$

where U is an upper triangular matrix, then

$$\begin{aligned}
J(\Delta x) &= [\tilde{\Delta z} - \tilde{H}\Delta x]^T Q^T Q [\tilde{\Delta z} - \tilde{H}\Delta x] \\
&= \|Q\tilde{\Delta z} - Q\tilde{H}\Delta x\|^2 \\
&= \|\Delta y_1 - U\Delta x\|^2 + \|\Delta y_2\|^2
\end{aligned} \tag{2.25}$$

$$\text{where } Q\tilde{\Delta z} = \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} \tag{2.26}$$

Minimum of Eqn. (2.25) occurs at

$$U\Delta x = \Delta y_1 \tag{2.27}$$

To summarize, the method starts with first performing the orthogonal transformation (2.24) and (2.26) of \tilde{H} and $\tilde{\Delta z}$, and then solve (2.27) by back substitutions.

3. Hybrid method [7] (HYBRID)

It can easily be derived using Eqn. (2.24) that

$$G = H^T W H = \tilde{H}^T \tilde{H} = U^T U \tag{2.28}$$

The hybrid method solves normal equations using the orthogonal factors:

$$U^T U \Delta x = H^T W \Delta z \tag{2.29}$$

Here U is the upper triangular matrix and thus exploits sparsity along with advantages of ORTHO. There are two major steps in the Hybrid method. The first step is to perform

orthogonal transformation Q and the second step is to solve the normal equations.

4. Normal equations with constraints [8] (NE/C)

To account for constraints the measurements are partitioned into telemetered measurements $z = h(x) + e$ and virtual measurements $c(x) = 0$. Consequently the Jacobian is partitioned into H and C . Let the ratio between the weighting factors of the virtual measurements and the telemetered measurements be r . Then the normal equations become:

$$[H^T H + rC^T C] \Delta x = H^T \Delta z + rC^T \Delta c \quad (2.30)$$

For very large r , the second term $rC^T \Delta c$ in the coefficient matrix dominates. However, usually there are not enough virtual measurements to make the matrix C full rank. Therefore for large r , the coefficient matrix in Eqn. (2.30) tends to be singular, causing the ill-conditioning problem.

The problem is to find an estimate of the state vector x which minimizes the weighted least square $J(\Delta x) = [\Delta z - H\Delta x]^T W [\Delta z - H\Delta x]$ while the equality constraints $c(x) = 0$ are satisfied. The method of Lagrange multipliers may be applied to solve the constrained minimization problem. Where the constraints are included in the cost function as

$$J(\Delta x, \lambda) = [\Delta z - H\Delta x]^T W [\Delta z - H\Delta x] + \lambda [\Delta c(x) - C\Delta x]^T \quad (2.31)$$

Where λ represents the Lagrangian Multiplier and $c(x)$ the constraints such that:

$$C(x) = c(\hat{x}) + C \Delta x \quad (2.32)$$

Using optimality condition-

$$\frac{\partial J}{\partial \Delta x} = 0 \Rightarrow H^T W H \Delta x + C^T \lambda = H^T W \Delta z \quad (2.33)$$

$$\text{Here } C = \frac{\partial c(x)}{\partial x} \quad (2.34)$$

Eqns. (2.32) and (2.33) can be written as:

$$\begin{bmatrix} H^T W H & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda \end{bmatrix} = \begin{bmatrix} H^T W \Delta z \\ \Delta c \end{bmatrix} \quad (2.35)$$

5. Hachtel's augmented matrix method [10, 11] (HACHTEL)

The constrained minimization problem may be solved for \bar{x} by Hachtel's augmented matrix method. This method uses Eqs. (2.33) and (2.34), along with error vector expressed as:

$$\Delta r = \Delta z - H\Delta x$$

It can also be written as

$$\alpha W^{-1}(\alpha W\Delta r) + H\Delta x = \Delta z \quad (2.36)$$

Here α is the parameter used to control the numerical stability and W is the weighted diagonal matrix. Eqs. (2.33, 2.34 and 2.36) can be written as:

$$\begin{bmatrix} 0 & 0 & C \\ 0 & \alpha W^{-1} & H \\ C^T & H^T & 0 \end{bmatrix} \begin{bmatrix} -\alpha^{-1}\lambda \\ \alpha^{-1}W\Delta r \\ \Delta x \end{bmatrix} = \begin{bmatrix} \Delta c \\ \Delta z \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & C \\ 0 & \alpha W^{-1} & H \\ C^T & H^T & 0 \end{bmatrix} \begin{bmatrix} \lambda' \\ \Delta r' \\ \Delta x \end{bmatrix} = \begin{bmatrix} \Delta c \\ \Delta z \\ 0 \end{bmatrix}$$

where $\Delta r' = \alpha^{-1}W\Delta r$ and $\lambda' = -\alpha^{-1}\lambda$

$H = \frac{\partial h(x)}{\partial x}$ and $C = \frac{\partial c(x)}{\partial x}$ are the Jacobian matrices, $\Delta z = z - h(x)$; $\Delta c = -c(x)$,

$\Delta r = \Delta z - H\Delta x$ and $x = x^k$ at the k^{th} iteration,

The formulation of state estimation presented here is sometimes called the full WLS version. A fast decoupled version also exists [38], in which (i) the real and reactive power measurements are divided by voltages, (ii) the real and reactive equations are decoupled, and (iii) two constant decoupled sub matrices of H are used throughout the iterations.

CHAPTER-3

HOPFIELD NEURAL NETWORK

This chapter discusses the structure and dynamic behavior of Hopfield neural network together with how the concept of energy minimization can be used to solve optimization problems with it.

3.1 INTRODUCTION

Neural networks are potentially powerful alternative approach for solving science and engineering problems. They work in parallel, with much potential for rapid hardware implementation. It has been found that neural networks, when simulated on a digital machine, are able to obtain near-optimal solutions to practical optimization problems. This finding, coupled with their potential for hardware implementation, makes them an attractive alternative to traditional optimization techniques.

Hopfield neural network evolves by minimizing system energy function. In its original form, the Hopfield energy function involves many parameters, which need to be tuned, and constructing a suitable energy function, which enables the network to arrive at feasible near-optimal solutions, is a difficult task. The original Hopfield neural network, when used for solving optimization problems suffer from problems such as local minima trapping, infeasibility of solution etc. Modifications were made to the Hopfield network to enable escape from local minima, while feasibility of the solutions is ensured [39].

Optimal solution is not as imperative as arriving at a near-optimal solution quickly. Certainly, one of the principal advantages of neural techniques is the rapid computation power and speed, which can be obtained through hardware implementation [40, 41], and this consideration is even more valuable in industrial applications. It has been realized that neural networks hold much potential for rapid solution to many problems, by utilizing their inherent parallelism and hardware implementation ability.

3.2: HOPFIELD NETWORKS

In 1982, John Hopfield described a new way of modeling a system of neurons capable of performing 'computational' tasks. This model considers neural networks of the

brain as a dynamical system, with a feedback mechanism, which can be defined as a set of differential equations.

John Hopfield introduced hardware neural networks from electronic components, and utilized these circuits to solve optimization problems. Operational amplifiers were used to simulate neurons, and the synapses between neurons were simulated by a combination of resistors and capacitors wired between the amplifiers (Fig.3.1). The output of a particular neuron was modeled by the output voltage of the amplifier in question, and an inhibitory impulse from one neuron to another was modeled by connecting the positive terminal of the first amplifier to the negative terminal of the second, an excitatory impulse being modeled by a positive to positive connection. The strength (weight) of a connection between two amplifiers i and j was defined by the value of a resistor connecting the two amplifiers.

The discrete Hopfield network comprises a fully interconnected system of n neurons, each of which is considered to be a binary unit (0 or 1). At each moment; a binary state vector can represent the entire state of the network. Each interconnection has a weight, denoted by W_{ij} from neuron j to i . The Hopfield network considers bidirectionality in the connections, using the symmetric weight matrix, $W_{ij}=W_{ji}$, and that no neuron is connected to itself ($W_{ii}=0$). For each neuron i , there exists a fixed threshold value U_i . Each neuron randomly and asynchronously updates according to the Eqn. (3.1). The neurons are assumed to be threshold logic units so that a state has one of the two possible values. The following is the firing rule of an arbitrary neuron i :

$$V_i = 1 \text{ if } \sum W_{ij}V_j < U_i$$

$$= 0 \text{ if } \sum W_{ij}V_j > U_i$$

Where V_i denotes the output of neuron i and U_i is its threshold. This can be written with a neuron function, $f(\cdot)$:

$$V_i = f\left(\sum_{j=1}^n W_{ij}V_j - U_i\right) \quad (3.1)$$

Where $f(x) = \text{sgn}(x) = 1$ if $x > 0$,
 0 if $x < 0$;

Each network has an associated energy in a quadratic form:

$$E = -\frac{1}{2} \left(\sum \sum W_{ij} V_i V_j + \sum V_i U_i \right) \quad (3.2)$$

The algorithm for the time evolution of the system is based on asynchronous parallel processing.

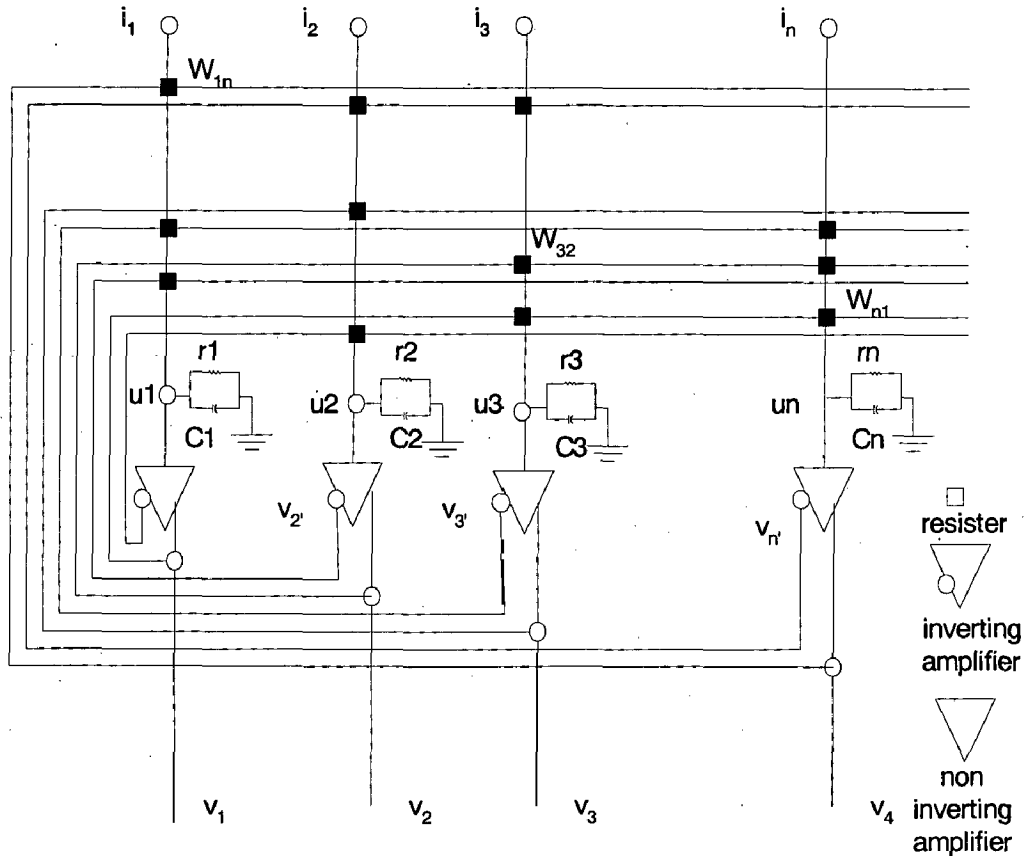


Figure 3.1: An electronic circuit representation of the Hopfield neural network

The continuous Hopfield network, as described in [42], comprises a fully interconnected system of 'n' computational elements or neurons. In the Hopfield model the output of each unit is fed to all the other units with weights W_{ij} , for all i , and j . The internal state of each neuron u_i is equivalent to the weighted sum of the external states of all connecting neurons. The external state of neuron 'i' is given by V_i . An external input, i_i to each neuron 'i' is also incorporated which acts as input bias. The relationship between the internal state of a neuron and its output level in this continuous Hopfield

network is determined by an activation function $g(u_i)$, which is bounded below by 0 and above by 1. Commonly, this activation function is given by

$$v_i = g_i(u_i) = \frac{1}{2} \left(1 + \tanh \left(\frac{u_i}{T} \right) \right)$$

T is a parameter, for controlling the gain (or slope) of the activation function. In the biological system, u_i lags behind the instantaneous outputs, V_j , of the other neurons because of the input capacitance, C_i of the cell membrane, the trans-membrane resistance R_i , and the finite impedance $R_{ij} = |W_{ij}|^{-1}$ between the output V_j and the cell body of neuron i . Thus, the following resistance-capacitance differential equation determines the rate of change of u_i , and hence the time evolution of the continuous Hopfield network:

$$C_i \left(\frac{du_i}{dt} \right) = \sum W_{ij} V_j - \left(\frac{u_i}{R_i} \right) + i_i$$

$$u_i = g_i^{-1}(v) \tag{3.3}$$

The same set of equations represents the resistively connected network of electronic amplifiers shown in Fig.3.1. The synapse (or weight) between two neurons is now defined by a conductance W_{ij} , which connects one of the two outputs of amplifier j to the input of amplifier i . The connection is made with a resistor of value $R_{ij} = |W_{ij}|^{-1}$, V_i is the output of the inverted amplifier. Fig.1 also includes an input resistance, r_i , for amplifier i . However, this value can be eliminated from the equation because R_i is a parallel combination of r_i and the R_{ij} :

$$\frac{1}{R_i} = \frac{1}{r_i} + \sum_j \frac{1}{R_{ij}}$$

For simplicity, each neuron/amplifier is assumed to be identical

$$g_i = g, R_i = R, \text{ and } C_i = C$$

Dividing Eqn.(2.3) by C and redefining $\frac{W_{ij}}{C}$ and $\frac{i_i}{C}$ to be W_{ij} and i_i , respectively, we

arrive at the equations of motion:

$$\left(\frac{du_i}{dt} \right) = \sum W_{ij} V_j - \left(\frac{u_i}{\tau} \right) + i_i \tag{3.4}$$

$$u_i = g_i^{-1}(v) \text{ and } \tau = RC$$

τ is the value of the time constant of the amplifiers, and without loss of generality can be assigned a value of unity, provided the time step of the discrete time simulation of Eqn. (3.4) is considerably smaller than unity. For the continuous Hopfield network, a Liapunov function can be constructed for the system, which guarantees convergence to stable states. Consider the energy function,

$$E_c = -\frac{1}{2} \sum_i \sum_j W_{ij} V_i V_j - \sum_i i_i V_i + \int_0^{v_i} g_i^{-1}(v) dv \quad (3.5)$$

Provided the matrix of weights W is symmetric

Time derivative of E is

$$\begin{aligned} \frac{dE_i}{dt} &= -\sum_i \frac{dV_i}{dt} \left(\sum_j W_{ij} V_j - \frac{u_i}{\tau} + i_i \right) \\ &= -\sum_i \left(\frac{dV_i}{dt} \right) \left(\frac{du_i}{dt} \right) \\ &= -\sum_i g_i^{-1}(V_i) \left(\frac{dV_i}{dt} \right)^2 \end{aligned} \quad (3.6)$$

Because $g_i^{-1}(V_i)$ is a monotonically increasing function, then

$$\frac{dE_i}{dt} \leq 0,$$

$$\text{also } \frac{dE_i}{dt} = 0 \text{ if } \frac{dV_i}{dt} = 0 \quad \forall i$$

The continuous Hopfield network therefore relates directly to the discrete version in the high-gain limit of the activation function (Eqn. 3.1). In this high-gain limit, $g(u_i)$ approximates the behavior of the discrete threshold function and the local minima of E_c coincide with the local minima of E_d and all lie at the vertices of the unit hypercube. Consequently, in the high-gain limit, provided the weight matrix is symmetric and that the inverse function of $g(u_i)$ (the first derivative of the activation function) exists, the continuous Hopfield network converges to a 0-1 vertex, which minimizes E_d given by Eqn. (3.6).

3.3 THE HOPFIELD NEURAL NETWORK APPROACH FOR SOLVING OPTIMIZATION PROBLEMS

Hopfield and Tank [21] realized that networks of neurons with this basic organization could be used to compute solutions to specific optimization problems by selecting weights and external inputs that properly represent the function to be minimized and the desired states of the problem. The analog nature of the neurons and the parallel processing of the updating procedure could be combined to create a rapid and powerful solution technique. Using the method proposed by Hopfield and Tank, the network energy function is made equivalent to the objective function of the optimization problem that needs to be minimized, while the constraints of the problem are included in the energy function as penalty terms. Consider the optimization problem:

$$\begin{aligned}
 & \text{(P1) minimize } f(\mathbf{V}) \\
 & \text{Subject to } [\mathbf{A}]_1 \mathbf{V} = b_1 \\
 & \quad \quad \quad [\mathbf{A}]_2 \mathbf{V} = b_2 \\
 & \quad \quad \quad \dots \dots \dots \\
 & \quad \quad \quad [\mathbf{A}]_r \mathbf{V} = b_r
 \end{aligned}$$

where $[\mathbf{A}]_i$ (the i^{th} row of the constraint matrix \mathbf{A}) and \mathbf{V} are n -dimensional vectors, and r is the number of constraints. Then the H-T energy function is

$$E(\mathbf{V}) = \alpha f(\mathbf{V}) + \beta_1 ([\mathbf{A}]_1 \mathbf{V} - b_1)^2 + \beta_2 ([\mathbf{A}]_2 \mathbf{V} - b_2)^2 + \dots + \beta_m ([\mathbf{A}]_m \mathbf{V} - b_m)^2$$

$\alpha, \beta_1, \beta_2, \beta_3$ are penalty parameters that are chosen to reflect the relative importance of each term in the energy function. The non-linearity of the terms makes determination of the optimal penalty parameters unlikely. Clearly, a constrained minimum of P1 will also optimize the energy function, because the objective function, $f(\mathbf{V})$, will be minimized, and constraint satisfaction implies that the penalty terms will be zero. Once a suitable energy function has been chosen, the network parameters (weights and inputs) can be inferred by comparison with the standard energy function given by Eqn. (3.6). The weights of the continuous Hopfield network, W_{ij} , are the coefficients of the quadratic terms $V_i V_j$, and the external inputs, i_i , are the coefficients of the linear terms V_i in the chosen energy function. The network can then be initialized by setting the activity level V_i of each neuron to a small

random perturbation around 0.5. This places the initial state of the system at approximately the center of the n-dimensional hypercube, and ensures that the initial state is unbiased. From its initialized state, asynchronous updating of the network will then allow a minimum energy state to be attained, because the energy level never increases during state transitions. 0–1 solutions of combinatorial problems can be encouraged if desired by setting the parameter T of the activation function to a small enough value that the function approximates the discrete threshold (step) function given by Eqn. (3.5).

Although Hopfield networks do provide a useful tool in solving optimization problems, they are prone to get stuck in local minima as they basically employ a gradient descent process, this problem can be overcome by using the concept of simulated annealing in the network.

Hardware implementation of a neural network is ideal for industrial applications, where the same problem will need to be solved many times as the environment changes, Fortunately, recent work in the area of Field Programmable Gate Arrays (FPGA's) has enabled the speed advantages of hardware implementation to be simulated on a digital computer using reconfigurable hardware with desktop programmability. Such a simulation can easily achieve speeds of several million interconnections per second, making the advantages associated with hardware implementation of neural networks more readily attainable. Certainly, satisfactory hardware implementation is still the topic of much research and many design challenges lie ahead in this field. The Hopfield neural network techniques, for optimization problems, can effectively compete with traditional heuristic and exact approaches when simulated on a digital computer for solution quality.

CHAPTER-4

A MODIFIED HOPFIELD NEURAL NETWORK FOR DETERMINATION OF TOPOLOGICAL OBSERVABILITY IN POWER SYSTEM

A new method for determining the network observability of the power networks using Hopfield neural network algorithm; embedded with simulated annealing is presented in this chapter. An extended method for handling the inequality constraints is proposed by assigning a dedicated neural network. The, network observability problem related to the power network configuration or network topology, called as the topological observability, is considered. This neural network based method has been applied on a sample power network and results are presented.

4.1 INTRODUCTION

Topological observability analysis is necessary to examine whether the relationship between measurement allocation and power system configuration is appropriate. Unless, at least one measurement is assigned to buses in a one-to-one manner, it is impossible to evaluate bus voltages with a state estimation technique. Every time when a system configuration is changed for some reasons, a topological observability test should be executed prior to performing the state estimation to check one-to-one correspondence between measurements and buses. If this is not the case, observability analysis methods can provide the minimum set of additional measurements needed to restore observability. Hence an efficient implementation of the topological observability test is a crucial step in achieving a satisfactory performance for the entire state estimation process and whole real-time monitoring and control of power systems.

Observability analysis has so far been accomplished with the help of either topological or numerical approaches. The topological approach makes use of the graph theory and determines network observability strictly based on the type and location of measurements. It does not use any floating point arithmetic and is implemented independent of the state estimation itself. Numerical approaches are based on the

measurement Jacobian and the associated gain matrix. It uses an iterative scheme to determine all the observable islands if the system is found to be unobservable.

Topological observability problem can also be formulated as a combinatorial optimization problem. Conventional methods based on graph theory are available but the computational time increases drastically as the number of buses increases [43-45]. In 1975, Clements and Wollenberg developed a method for topological observability [43]. Since then, a lot of work has been done in this area. Quintana, et al. developed a graph-theory based method [45]. Monticello and Wu presented an approach with the Gauss Elimination [46]. After that, Bargiela, et al., contributed to developing a technique for solving topological observability using the maximum flow [47]. Recently, Mori and Tsuzuki proposed artificial neural networks-based approaches using recurrent networks [48, 49].

Hopfield network consists of a large number of symmetrically connected neurons, for which an energy function can be defined as explained in chapter 3. Of course, an energy function that can be defined is specific to a particular connection. Therefore, if an optimization problem can be mapped onto the Hopfield type neural network; there is a great possibility that the problem will be solved. Here the term “mapping” means the variables of the problem can be correlated to neuron’s output and its objective function can be equated to the energy function of the constructed neural network.

In solving Topological observability problem with a Hopfield network, equality constraints can be taken into consideration by adding corresponding terms to the energy function. However, Inequality constraints are indispensable in some cases and hence they should be handled in the network. Mori and Tsuzuki [50] introduced slack variables to handle the inequality constraints by converting them into equality constraint. On the other hand, the slack variables make the problem more complicated by increasing the number of neurons and thereby increasing number of local minima.

In this chapter, Hopfield neural network is examined for solving power system topological observability problem. An extended method for handling the inequality constraints is proposed by assigning a dedicated neural network, whereas when the slack variable formulation is used, extra nodes are required to represent those variables. The

proposed method is computationally simpler and requires reduced number of neurons in the Hopfield network. The advantage of reducing number of neurons is that while implementing the neural network in hardware for a greater computational speed, the hardware circuit requirement will be lessened.

4.. NETWORK OBSERVABILITY:

Observability analysis may be divided into two categories: numerical and topological observability.

A: Topological observability

Topological observability algorithms, which use information about the network topology and measurement, were developed in order to avoid the rather difficult task of numerical computation of the rank of the measurement Jacobian matrix. Such algorithms have been widely used in the state estimator observability programs. In the case of networks containing only line flow measurements in which real and reactive measurements occur in pairs, the topological condition for observability is that there exists at least one bus voltage magnitude measurement and that a spanning tree of the entire network can be built using only measured lines. Finding such a tree can be done using one of the well-known tree search methods such as breadth-first or depth-first search. For an N-bus network with only bus injection measurements, the determination of observability is even simpler; there must be at least one bus voltage measurement and at least N-1 bus injection measurements.

The measurement model for state estimator is

$$z = h(x) + e$$

Where

z represents all measurements, including power injection, power flow and bus voltage magnitude measurements,

e is the measurement noise vector,

x is the state vector composed of the phase angles and magnitudes of the voltages at network buses,

$h(.)$ stands for the nonlinear measurement functions relating measurements with state variables.

The above measurement model is approximated by a linear plus a constant term model

$$\Delta z = [H] \Delta x + c + e$$

where

$$H = \begin{bmatrix} H_{\theta} & 0 \\ 0 & H_V \end{bmatrix}$$

"An N-bus power network is observable with respect to a given measurement set M if and only if the rank of the gain matrix H is equal to 2N-1".

By applying the P- θ /Q-V decoupling principal, the observability problem is decoupled in P- θ observability and Q-V observability. A network is said to be P- θ observable if the rank of matrix H_{θ} is equal to N-1, and it is said to be Q-V observable if rank of the H_V matrix is equal to N.

B: Numerical observability:

Generally a numerical observability analysis is based on triangular decomposition of the information matrix G(x). If G(x) can be successfully factored without encountering any zeros in the diagonal, the system is observable. On the other hand, if the network is not observable then one or more zeros will appear on the diagonal of the triangular factor. When this happens, a pseudo measurement of bus voltage angle in the P- θ observability test at the bus corresponding to the zero diagonal elements is added. These buses then are automatically identified as buses requiring injection measurements for observability.

The numerical observability algorithms are new and are starting to be implemented at some control centers. They have the virtue of being conceptually simple and of employing numerical routines that are already needed for computation of state estimation. There is however, the potential of difficulty in determining whether a small number appearing on the diagonal is either a nonzero value or is actually zero, because of numerical round off. Topological algorithms on the other hand require additional non numerical routines that may be rather complex but generally run faster than numerical tests.

4.2.1: Formulation of topological observability problem

In this section topological observability problem is reformulated as an integer programming problems. The condition that a network is topological observable implies that a bus has at least one measurement and a measurement is assigned to only one bus

[49]. Let (w_{ij}) be a graph representing relationship between “ m ” measurements and “ n ” buses. For example, consider a five bus system in Fig.4.1, with five measurements $z_1 - z_5$. graph w can be written as

$$w = \begin{matrix} & n_1 & n_2 & n_3 & n_4 & n_5 \\ \begin{matrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{matrix} & \begin{bmatrix} w_{11} & 0 & w_{13} & 0 & 0 \\ w_{21} & w_{22} & 0 & 0 & 0 \\ w_{31} & w_{32} & w_{33} & w_{34} & 0 \\ w_{41} & w_{42} & w_{43} & 0 & 0 \\ 0 & w_{52} & 0 & 0 & w_{55} \end{bmatrix} \end{matrix}$$

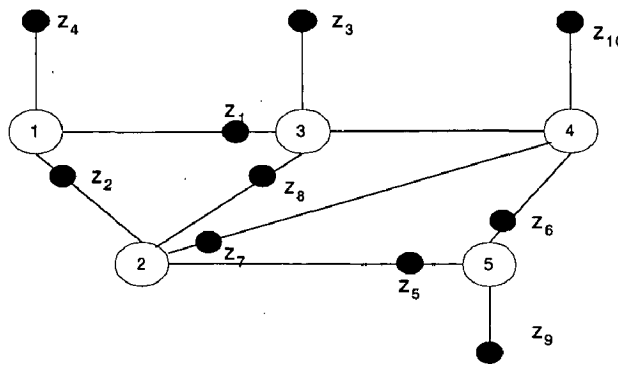


Figure 4.1: A five bus system

Each element w_{ij} of matrix w is defined as follows:

$$\begin{aligned} w_{ij} &= 1 && \text{if measurement } z_i \text{ is assigned to bus } j. \\ &= 0 && \text{if measurement } z_i \text{ is not assigned to bus } j. \end{aligned}$$

Topologically observability is examined by examining whether Eqn. (4.1) and (4.2) holds.

$$\sum_{j=1}^n w_{ij} = 1 \quad \text{for } i=1,2,\dots \dots m \text{ (row constraint)} \quad (4.1)$$

$$\sum_{i=1}^m w_{ij} \geq 1 \quad \text{for } j=1,2,\dots \dots n \text{ (column constraint)} \quad (4.2)$$

Eqn. (4.1) implies that measurement z_i is assigned to only one bus. On the other hand, Eqn. (4.2) indicates that bus n_j has at least one measurement

4.3. NEURAL NETWORKS METHODOLOGY

Tsuzuki and Mori proposed a method using slack variables to handle the inequality constraints. On the contrary, the slack variables result in increasing the number of additional neurons [50], which implies that the original problem become more complicated due to increase of local minima points.

Proposed method:

Let us tentatively ignore inequality constraints. A Hopfield neural network can then be used for solving topological observability problem given by Eqns. (4.1) and (4.2). Variable w_{ij} can be related to neuron output V . Let there are n buses and m measurements in the network, and graph of the network is defined in matrix form where buses and meters are indexed by “ x ” and “ i ” respectively.

$$V_{xi} = \begin{cases} 0 & \text{if the bus } x \text{ is not assigned measurement } i \\ 1 & \text{if bus } x \text{ is assigned measurement } i \end{cases}$$

The objective function is defined as

$$E = \frac{A}{2} \sum_{x=1}^n \sum_{i=1}^m \sum_{y=1}^m V_{xi} V_{yi} + \frac{B}{2} \left(\sum_{x=1}^n \sum_{i=1}^m V_{xi} - n \right)^2 \quad (4.3)$$

The first part is included to correspond to the constraint that exact one neuron in each column can output a “1”. Thus, we get 0 for this term with a valid solution. The second term of right-hand side has a minimum value of 0, which is attained if and only if exactly m of the $m \times n$ output states (V_{ij}) have value 1 and the rest 0. A , B are constants reflecting relative importance of terms in the energy function.

Obtaining weight matrix:

Method 1: Rewrite E in the form of

$$E = -\frac{1}{2} \sum_{xi=1}^n \sum_{yj=1}^m W_{xi,yj} V_{xi} V_{yj} - \sum_{xi=1}^n \theta_{xi} V_{xi} \quad (4.4)$$

Where $W_{xi,yj}$ and θ_{xi} are in terms of parameters A and B There is no systematic procedure for such conversion.

Method 2: Determine local function of motion $\frac{du}{dt}$ from E such that it always decrease E , also by

continuous Hopfield method

$$\frac{du_{xi}}{dt} = \sum_{i=1}^n \sum_{j=1}^m \delta_{xi-yj} W_{xi,yj} V_{yj} + \theta_{xi}$$

Now find $W_{xi,yj}$, θ_{xi} and $\frac{du}{dt}$

Steps are as follows

Determine $\frac{du_{xi}}{dt}$ from E so that $\frac{dE}{dt} < 0$

$$\frac{dE}{dt} = \sum_{xi} \left(\frac{dE}{dV_{xi}} \right) \left(\frac{dV_{xi}}{du_{xi}} \right) \left(\frac{du_{xi}}{dt} \right)$$

If $\frac{du_{xi}}{dt} = -\frac{dE}{dV_{xi}}$, then

$$\frac{dE}{dt} = -\sum \left(\frac{dE}{dV_{xi}} \right)^2 \left(\frac{dV_{xi}}{du_{xi}} \right) \leq 0$$

$$-\frac{dE}{dV_{xi}} = -A \sum_{j=1}^n \sum_{i=1}^m V_{yj} - B \left(\sum_{j=1}^n \sum_{i=1}^m V_{xi} - m \right)$$

Since $\frac{du_{xi}}{dt} = \sum_{i=1}^n \sum_{j=1}^m \delta_{xi-yj} W_{xi,yj} V_{yj} + \theta_{xi}$

$$W_{xij} = -A\delta_{ij}(1 - \delta_{xy}) - B \quad (4.5)$$

Where $\delta_{xy} = 1$ if $x = y$

=0 otherwise

Each node also has a positive bias $\theta_{xi} = Bm$

The above algorithm gives minimum of energy function E which satisfies the equality constraints only. The architecture of the network is shown in Fig. 4.2.

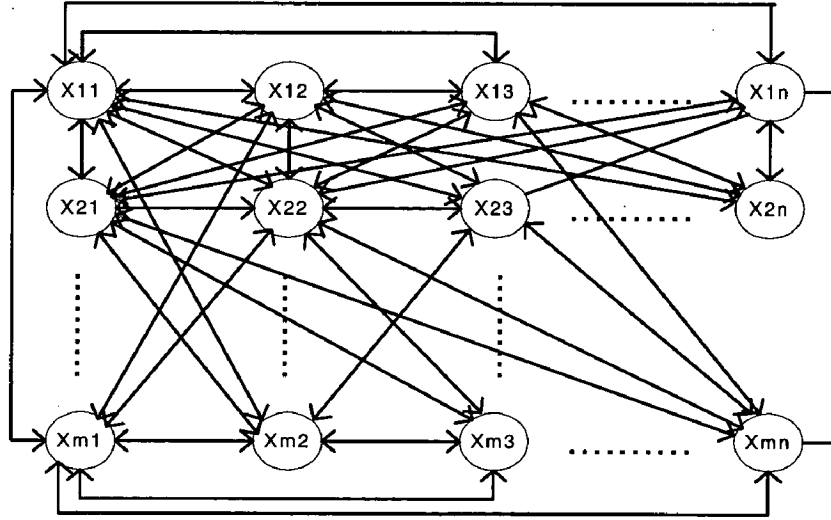


Figure 4.2: Architecture of Hopfield Network

4.3.1 Neural network for handling inequality constraints in the topological observability problem:

The standard Hopfield network has a serious shortcoming that it cannot solve a combinatorial optimization problem with inequality constraints. Hence the neural network, which has been proposed by tank and Hopfield to solve a simple linear programming problem with two variables and four constraints [21], is used with slight modification that is, the sigmoid characteristic is used instead of the linear relationship in the original neural network for solving the inequality constraints in the topological observability problem.

A linear programming (LP) problem including inequality constraints may be defined as follows:

$$\text{Minimize: } P = A \cdot V \tag{4.6}$$

$$\text{Subject to: } D_j \cdot V \geq B_j \quad (j=1 \dots M) \tag{4.7}$$

$$D_j = [D_{j1}, D_{j2} \dots D_{jn}]^T \tag{4.8}$$

Where, the D_j , for each j , contains the N variable coefficients in a constraint equation and the B_j are the bounds. The neural network to solve LP problem is shown in Fig.4.3. The N outputs (V_j) of the left-hand set of amplifiers will represent the values of the variables in the linear programming problem. The components of 'A' are proportional

to input currents fed into these amplifiers. Outputs (ψ_j) of the right-hand set of amplifiers represent constraint satisfaction. As indicated in the figure, the output (ψ_j) of the j^{th} amplifier on the right-hand side injects current into the input lines of the V_i variable amplifiers by an amount proportional to $-D_{ji}$, the negative of the constraint coefficient for the i^{th} variable in the j^{th} constraint equation. Each of the constraint ψ_j amplifiers are fed with a constant current proportional to the j^{th} bound constant (B_j) and receives input from the i^{th} variable amplifier by an amount proportional to D_{ji} . Each of the V_i amplifiers in the linear programming network has an input capacitor C_i and an input resistor ρ_i in parallel, which connect the input line to ground. The input-output relations of the V_i amplifiers are linear and characterized by a linear function g in the relation $V_i = g(u_i)$.

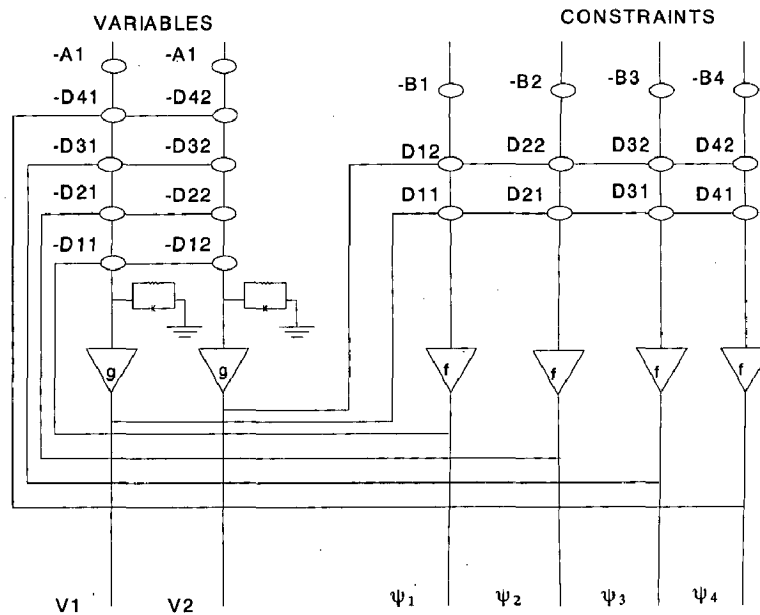


Figure 4.3: Organization of a network to solve a 2-variable 4-constraint linear programming problem [21]

The ψ_j amplifiers have the nonlinear input-output relation characterized by the function

$$\begin{aligned}
 \psi_j &= f(u_j) \\
 u_j &= D_j \cdot V - B_j \\
 f(u_j) &= 0 \quad (u_j \geq 0) \\
 f(u_j) &= Ku_j \quad (u_j < 0)
 \end{aligned} \tag{4.9}$$

Where K is a positive constant and function f give large output to variable

amplifier when the corresponding constraint is not satisfied. The input output characteristic of variable amplifiers should be changed so that they take on an integer value of 0 or 1;

$$V_i = g(u_i) \tag{4.10}$$

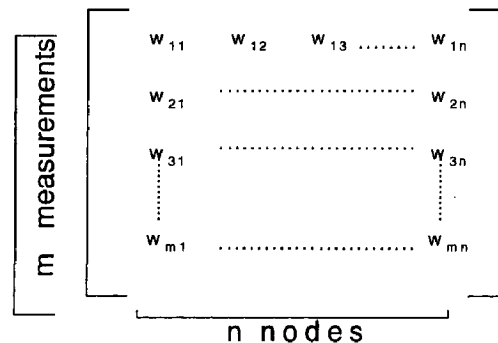
$$g(u_i) = \frac{1}{2}(1 + \tanh(u_i)) \tag{4.11}$$

The specific features of this neural network may be summarized as follows:

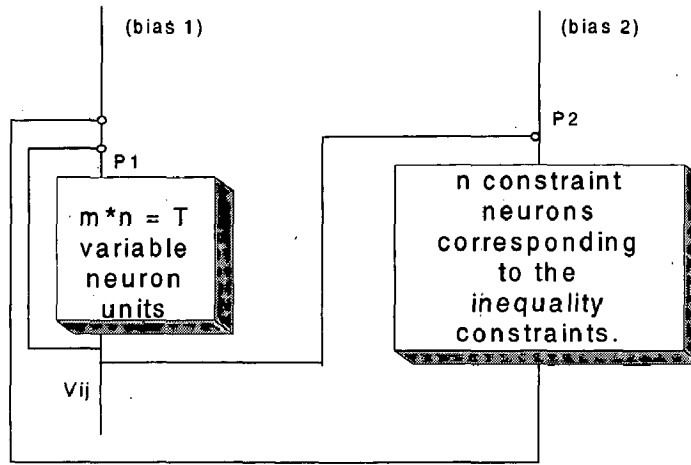
- (a) Inequality constraints need not be included in its energy function.
- (b) It always converges to a solution, which satisfies inequality constraints.

Connections between the constraint neurons and variable neurons must be determined so that the output of the former make the outputs of the latter satisfy constraints if they are violated. In order to realize this, all the inequality constraints are first changed into the form of $h(V) \geq 0$ so that they are constrained from their lower limits. Then the coefficients of the constraints are directly used as connections. It should be noted that all the constraints are expressed by linear function of V .

Matrix relating “m” and “n” looks as shown below



The proposed network contains $m \times n$ variable neurons and n constraint neurons as shown in Fig 4.4.



m: Number of measurements n: Number of nodes

Figure 4.4: Neural network for solving topological observability

P_2 : connection from variable to constraint neuron which is matrix of size $n \times m$ and having elements as 1 corresponding to connections and 0 otherwise.

$$\theta_2 = \text{bias}_2 = -1$$

4.4 ALGORITHM:

- 1) Initialize the network.
- 2) Assign initial input states to the network, and compute the activation and output of the network. After that, the network is evolved, and it proceeds to cycle through a succession of states, until it converges on a stable solution;
- 3) The activation of neuron in the i^{th} row and j^{th} column is defined as by U_{ij} , and the output is denoted as V_{ij} . A time constant τ and a gain λ are used as well. Also, Δt denotes the increment in time, from one cycle to the next. Change in $U_{ij}(n)$ which decreases the energy, is given as in Eqn.(4.12);

$$\Delta U_{ij}(n) = \Delta t \left[-U_{ij}/\tau - A \sum_{k=i} V_{kj} - B \left(\sum_i \sum_k V_{ik} - N \right) \right] \quad (4.12)$$

$$U_{ij}(n+1) = U_{ij}(n) + \Delta U_{ij}(n);$$

$$V_{ij}(n+1) = F(U_{ij}(n+1));$$

It is to be noted here that the function $F(\cdot)$ is a continuous function instead of a hard switch, which makes the network a continuous Hopfield network. The reason to employ such a continuous function is to prevent an extremely fast convergence to a local optimum, which is always caused by a hard switch function.

- 4) Enforce inequality constraint: Use the procedure described in section 4.3.1.
- 5) Simulated annealing Suppose we have got a solution matrix V , U and the corresponding energy E . We statistically train U , V by simulating the annealing in metallurgy, in order to get better solution. Firstly, set the initial temperature T^0 and parameter K ; Secondly, for each iteration, gradually cool down (decrease the temperature T) using function $T = \frac{T_0}{\log(1+k)}$, where k denotes iteration and $k < N^2$. Randomly change U by ΔU , and see if energy E decreases: if so, accept the change and update V , and E ; if not, then compute probability of ΔE by equation $p(\Delta E) = \exp(-\Delta E / KT)$ and see if it is greater than a random value $r = K * \text{rand}$, if so, still accept the changes; if not, reject it. Operate the training for N^2 iterations.
- 6) Iterate the updating to the activation and output until the network converges to a stable solution. It happens when the change of the energy is lower than a pre-set small positive number.

The topological observability problem is solved following the procedure shown in Fig (4.5).

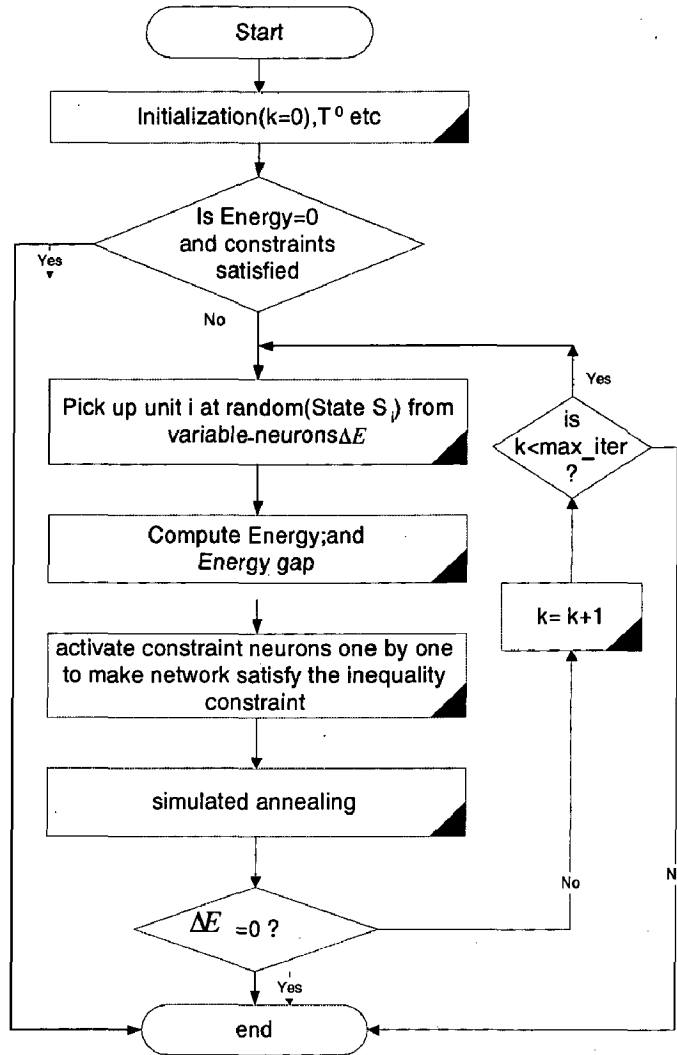


Figure 4.5: Flow chart for solving topological observability problem with Hopfield network.

4.5 RESULTS AND DISCUSSIONS

The proposed method is tested in a five bus system of Fig.4.1, where measurements z_1 - z_{10} are allocated. The system is studied for cases when number of measurements (m) lies between 5 and 10. Fig.4.7 shows the bipartite graph in the five bus system. The following convergence criterion is adopted.

$\Delta E=0$ or max iteration count=100.

The cooling schedule in simulated annealing is $T=T_0/\log(1+k)$, where k is iteration count. T_0 is taken as 3.0. Probability of obtaining global minima is significantly improved as compared to basic Hopfield model. The following table shows the

measurement set for cases (a-f)

Table 4.1: Measurement for each case

Cases	Measurements	No. of measurements	Mean of iteration count
a	z1-z5	5	13
b	z1-z6	6	15
c	z1-z7	7	14
d	z1-z8	8	16
e	z1-z9	9	19
f	z1-z10	10	22

Output for 5 buses and 8 measurements is shown as below:

$$V = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Convergence of energy function with respect to iteration count is shown in Fig.4.7 for case (d)

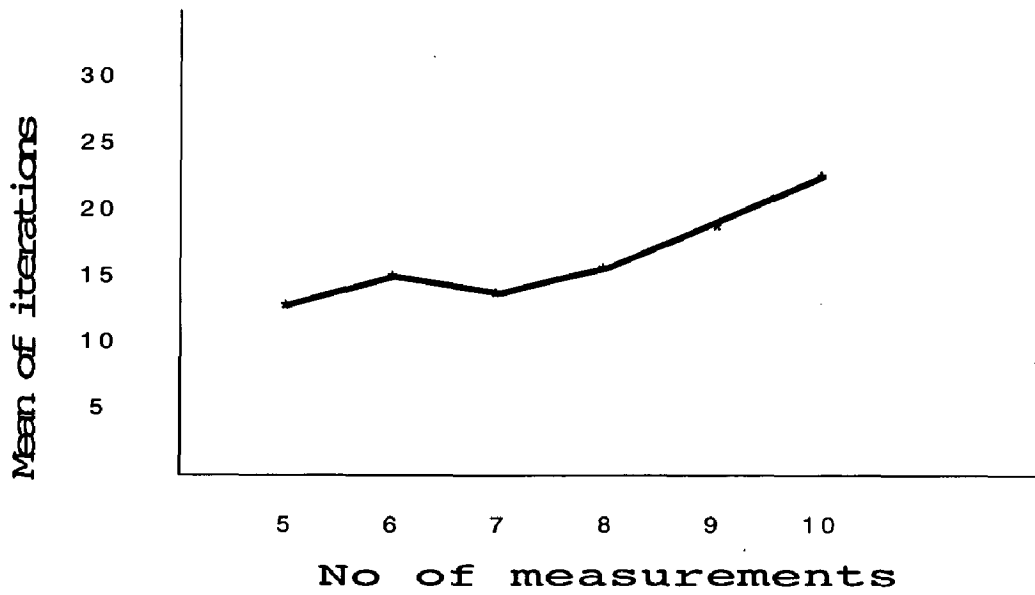


Figure 4.6: mean of iterations Vs no of measurements

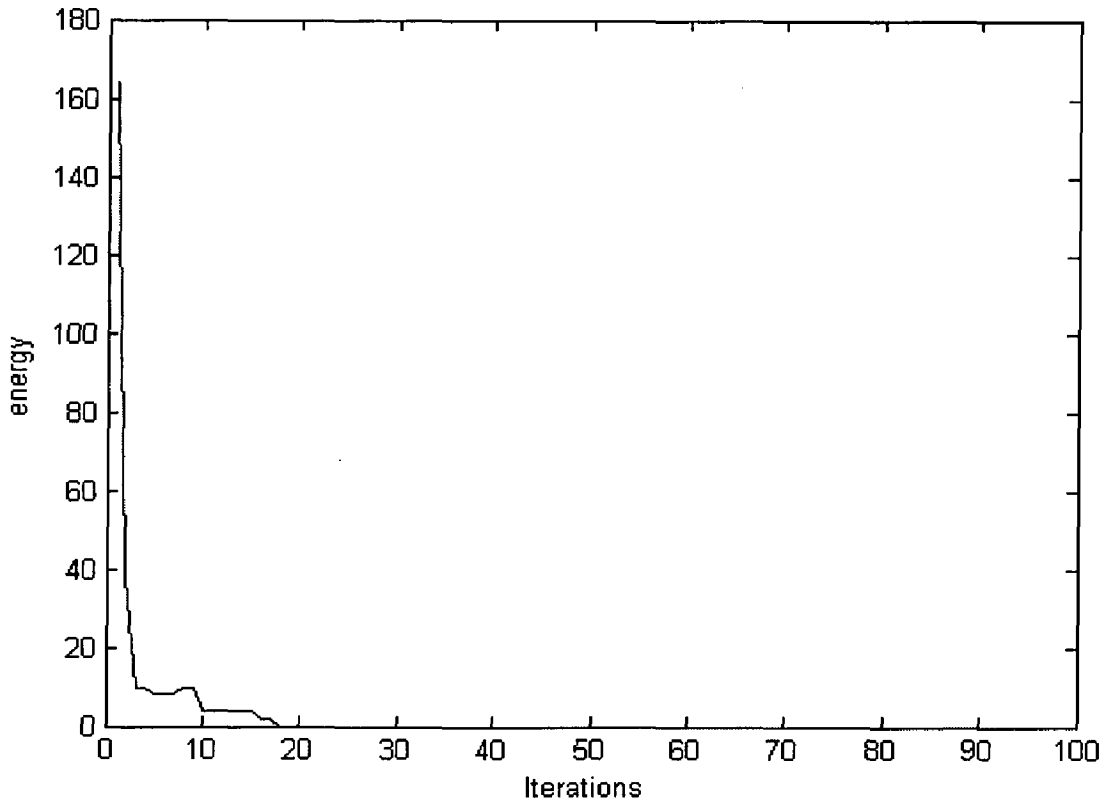


Figure 4.7: Convergence of energy function

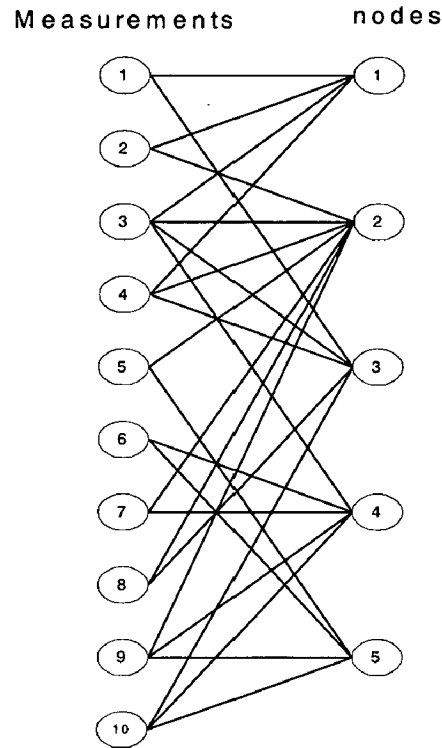


Figure 4.8: Bipartite graph of 5 bus network

The power system configurations often vary along the daily load scheduling. As a result, there is a possibility that measurements are lost due to some faults. Hence, it is necessary to consider the relationship between meter placement and system configurations each time they are changed, or a part of measurements is lost or modified. That is because such changes may make state estimation impossible for lack of some key measurements. In this chapter a new method based on Hopfield neural network has been developed for judging network topological observability and identifying where the meters should be placed to recover the network topological observability.

The proposed method is effective in reducing the number of neurons together with an assurance for finding global solution with the incorporation of simulated annealing. Recently much work has been done towards hardware implementation of Hopfield neural network on field programmable gate array (FPGA) [51, 52]. Proposed method is more suitable for FPGA hardware implementation because the hardware circuit requirement is reduced.

CHAPTER 5

A MODIFIED HOPFIELD NEURAL NETWORK METHOD FOR EQUALITY CONSTRAINED STATE ESTIMATION

In this chapter, state estimation problem is formulated as a general nonlinear programming problem with equality constraints and boundary limits on the state variables and a method based on neural network technique is presented. The method is tested on a six-bus system and the IEEE 14-bus system, which shows that when it is implemented on hardware, it can be used in real time state estimation of power system.

5.1 INTRODUCTION

State estimation processes a set of measurements to obtain the best estimate of the current state of the power system. The set of measurements includes telemetered measurements and pseudo-measurements. Telemetered measurements are the online telemetered data of bus voltages, line flows, injections, etc. Pseudo-measurements are manufactured data such as guessed MW generation or substation load demand based on historical data, in most cases. Telemetered measurements are subject to noise or error in metering, communication system, etc. The errors of some of the pseudo-measurements, especially the guessed ones, may be large. However, there is a special type of pseudo-measurements, known as the zero injections, for which the information contains no error. Zero injection occurs at a node, for example, representing a switching station where the power injection is equal to zero. Zero injection is an inherent property of such a node and no meter needs to be installed but the information is always available. A state estimation algorithm must compute estimates, which satisfy exactly such constraints, independent of the quality of online measurements. The enforcing of constraints is in particular useful in networks, consisting of large unobservable parts of network or having very low measurement redundancy.

In its conventional form, the weighted least square method does not enforce the equality and limit constraints explicitly. However, the constraints contain reliable information about physical restrictions and equipment limits and can be used to increase

the quality of state estimation result. The zero injections can be represented by a set of equalities. Various methods have been proposed to process constraints, literature review section lists some of the proposed methods for solving equality constrained state estimation problem.

Various algorithms of state estimation using the conventional computer are reaching a limit as far as the solution techniques are concerned, and as long as these computer based algorithms are used, faster methods can not be expected. However for security monitoring and control in power system, improvement in calculation time is always desired in order to obtain necessary information more quickly and accurately.

In recent years, it has been found that Artificial Neural Networks (ANNs) are well suited as computational tools for solving certain classes of complex problems, although software implementations of the algorithm on general-purpose computers can be too slow for time-critical applications, but the small number of computational 'primitives', suggests advantages of hosting ANNs on dedicated Neural Network Hardware (NNH) to maximize performance at a given cost target. ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

In this chapter a new method for enforcing equality and limit constraints in state estimation algorithm using a modified Hopfield neural network is presented. The main advantages of using the modified Hopfield neural network proposed in this work are i-) the internal parameters of the network are explicitly obtained by the valid-subspace technique of solutions, ii-) lack of need for adjustment of penalty factors for initialization of constraints, and iii-) for real time application, the modified Hopfield network offers simplicity of implementation in analog hardware or a neural network processor, and (iv) training and testing of the neural network under human supervision is not required.

5.2: STATE ESTIMATION WITH CONSTRAINTS

State vector of an electric network consists of the complex voltages at the buses. Unmeasured tap positions of transformers may also be included into the state vector. A measurement vector consists of power flows, power injections, voltage and current magnitudes and tap positions of transformers. For a N bus system, the state vector

$x = [\delta, V]^T$, of dimension $n=2N-1$, consists of the $N-1$ bus voltage angles δ_i with respect to a reference bus and the N bus voltage magnitudes V_i for $i=1,2,3,\dots,N$.

The static state estimator measurement model is given as:

$$z = h(x) + \varepsilon \quad (5.1)$$

where z is the measurement vector, $h(\cdot)$ is a vector of nonlinear functions, relating the measurement and state vectors, and ε is the vector of measurement errors.

The error-free data are modeled as equality constraints

$$g(x)=0 \quad (5.2)$$

Limits on some network variables are modeled as inequality constraints which can be expressed in a compact form by p -dimensional functional inequalities

$$f(x) \leq 0 \quad (5.3)$$

General nonlinear programming algorithms for the solution of a constrained minimization problem [53] are not efficient enough for the on-line application. Hence a neural network approach is used for solving this nonlinear programming problem.

Objective function

The objective is to minimize the weighted squared mismatch between measured and calculated quantities. Considering system to be observable and with $m > n$, where m is the total number of measurements and n is the number of state variables, the mathematical problem is given as follows:

$$\min \frac{1}{2} [z - h(x)]^T R^{-1} [z - h(x)] \quad (5.4)$$

Subject to the equality and inequality constraints as defined below.

The diagonal matrix R^{-1} represents the weights of the individual measurements in the objective function.

Equality constraints

Power flow equations, corresponding to both real and reactive power balance are the equality constraints for all the buses characterized as zero injections, which can be expressed as follows:

$$P_i = \sum_{m=1}^{N_b} V_i V_m (g_{im} \cos \delta_{im} + b_{im} \sin \delta_{im}) = 0 \quad (5.5)$$

$$Q_i = \sum_{m=1}^{N_b} V_i V_m (g_{im} \sin \delta_{im} - b_{im} \cos \delta_{im}) = 0 \quad (5.6)$$

for $i \in$ (set of zero injection buses)

Where

P_i = Real power injection at bus- i

Q_i = Reactive power injection at bus- i

V_i = Voltage magnitude at bus- i

δ_i = Load angle at bus- i

$Y_{ij} = g_{ij} + jb_{ij} = i$ - j th element of Y-bus Matrix.

N_b, N_l, N_g =number of total buses, load buses and generator buses in the system respectively.

Inequality Constraints:

(i) Voltage Limit: This includes upper (V_i^{max}) and lower (V_i^{min}) limits on the bus voltage magnitude.

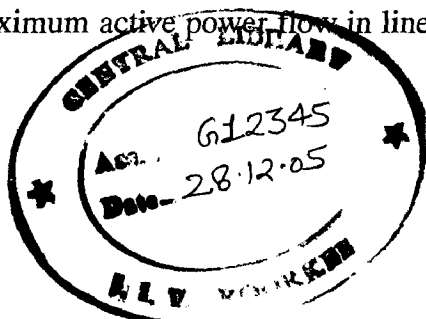
$$V_i^{min} \leq V_i \leq V_i^{max} \quad i = 1, 2, \dots, N_b \quad (5.7)$$

(ii) Phase Angle Limits: The phase angle at each bus should be between lower (δ_i^{min}) and upper (δ_i^{max}) limits.

$$\delta_i^{min} \leq \delta_i \leq \delta_i^{max} \quad i = 1, 2, \dots, N_b \quad (5.8)$$

These limits may vary depending upon the problem under consideration. Imposing phase angle limits at load buses is another way of limiting the power flow in the transmission lines and for generator buses this limiting is done for stability reasons. Along with the above two constraints the following constraints can also be imposed.

(a) Line Flow Limit, representing the maximum power flow in a transmission line and is usually based on thermal and dynamic stability considerations. Let P_{li}^{max} be the maximum active power flow in line- i respectively. The line flow limit can be written as



$$P_{ii}^{\max} \geq P_{ii} \quad i = 1, 2, \dots, N_l \quad (5.9)$$

(b) Reactive Power Generator Limit: Let Q_{gi}^{\min} and Q_{gi}^{\max} are the minimum and maximum reactive power generation limit of the reactive source generators (N_q) respectively.

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max} \quad i = 1, 2, \dots, N_g \quad (5.10)$$

5.3: THE MODIFIED HOPFIELD NEURAL NETWORK

Artificial neural networks attempt to achieve good performance via dense interconnection of simple computational elements. Hopfield networks [21] are single-layer networks with feedback connections between nodes. In the standard case, the nodes are fully connected. The node equation for the continuous-time network with n-neurons is given by:

$$u_i(t) = -\eta_i u_i(t) + \sum_{j=1}^n T_{ij} v_j(t) + i_i^b \quad (5.11)$$

$$v_i(t) = g(u_i(t)) \quad (5.12)$$

Where $u_i(t)$ is the current state of the i^{th} neuron, $v_j(t)$ is the output of the j^{th} neuron., i_i^b is the offset bias of the i^{th} neuron., $\eta_i u_i(t)$ is the passive decay term, and T_{ij} is the weight connecting the j^{th} neuron to i^{th} neuron.

In Eqn. (5.12), $g(u_i(t))$ is a monotonically increasing threshold function that limits the output of each neuron to ensure that network output always lies in or within a hypercube. It is shown in [25] that the equilibrium points of the network correspond to values of $v(t)$ for which the energy function associated with the network is minimized:

$$E(t) = -\frac{1}{2} v(t)^T \cdot T \cdot v(t) - v(t)^T i^b \quad (5.13)$$

Mapping of constrained nonlinear optimization problems using a Hopfield network consists of determining the weight matrix T and the bias vector i^b to compute equilibrium points. Some mapping techniques codes the validity constraints as terms in the energy function which are minimized when the constraints ($E^{\text{cons}}_i = 0$) are satisfied :

$$E(t) = E^{op}(t) + b_1.E^{cons_1}(t) + b_2.E^{cons_2}(t) + \dots \quad (5.14)$$

Where $E^{op}(t)$ represents the objective function to be optimized and E^{cons} represents the constraints of the problem. The b_i parameters in Eqn. (5.14) are constant weightings given to various energy terms. The multiplicity of terms in the energy function tends to frustrate one another, and success of the network is highly sensitive to the relative values of b_i . It has been shown in [24] that the E^{op} and E^{cons} terms in Eqn. (5.14) can be separated into different subspaces so that they no longer frustrate one another. A modified energy function $E'(t)$ can be defined as follows:

$$E'(t) = E^{conf}(t) + E^{op}(t) \quad (5.15)$$

Where $E^{conf}(t)$ is a confinement term that groups all the constraints imposed by the problem, and $E^{op}(t)$ is an optimization term that conducts the network output to the equilibrium points. Thus, the minimization of $E'(t)$ of the modified Hopfield network is conducted in two stages:

1): minimization of the term $E^{conf}(t)$:

$$E^{conf}(t) = -\frac{1}{2}v(t)^T.T^{conf}.v(t) - v(t)^T.i^{conf} \quad (5.16)$$

Where: $v(t)$ is the network output, T^{conf} is weight matrix and i^{conf} is bias vector belonging to $E^{conf}(t)$.

2): minimization of the term $E^{op}(t)$:

$$E^{op}(t) = -\frac{1}{2}v(t)^T.T^{op}.v(t) - v(t)^T.i^{op} \quad (5.17)$$

Where: T^{op} is weight matrix and i^{op} is bias vector belonging to E^{op} . This minimization moves $v(t)$ towards an optimal solution (the equilibrium points).

Thus, the operation of the modified Hopfield network can be summarized as combination of three main steps, as shown in Fig 5.1:

Step (1): Minimization of E^{conf} Corresponding, to the projection of $v(t)$ in the valid subspace defined by [25,54]:

$$\mathbf{v}(t) = \mathbf{T}^{\text{conf}} \cdot \mathbf{v}(t) + \mathbf{i}^{\text{conf}} \quad (5.18)$$

Where: \mathbf{T}^{conf} is a projection matrix such that $\mathbf{T}^{\text{conf}} \cdot \mathbf{T}^{\text{conf}} = \mathbf{T}^{\text{conf}}$ and \mathbf{i}^{conf} is defined such that $\mathbf{T}^{\text{conf}} \cdot \mathbf{i}^{\text{conf}} = \mathbf{0}$. This operation corresponds to an indirect minimization of $E^{\text{conf}}(t)$.

Step (2): Application of a nonlinear 'symmetric ramp' activation function constraining $\mathbf{v}(t)$ in a hypercube

$$\begin{aligned} g_i(v_i) &= v^{\min} \quad \text{if } v^{\min} > v_i \\ &= v_i \quad \text{if } v^{\min} \leq v_i \leq v^{\max} \\ &= v^{\max} \quad \text{if } v_i > v^{\max} \end{aligned}$$

where $v_i \in [v^{\min}, v^{\max}]$

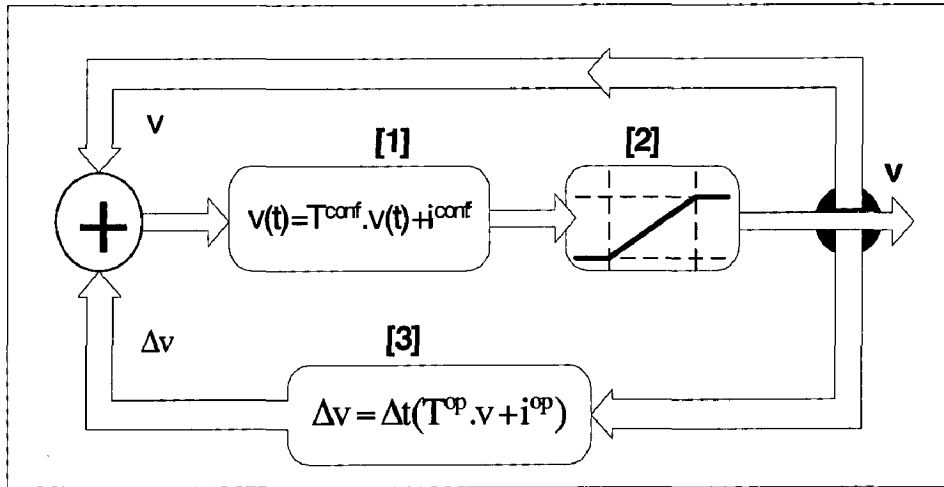


Figure-5.1: Modified Hopfield Neural Network

Step (3): Minimization of E^{op} , which involves updating of $\mathbf{v}(t)$ so as to direct it to an optimal solution (defined by \mathbf{T}^{op} and \mathbf{i}^{op}) corresponding to network equilibrium points, which are the solutions for the constrained optimization problems. Using the symmetric ramp activation function and $\eta = 0$, Eqn. (5.12) becomes.

$$\mathbf{v}(t) = \mathbf{g}(\mathbf{u}(t)) = \mathbf{u}(t)$$

Comparing Eqn. (5.11) and Eqn. (5.16),

$$\frac{d\mathbf{v}}{dt} = \dot{\mathbf{v}} = -\Delta t \cdot \nabla E^{\text{op}}(\mathbf{v}) = \Delta t (\mathbf{T}^{\text{op}} \cdot \mathbf{v} + \mathbf{i}^{\text{op}})$$

$$\Delta v = \Delta t \cdot \dot{v} \quad (5.19)$$

Therefore, minimization of E^{op} consists of updating $v(t)$ in the opposite direction to the gradient of E^{op} . Each iteration has two distinct stages. First, as described in Step (iii) v is updated using the gradient of the term E^{op} alone. Second, after each updating, v is directly projected in the valid subspace. In the next section, the parameters T^{conf} , i^{conf} , T^{op} and i^{op} are defined.

5.4: FORMULATION OF STATE ESTIMATION PROBLEM BY MODIFIED HOPFIELD NETWORK METHOD

Consider the following nonlinear optimization problem:

$$\text{Minimize } E^{op}(x) = f(x) = \frac{1}{2}[z - h(x)]^T R^{-1}[z - h(x)] \quad (5.20)$$

Where $x = [\delta, V]$, z = measurement vector and $h(x)$ represent nonlinear relationship between state vector x and z ,

$$\text{Subject to } E^{conf}(x): h_i(x) = 0, \quad (5.21)$$

i.e $P_i=0$ and $Q_i=0$

for $i \in$ (buses identified as zero injections)

$$V^{\min} \leq V \leq V^{\max}$$

$$\delta^{\min} \leq \delta \leq \delta^{\max} \quad (5.22)$$

Where V , V^{\min} , V^{\max} , δ , δ^{\max} , $\delta^{\min} \in \mathbb{R}^n$; and all first and second order partial derivatives of $f(x)$ and $h_i(x)$ exist and are continuous. The conditions in Eqns. (5.21) and (5.22) define a bounded convex polyhedron. The vector x must remain within this polyhedron if it is to represent a valid solution for the optimization problem (Eqn. 5.20). However if inequality constraints are also present, they must be transformed into equality constraints by introducing a slack variable s_w for each inequality constraints prior to calculating the parameters T^{conf} and i^{conf} . It is to be noted here that E^{op} does not depend on the slack variables s_w . A projection matrix to the system can be shown as [54].

$$\mathbf{T}^{\text{conf}} = [\mathbf{I} - \nabla \mathbf{h}(\mathbf{x})^T \cdot (\nabla \mathbf{h}(\mathbf{x}) \cdot \nabla \mathbf{h}(\mathbf{x})^T)^{-1} \cdot \nabla \mathbf{h}(\mathbf{x})] \quad (5.23)$$

where

$$\nabla \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \frac{\partial h_1(\mathbf{x})}{\partial x_1} & \frac{\partial h_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial h_1(\mathbf{x})}{\partial x_N} \\ \frac{\partial h_2(\mathbf{x})}{\partial x_1} & \frac{\partial h_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial h_2(\mathbf{x})}{\partial x_N} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial h_p(\mathbf{x})}{\partial x_1} & \frac{\partial h_p(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial h_p(\mathbf{x})}{\partial x_N} \end{bmatrix} \quad (5.24)$$

Inserting the value of \mathbf{T}^{conf} from Eqn. (5.23) into Eqn. (5.18).

$$\mathbf{v} = [\mathbf{I} - \nabla \mathbf{h}(\mathbf{x})^T \cdot (\nabla \mathbf{h}(\mathbf{x}) \cdot \nabla \mathbf{h}(\mathbf{x})^T)^{-1} \cdot \nabla \mathbf{h}(\mathbf{x})] \cdot \mathbf{x} + \mathbf{i}^{\text{conf}} \quad (5.25)$$

By the definition of the Jacobian, when \mathbf{v} leads to equilibrium point $\mathbf{h}(\mathbf{v})$ may be approximated as follows:

$$\mathbf{h}(\mathbf{v}) \approx \mathbf{h}(\mathbf{v}_c) + \mathbf{J} \cdot (\mathbf{v} - \mathbf{v}_c) \quad (5.26)$$

where $\mathbf{J} = \nabla \mathbf{h}(\mathbf{v})$

In the proximity of the equilibrium point $\mathbf{v}_c = 0$,

$$\lim_{\mathbf{v} \rightarrow \mathbf{v}_c} \frac{\|\mathbf{h}(\mathbf{v})\|}{\|\mathbf{v}\|} = 0 \quad (5.27)$$

Finally from Eqns. (5.25-5.27), \mathbf{v} can be written as

$$\mathbf{v} = \mathbf{v} - \nabla \mathbf{h}(\mathbf{v})^T \cdot ((\nabla \mathbf{h}(\mathbf{v}) \cdot \nabla \mathbf{h}(\mathbf{v})^T)^{-1}) \cdot \mathbf{h}(\mathbf{v}) \quad (5.28)$$

Parameters \mathbf{T}^{op} and \mathbf{i}^{op} in this case are such that the vector \mathbf{v} is updated in the opposite gradient direction of the energy function E^{op} . Since Eqns. (5.21) and (5.22) define a bounded convex polyhedron, the objective function (5.20) has a unique global minimum. Thus, the equilibrium points of the network can be calculated by assuming the following values of \mathbf{T}^{op} and \mathbf{i}^{op} ,

$$i^{op} = - \left[\frac{\partial f(v)}{\partial v_1}, \frac{\partial f(v)}{\partial v_2}, \dots, \frac{\partial f(v)}{\partial v_N} \right] \quad (5.29)$$

$$T^{op} = 0 \quad (5.30)$$

5.4.1 ESTIMATION ALGORITHM

The steps followed have been given as under:

Step 1: Get the system data, measurements and define the zero injection buses together with boundary limits on the state variables.

Step 2: Select an initial erroneous state vector, tolerance limit and set the iteration count.

Step 3: Calculate the objective function and say it $f(v)_{old}$.

Step 4: Calculate P_i and Q_i corresponding to equality constrained buses.

Step 5: Find $\nabla h(v)$ by differentiating zero injection equations w.r.t. state variables using load flow equations..

Step 6: Calculate updated state variables by Eqn. (5.28).

Step 7: Enforce the boundary limits by passing the state variables through a symmetrical ramp activation function defined by limits $[V_{max}, V_{min}]$ and $[\delta^{max}, \delta^{min}]$ corresponding to each state variable.

Step 8: Find i^{op} by differentiating the objective function w.r.t. state variables.

Step 9: Find Δv by Eqn. (5.19) and update v .

Step 10: Find the mismatch vector between measurements and calculated values and get its weighted squared sum to find out the new objective function value and find the difference between $f(v)_{new}$ and $f(v)_{old}$. If this difference is less than tolerance go next step, else go to step 3 after increasing the iteration count.

Step 11: Display the results and Stop.

5.5: TEST RESULTS

In this chapter, a simple 6 bus system [55] and the IEEE 14 bus system are used for simulation. The true values were obtained by the result of the load flow calculation, and the measurement values were made by adding ($\sigma=0.01$) errors to

those true values. As equality constraints, zero power injection at the nodes with no load and no generation are used.

(A): 6 bus system [55]

The measurement set base value for the 6 bus system is shown in Fig.5.2 and table (5.1). Bus no 3 and 4 are characterized as zero injection buses.

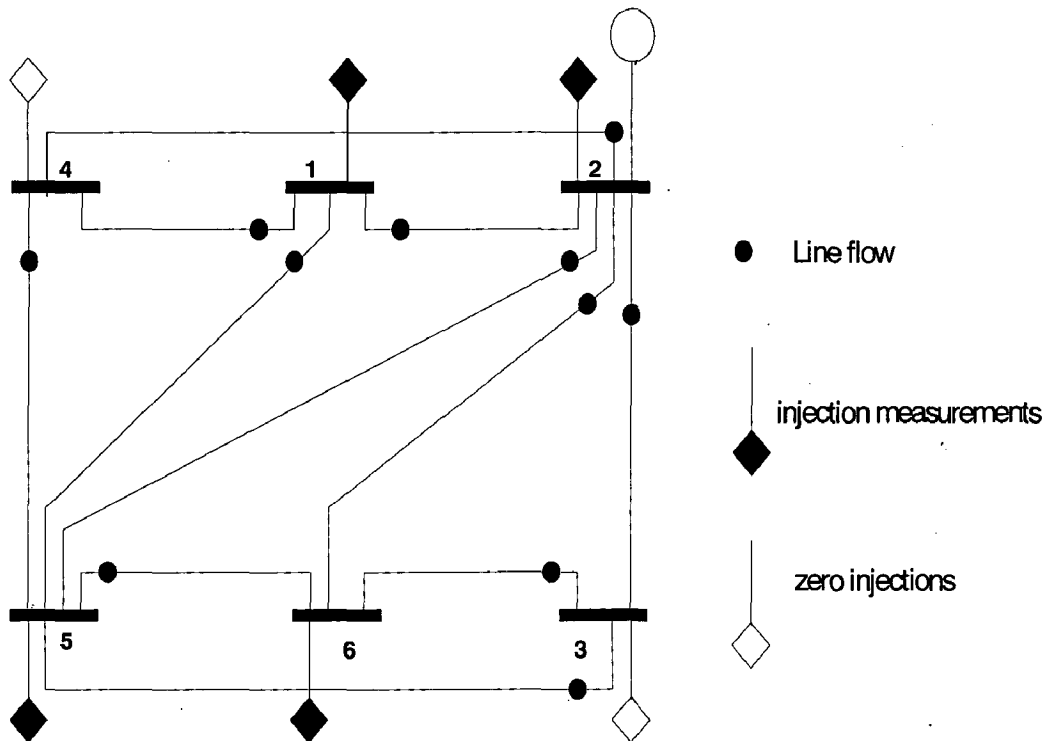


Figure 5.2: Measurement set for 6 bus system

Table 5.1

Measurements	Type	Buses	P	Q
z_1	Injection	1	0.9740	-0.0661
z_2	Injection	2	0.5005	0.5075
z_3	Injection	5	-0.7007	-0.7007
z_4	Injection	6	-0.7007	-0.7007
z_5	Line flow	1-2	0.2880	-0.1550
z_6	Line flow	1-4	0.2830	-0.0880
z_7	Line flow	1-5	0.4010	0.1760
z_8	Line flow	2-3	0.2310	0.1940
z_9	Line flow	2-4	-0.090	-0.0700

z_{10}	Line flow	2-5	0.2060	0.2110
z_{11}	Line flow	2-6	0.4320	0.0440
z_{12}	Line flow	3-5	0.0110	0.0520
z_{13}	Line flow	3-6	0.2150	0.1810
z_{14}	Line flow	4-5	0.1890	0.0900
z_{15}	Line flow	5-6	0.073	-0.044

The estimated state using the method with equality constraints are as shown in table 5.2

Table 5.2

Bus	1	2	3	4	5	6
V	1.0503	1.0494	0.9892	1.0503	0.9656	0.9683
δ	0	-4.7065	-7.6059	-3.8441	-6.9388	-8.8593

Following table shows the errors of the estimate values.

Table. 5.3

Measurements	ΔP	ΔQ
z_1	-0.021	0.0051
z_2	0.0068	0.0005
z_3	0.0037	-0.0003
z_4	0.0077	-0.0093
z_5	-0.0083	0.0008
z_6	-0.0068	-0.0013
z_7	-0.006	-0.0022
z_8	-0.0021	-0.0014
z_9	0.0046	-0.0131
z_{10}	-0.0001	-0.0016
z_{11}	-0.0033	-0.0233
z_{12}	0.0012	-0.0038
z_{13}	-0.0027	0.002

z_{14}	-0.0011	-0.0036
z_{15}	-0.0019	-0.0007

The energy mismatch ΔE was used for the convergence criteria with the tolerance 10^{-02} . The time step used was $\Delta t=10^{-04}$ in Eqn.(5.19).

The convergence characteristics of the energy function with respect to number of iterations is shown in Fig. 5.3.

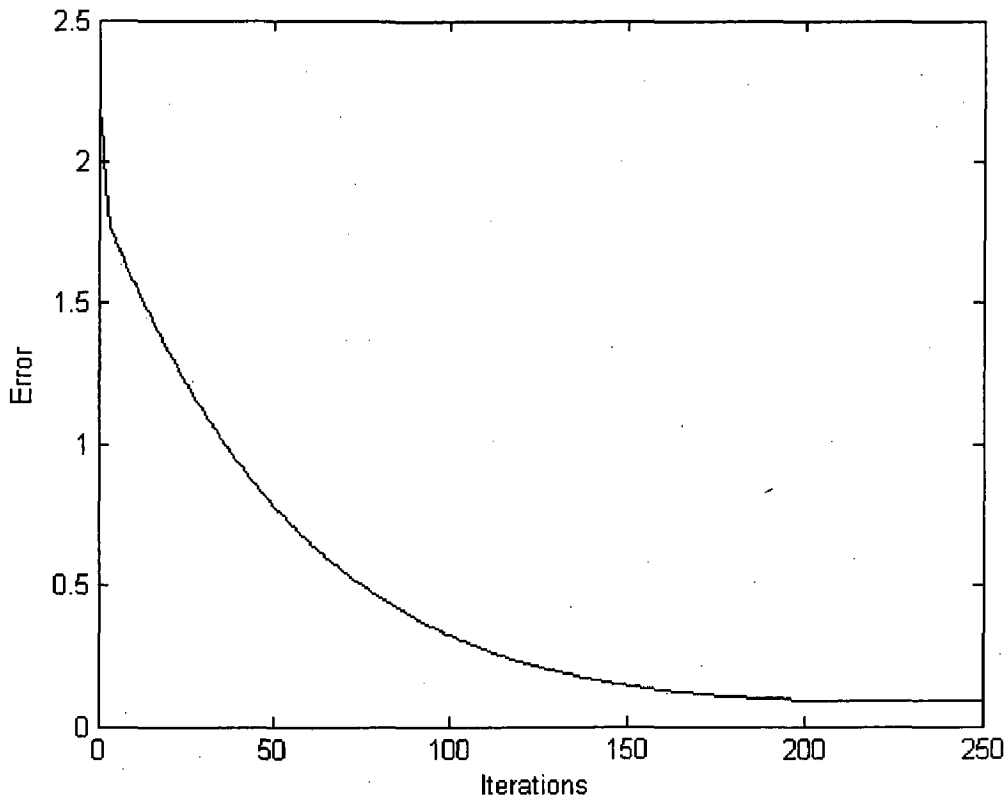


Figure 5.3: Convergence of energy function

(B) :IEEE 14 bus system

The measurement set base value for the IEEE 14 bus system is shown in Fig.5.4 and table (5.4). Bus no 5 and 7 are characterized as zero injection buses. The energy mismatch ΔE was used for the convergence criteria with the tolerance 10^{-05} . the time step used was $\Delta t=10^{-04}$ in Eqn. (5.19).

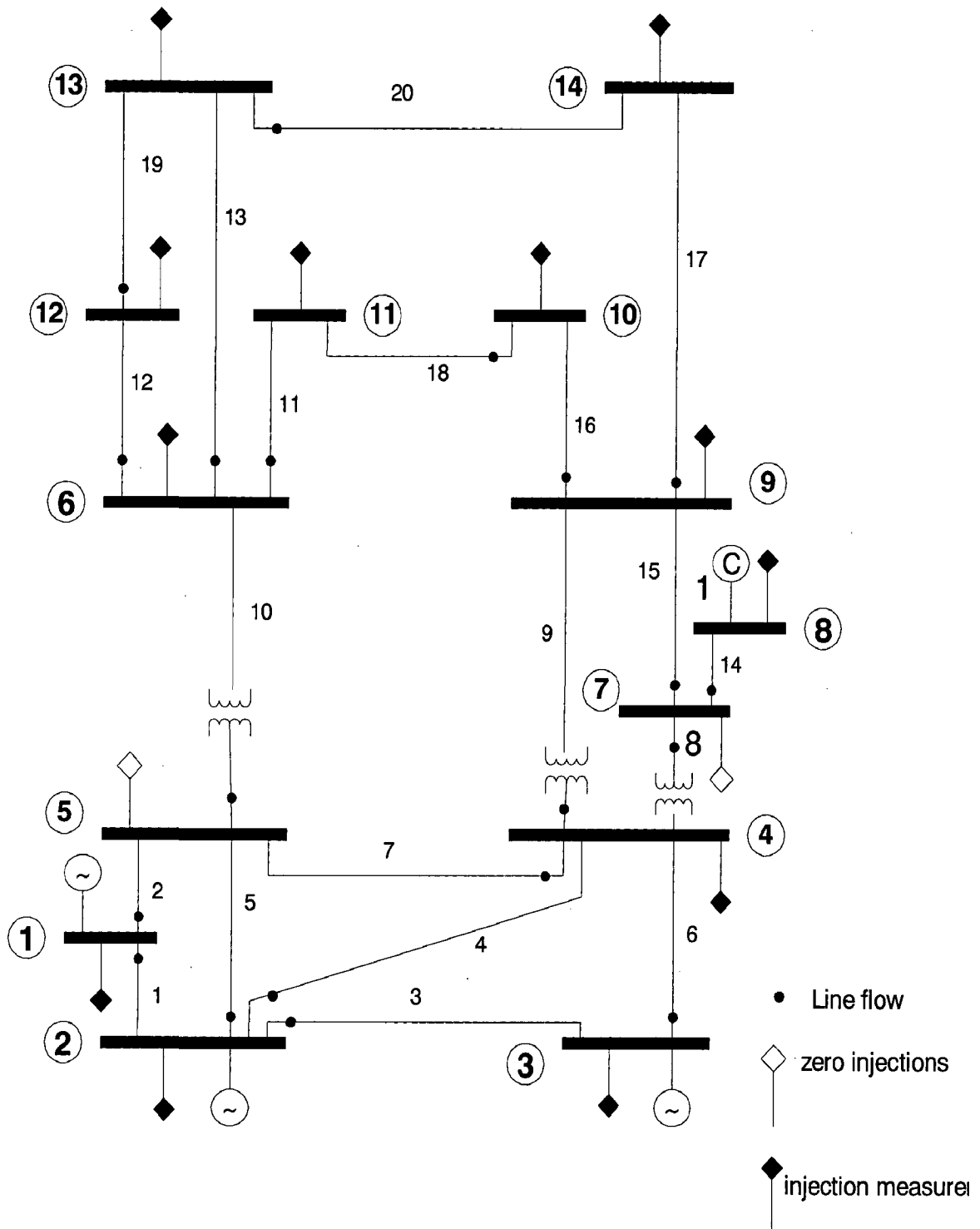


Figure 5.4: Measurement set for IEEE 14 bus system.

Table:5.4

Measurements	Type	Buses	P	Q
z ₁	Injection	1	2.2462	-0.1722
z ₂	Injection	2	0.1823	0.2535
z ₃	Injection	3	-0.9453	0.0426
z ₄	Injection	4	-0.4783	0.0704
z ₅	Injection	6	-0.1129	0.0344
z ₆	Injection	8	0.000	0.1733
z ₇	Injection	9	-0.2955	0.0234
z ₈	Injection	10	-0.0922	-0.0635
z ₉	Injection	11	-0.0327	-0.0125
z ₁₀	Injection	12	-0.061	-0.016
z ₁₁	Injection	13	-0.1366	-0.0605
z ₁₂	Injection	14	-0.1487	-0.0489
z ₁₃	Line flow	1-2	1.5196	-0.1628
z ₁₄	Line flow	1-5	0.7265	0.0479
z ₁₅	Line flow	2-3	0.7243	0.0603
z ₁₆	Line flow	2-4	0.5447	-0.0123
z ₁₇	Line flow	2-5	0.3926	0.0099
z ₁₈	Line flow	3-4	-0.2437	0.036
z ₁₉	Line flow	4-5	-0.6384	0.139
z ₂₀	Line flow	4-7	0.2806	-0.1972
z ₂₁	Line flow	4-9	0.1607	-0.0579
z ₂₂	Line flow	5-6	0.444	-0.1794
z ₂₃	Line flow	6-11	0.0737	0.035
z ₂₄	Line flow	6-12	0.0784	0.0256
z ₂₅	Line flow	6-13	0.1791	0.0745
z ₂₆	Line flow	7-8	0.000	-0.1688
z ₂₇	Line flow	7-9	0.2805	0.0714

z_{28}	Line flow	9-10	0.0521	0.0428
z_{29}	Line flow	9-14	0.0936	0.0348
z_{30}	Line flow	10-11	-0.0402	-0.021
z_{31}	Line flow	12-13	0.0166	0.008
z_{32}	Line flow	13-14	0.0568	0.0177

The state estimation results are shown below:

Bus No.	V	δ	Bus No.	V	δ
1	1.060	0	2	1.045	-4.731
3	1.010	-12.309	4	1.022	-9.615
5	1.024	-8.046	6	1.071	-12.68
7	1.062	-12.080	8	1.090	-11.922
9	1.055	-13.481	10	1.051	-13.553
11	1.058	-13.167	12	1.057	-13.296
13	1.051	-13.443	14	1.037	-14.258

Following table shows the errors of the estimate values.

Table: 5.5

Measurements	PROPOSED METHOD		NORMAL METHOD	
	ΔP	ΔQ	ΔP	ΔQ
z_1	0.0061	-0.0046	0.0037	-0.0019
z_2	0.0042	-0.0066	0.0018	-0.0061
z_3	0.0018	-0.0025	0.0028	-0.0028
z_4	0.0017	0.0023	0.0014	0.0024
z_5	-0.0017	-0.0051	-0.0016	-0.0022
z_6	-0.0018	0.0021	-0.0012	0.0081
z_7	-0.0017	0.0014	-0.0082	0.0126
z_8	-0.0011	0.0012	-0.0028	0.0155
z_9	0.0016	0.0022	0.0019	0.0657

z_{10}	0.0021	0.0055	0.0001	0.0509
z_{11}	0.0017	0.0016	0.0083	0.0852
z_{12}	-0.0023	-0.0066	-0.0405	-0.0067
z_{13}	0.0275	-0.0025	0.0329	-0.0087
z_{14}	0.0329	-0.0021	0.0161	-0.0433
z_{15}	0.0063	-0.0016	0.0173	-0.0147
z_{16}	0.0316	-0.0037	0.0128	-0.0046
z_{17}	0.0305	-0.0082	0.0085	-0.0054
z_{18}	0.0237	-0.0057	-0.0058	0.0013
z_{19}	-0.0063	-0.0138	-0.0018	-0.0129
z_{20}	0.0522	-0.0021	0.0096	-0.0276
z_{21}	0.0256	0.0015	0.0525	0.0058
z_{22}	0.0666	0.0095	0.0148	0.0499
z_{23}	0.0086	-0.0012	0.0012	-0.0057
z_{24}	0.0173	-0.0027	0.0003	-0.0046
z_{25}	0.0239	-0.0045	0.0001	-0.0086
z_{26}	0.0181	-0.0061	0.0126	-0.0074
z_{27}	0.0308	-0.0008	0.0083	-0.0016
z_{28}	0.0194	-0.0019	0.0243	-0.0081
z_{29}	0.0204	-0.0043	0.0298	-0.0043
z_{30}	0.0079	-0.0011	0.0047	-0.0075
z_{31}	-0.0034	0.0023	-0.0022	0.0032
z_{32}	0.0029	-0.0014	0.0011	-0.0041

The convergence characteristics of the energy function with respect to number of iterations is shown in Fig.5.5.

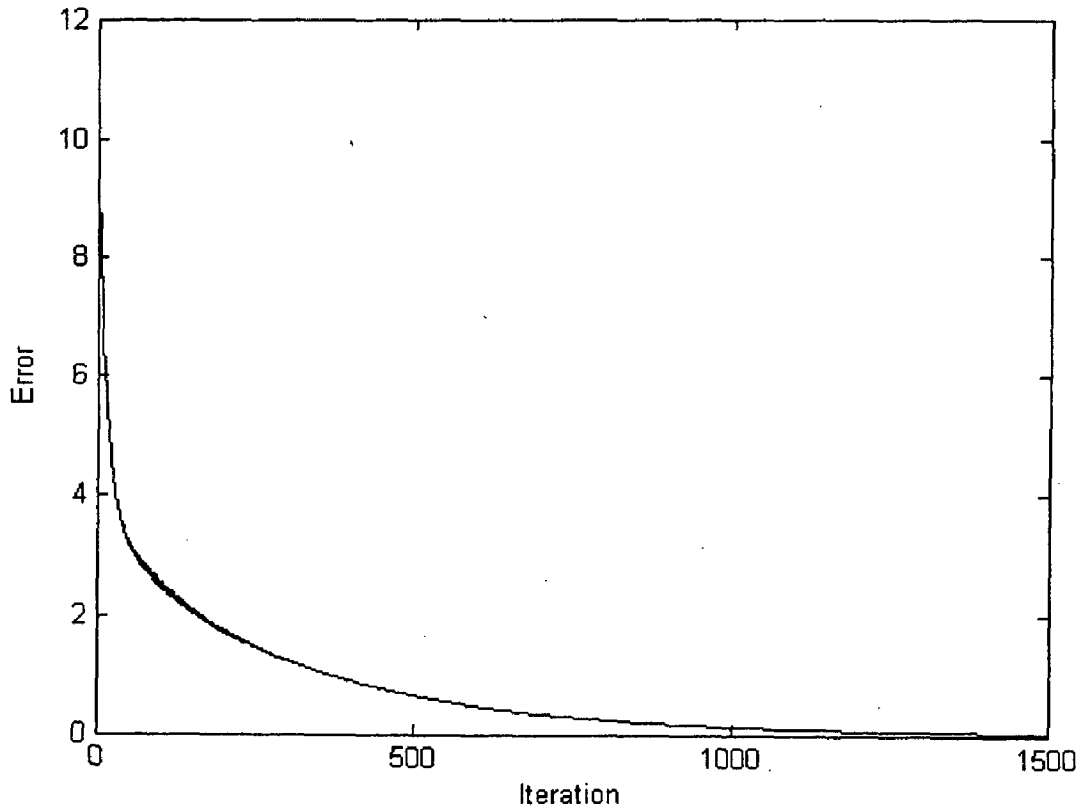


Figure 5.5: Convergence of energy function

The enforcing of limits and equality constraints increases the quality of state estimation results and produces the solution, which reflects correctly the physical behavior of a power system. In this chapter a modified Hopfield neural network method has been applied for solving state estimation problem, which neither require training nor testing of the neural network under human supervision and hence can be easily automated and applied to any large power system. The states of a power system are required to be estimated in real time and as for as conventional computer based algorithms are used; faster computation can not be expected. On the other hand, dedicated neural network processors [56] and special hardware devices are being designed and manufactured, and the computation time will become much shorter if the algorithm is implemented on neural network hardware.

CHAPTER 6

STATE ESTIMATION OF POWER SYSTEMS EMBEDDED WITH FACTS DEVICES

Flexible A.C. transmission systems (FACTS) are being used more in large power systems for their significance in manipulating line power flows. Traditional state estimation methods without integrating FACTS devices will not be suitable for power systems embedded with FACTS. In this chapter the state estimation of power systems in presence of FACTS devices is presented.

6.1 INTRODUCTION

After the establishment of power markets with transmission open access, the significance and use of FACTS devices for manipulating line power flows to relieve congestion and optimize the overall grid operation have increased. As a result, there is a need to integrate the FACTS device models into the existing power system applications. This chapter will present the modified Hopfield neural network algorithm for state estimation of network embedded with FACTS devices.

First a steady state model of UPFC is presented followed by load flow analysis of power system in presence of UPFC so as to obtain measurement vectors. Finally the algorithm discussed in chapter (5) is used to solve the state estimation problem.

6.2 FLEXIBLE AC TRANSMISSION SYSTEM (FACTS)

As a result of recent environmental legislation, rights of way issues, increase in construction cost and deregulation policies, there is an increasing recognition of the necessity to utilize existing transmission system assets to the maximum extent possible which can be achieved with the help of FACTS devices. The flexible ac transmission system is the result of related developments in electronic devices designed to overcome the limitations of traditional mechanically controlled power transmission systems. By using reliable, high-speed electronic controllers, the technology offers opportunity for

increased efficiency. Some advantages which FACTS devices can offer are;

- Greater control of power, so that it flows on prescribed transmission routes.
- Secure loading of transmission lines to level nearer their thermal limits.
- Prevention of cascading outages.
- Damping of power system oscillations.

The active power transmitted over an ac transmission line is defined by the Eqn. (6.1)

$$P = \frac{V_1 V_2}{X} \sin(\delta_1 - \delta_2) \quad (6.1)$$

Where V_1 and V_2 are the voltages at the ends of the transmission line. X is the Equivalent impedance of the transmission line, and $\delta_1 - \delta_2$ is the phase angle difference between both ends of transmission line. From the Eqn. (6.1) it is evident that the transmitted power is a function of three parameters: the magnitude of sending end voltage and receiving end voltage, impedance and voltage angle difference. Traditional techniques of reactive line compensation and step like voltage adjustment are generally used to alter these parameters to achieve power transmission control. Different type of facts devices [57] can be used to control one or more of these parameters to control the existing power system network. The fast response of FACTS devices improves the controllability of the system and thus makes the system more versatile.

6.2.1 ROLE OF FACTS DEVICES IN POWER SYSTEM

FACTS devices play an important role in enhancing performances of power system. Some are given below:

- 1) FACTS devices by controlling power, can improve the power system performance considerably such as improvement of power quality and security of the supply. FACTS technology can contain cascading outages by limiting the impact of multiple faults, thereby improving the reliability of power supply. Upgrading of transmission lines can increase power quality by increasing voltage and/or current capacity with the help of these devices.
- 2) The "free flow" mode of power system operation may be changed into a controlled power flow mode of operation due to powerful controllability of FACTS technology where the power flow in one or more transmission lines is controlled in

predetermined manner.

- 3) FACTS technology can increase secure loading of transmission lines their steady state, short time and dynamic limits. Thus it enhances Transient, voltage and small signal stability of power system.
- 4) Due to high capital cost of high voltage transmission, cost considerations are main concerns. Although the price of FACTS devices is high, compared to other methods of solving transmission loading problem, yet FACTS technology is probably the most viable resort because of their ability to effectively control power flows, the power system can be operated in more optimized situation. As a result, large amount of money may be saved. Also, as people pay more and more attention to the environmental impact of new projects. Thus, it becomes very common for environmental opposition to frustrate attempts to established new transmission routes. Using FACTS technology, however, it is possible to transfer more power over existing routes, thus meeting consumer demand without the construction of new transmission lines.
- 5) The static performance of power system can be improved significantly with these devices leading to losses reduction, Security enhancement, congestion management and available transfer capability enhancement.

6.2.2 BASIC TYPES OF FACTS CONTROLLERS

In this context, FACTS technology has been rapidly developed in the last ten years. The first FACTS device is Static Var Compensation (SVC) and has been in service for nearly last two decades. It is a shunt device to maintain a healthy voltage profile in system. TCSC, TCSR are series controllers used to control line flows. Three Thyristor Controlled Series Compensation (TCSC) projects have been working successfully in USA since 1991. In general, FACTS controller can be divided into four categories, the general symbols for which is shown in Fig. 6.1-Fig.6.4.

- Series Controller
- Shunt Controller
- Combined series-series Controller
- Combined series-shunt controller

Series controllers

The series controller could be variable impedance, such as capacitor, reactor, etc or power electronics based variable source of main frequency, sub synchronous and harmonic frequencies (or a combination) to serve the desired need. In principle, all series controller inject voltage in series with the line. Even variable impedance multiplied by the current flow through it represents an injected series voltage in the line. As long as the voltage is in phase quadrature with the line current, the series controller only supplies or consumes variable reactive power. Any other phase relationship will involve handling of real power as well. Common type or series FACTS controllers, which are generally used, are:

- **Static synchronous series compensator(SSSC)**
- **Thyristors controlled series capacitor(TCSC)**
- **Thyristor switched series capacitor(TSSC)**
- **Thyristor switched series reactor (TSSR)**
- **Thyristor controlled series reactor(TCSR)**

Shunt controller

As in the case of series controller, the shunt controllers may be variable impedance, variable source, or a combination of these. In principle, all shunt controllers inject current into the system at the point of connection. Even variable shunt impedance connected to line causes a variable current flow and hence represents injection of current into the line .as long as the injected current is in phase quadrature with the line voltage, the shunt controller only supplies or consumes variable reactive power. Any other phase relationship will involve handling of real power as well. The common type of shunt controllers in use are:

- **Static Synchronous Compensator(STATCOM)**
- **Static VAR Compensator(SVC)**
- **Thyristor Controlled Reactor(TCR)**
- **Thyristor Switched Reactor(TSR)**
- **Thyristor Switched Capacitor(TSC)**

Combined series-series controllers

This could be a combination of separate series controllers, which are controlled in a coordinated manner, in a multi-line system or it could be a unified controller in which series controllers provide independent series reactive compensation for each line but also transfer real power among the lines via the power link. The real power transfer capability of the unified series-series controller, referred to as interlink power flow controller, makes it possible to balance both the real and reactive power flow in the lines and thereby maximize the utilization of the transmission system.

Combined series-shunt controllers

This could be a combination of separate shunt and series controllers, which are controlled in a coordinated manner or a unified power flow controller with series and shunt elements. In principle, combined shunt and series controllers inject current into the system with the shunt part of the controller. However, when the shunt and series controllers are unified, there can be a real power exchange between the series and shunt controllers via the power link. Common types of these controllers are:

- **Unified Power Flow Controller(UPFC)**
- **Thyristor Controlled Phase Shifting Transformer(TCPST)**

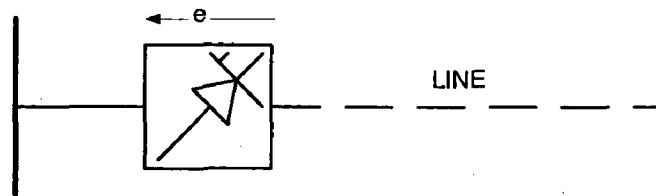


Fig. 6.1: Series Controllers

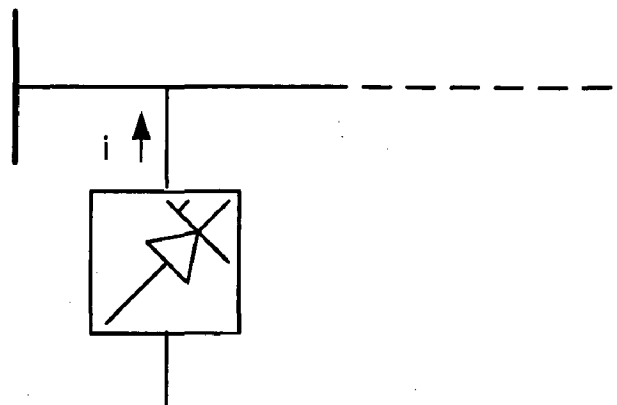


Fig. 6.2: Shunt Controller

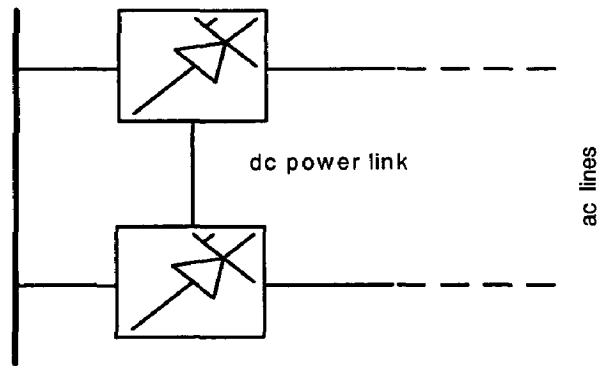


Fig. 6.3: Unified Series-Series Controller

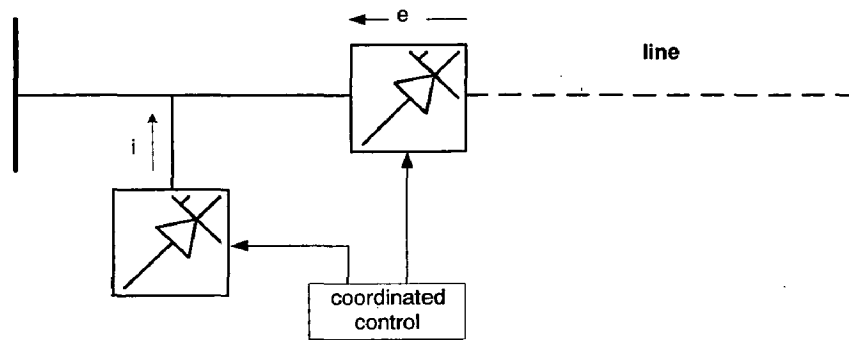


Fig. 6.4: Coordinated Series and Shunt Controller

The different FACTS devices are helpful in controlling different parameters effectively. Generally the FACTS devices with both series and shunt controllers are most versatile devices. The operation for which a particular device can be used is given in the table 6.1. The implementation of any of the new FACTS controllers is not an easy task. Although they offer substantial advantages for steady state and dynamic operation by controlling the power flow in the transmission line, it brings major challenges in power electronics, device control and protection design which involves huge cost and efforts.

Table 6.1

FACTS CONTROLLERS	CONTROLLED PARAMETERS
Static Synchronous Compensator (SSC)	Voltage Control, VAR Compensation, Damping Oscillations, Voltage Stability
Static VAR Compensator (SVC, TCR, TCS, TRS)	Voltage Control, VAR Compensation, Damping Oscillations, Transient and Dynamic Stability Voltage Stability
Thyristor-Controlled Breaking Resistor (TCBR)	Damping Oscillations, Transient And Dynamic Stability
Static Synchronous Series Compensator(SSSC)	Current Control, Damping Oscillations, Transient and Dynamic Stability, Voltage Stability, Fault Current Limiting
Thyristor Controlled Series Capacitor (TCSC,TSSC)	Current Control, Damping Oscillations, Transient and Dynamic Stability, Voltage Stability, Fault Current Limiting
Thyristor Controlled Series Reactor (TCSR,TSSR)	Current Control, Damping Oscillations, Transient and Dynamic Stability, Voltage Stability, Fault Current Limiting
Thyristor Controlled Phase Shifting Transformer(TCPR)	Active Power Control ,Damping Oscillations, Transient and Dynamic Stability, Voltage Stability
Unified Power Flow Controller(UPFC)	Active and Reactive Power Control, Voltage Control, VAR Compensation, Damping Oscillations
Interline Power Flow Controller(IPFC)	Reactive Power Control, Voltage Control, Damping Oscillations, Transient and Dynamic Stability, Voltage Stability

6.3 UNIFIED POWER FLOW CONTROLLER (UPFC)

The unified power flow controller (UPFC) is arguably the most comprehensive device to have emanated so far from the FACTS initiative. In principle at least, the UPFC offers new horizons in terms of power system control, with the potential to independently control three power system parameters namely bus voltage, line active and reactive power. The simplest UPFC consist of two converters one connected in shunt and one connected in series with transmission lines in a substation. It can control three quantities such as a bus voltage and independent active and reactive power flow of the line. The real power is exchanged among shunt and series converters via a common DC link. From the literature it is found that unified power flow controllers are very powerful and can give flexible and effective power flow control in power system.

The UPFC was first proposed by Gyugyi in 1991. It was devised for real-time control and dynamic compensation of ac transmissions system, providing multifunctional flexibility required to solve many of the problems facing the power delivery industry. From conceptual view-point, UPFC is a generalized synchronous voltage source, represented at the fundamental frequency by voltage phasor V_{s1} with controllable magnitude ($0 \leq V_{s1} \leq V_{s1max}$) and angle ($0 \leq \phi_{s1} \leq 2\pi$) in series with the transmission line. As far as construction is concerned a UPFC consists of shunt (exciting and series (boosting) transformer, which are connected by two voltage-sourced converters using GTO thyristors valves and a DC circuit. Inverter-2 is used to generate a voltage source at the fundamental frequency with variable amplitude ($0 \leq V_{s1} \leq V_{s1max}$) and phase angle ($0 \leq \phi_{s1} \leq 2\pi$), which is added to the AC transmission line by the series connected booster transformer. As the series transformer injects series voltage in line, the control of active and reactive power is possible by changing the magnitude and angle of inserted voltage. The real power flows from shunt converter to series converter via a DC link. As both inverters are capable of handling reactive power independently shunt transformer can also inject reactive power on the bus thus helps in maintaining better voltage profile. In this way the inverter output voltage injected in series with the line can be used for direct voltage control, series compensation, phase shifting and their combination and shunt current can be used to maintain good voltage profile. The

schematic diagram of UPFC is shown in Fig 6.5.

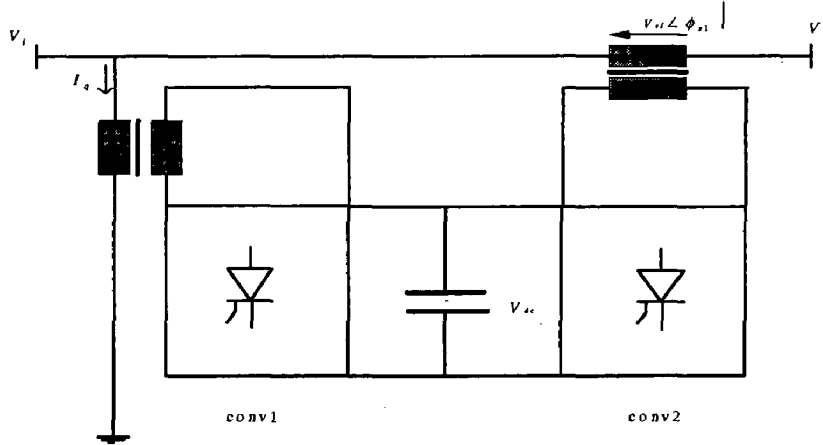


Fig. 6.5: The UPFC schematic diagram

6.3.1 STATIC MODELING OF UPFC

In this section the modeling of unified power flow controller (UPFC) has been presented for static analysis. Let there be n numbers of lines connected at bus- i . Let a UPFC be placed at the i^{th} end of the line- k having impedance $r_{ij} + jx_{ij}$ ($=1/(g_{ij} + jb_{ij})$) connected between bus- i and bus- j . UPFC has three controllable parameters, namely the magnitude and the angle of inserted voltage (V_{sl}, ϕ_{sl}) in line- k and the magnitude of the current (I_q). The circuit diagram is given in Fig. 6.6.

Based on the principle of UPFC operation and the circuit diagram, the basic mathematical relations can be written as,

$$I_{ij} = (V_i + V_{sl} - V_j) y_{ij} \quad (6.2)$$

$$\text{Arg}(I_q) = \text{Arg}(V_i) \pm \pi/2, \text{Arg}(I_T) = \text{Arg}(V_i), \quad (6.3)$$

$$I_T^* = \frac{\text{Re}[V_{sl} I_{ij}^*]}{V_i} \quad (6.4)$$

The power injection at bus- i can be written as

$$S_i = P_i + jQ_i = V_i I_{ij}^* + V_i (I_T + jI_q)^* + \sum_{\substack{i=1 \\ \neq j}}^n V_i I_{in}^* + V_i I_{sh}^* \quad (6.5)$$

Where, I_{sh} is the shunt current due to line charging.

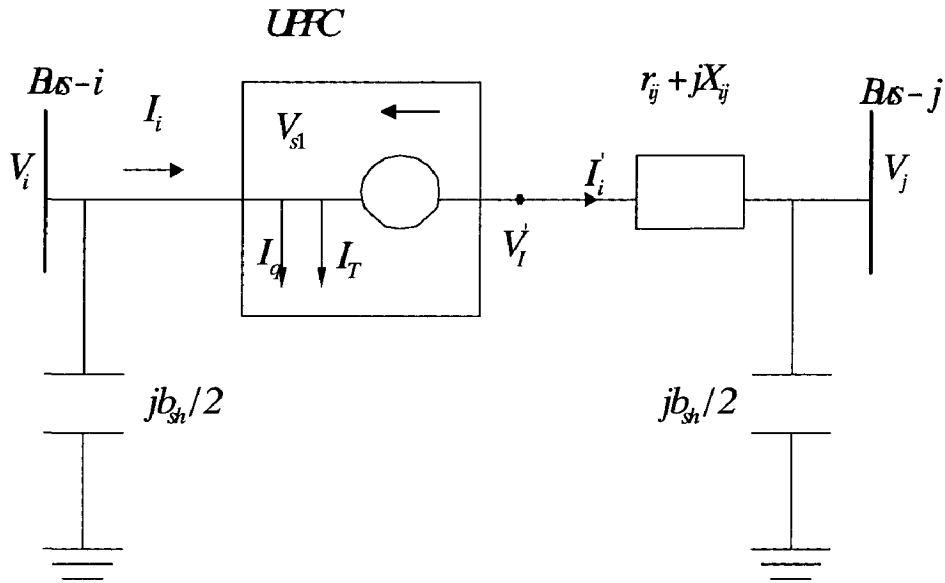


Fig. 6.6 Circuit diagram of UPFC

The effect of UPFC can be represented as injected power with the network as shown in Fig. 6.7. The injected complex powers $S_{ig} (= P_{ig} + jQ_{ig})$ at bus- i and $S_{jg} (= P_{jg} + jQ_{jg})$ at bus- j can be written as,

$$S_{ig} = S_i^0 - S_i = \{V_i V_{sl}^* y_{ij}^* + V_i (I_T + jI_q)^*\} \quad (6.6)$$

$$S_{jg} = S_j^0 - S_j = V_j V_{sl}^* y_{ij}^* \quad (6.7)$$

Where, S^0 is the complex power injection when there was no UPFC.

From Eqn. 3.5, the real and reactive power injections at bus- i can be derived as

$$P_{ig} = -\text{Re}(V_i V_{sl}^* y_{ij}^*) - V_i I_T^* \quad (6.8)$$

$$Q_{ig} = -\text{Im}(V_i V_{sl}^* y_{ij}^*) + V_i I_q \quad (6.9)$$

From Eqn. (6.4), we have

$$\begin{aligned}
V_i I_T^* &= \text{Re} \left\{ -V_{s1} \angle \phi_{s1} \times \left(V_i \angle \delta_i + V_{s1} \angle \phi_{s1} - V_j \angle \delta_j \right) \times y_{ij}^* \right\} \\
&= \text{Re} \left\{ \left(V_{s1} V_i \angle (\phi_{s1} - \delta_i) + V_{s1}^2 - V_{s1} V_j \angle (\phi_{s1} - \delta_j) \right) \times (g_{ij} - j b_{ij}) \right\}
\end{aligned} \tag{6.10}$$

Thus,

$$\begin{aligned}
V_i I_T &= V_{s1}^2 g_{ij} + V_{s1} V_i \left[g_{ij} \cos(\phi_{s1} - \delta_i) + b_{ij} \sin(\phi_{s1} - \delta_i) \right] \\
&\quad - V_{s1} V_j \left[g_{ij} \cos(\phi_{s1} - \delta_j) + b_{ij} \sin(\phi_{s1} - \delta_j) \right]
\end{aligned} \tag{6.11}$$

The real and imaginary values of $V_i V_{s1}^* y_{ij}^*$ can be written as,

$$\text{Re} \left(V_i V_{s1}^* y_{ij}^* \right) = V_i V_{s1} \left(g_{ij} \cos(\delta_i - \phi_{s1}) + b_{ij} \sin(\delta_i - \phi_{s1}) \right) \tag{6.12}$$

$$\text{Im} \left(V_i V_{s1}^* y_{ij}^* \right) = V_i V_{s1} \left(g_{ij} \sin(\delta_i - \phi_{s1}) - b_{ij} \cos(\delta_i - \phi_{s1}) \right) \tag{6.13}$$

The injected active and reactive powers at bus- i will be

$$P_{ig} = -V_{s1}^2 g_{ij} - 2V_{s1} V_i g_{ij} \cos g_{ij} \sin(\delta_i - \phi_{s1}) - b_{ij} \cos(\phi_{s1} - \delta_i) + V_{s1} V_j \left[g_{ij} \cos(\phi_{s1} - \delta_j) + b_{ij} \sin(\phi_{s1} - \delta_j) \right] \tag{6.14}$$

$$Q_{ig} = V_i I_q + V_i V_{s1} \left[g_{ij} \sin(\phi_{s1} - \delta_i) + b_{ij} \cos(\phi_{s1} - \delta_i) \right] \tag{6.15}$$

Similarly the real and reactive powers injections at bus- j and bus- h can be derived as

$$P_{jg} = V_j V_{s1} \left(g_{ij} \cos(\phi_{s1} - \delta_j) - b_{ij} \sin(\phi_{s1} - \delta_j) \right) \tag{6.16}$$

$$Q_{jg} = -V_j V_{s1} \left(g_{ij} \sin(\phi_{s1} - \delta_j) + b_{ij} \cos(\phi_{s1} - \delta_j) \right) \tag{6.17}$$

So, A UPFC can be represented as injected complex powers $S_{ig} (= P_{ig} + jQ_{ig})$ at bus- i , and $S_{jg} (= P_{jg} + jQ_{jg})$ at bus- j , into the network as shown in Fig 6.7.

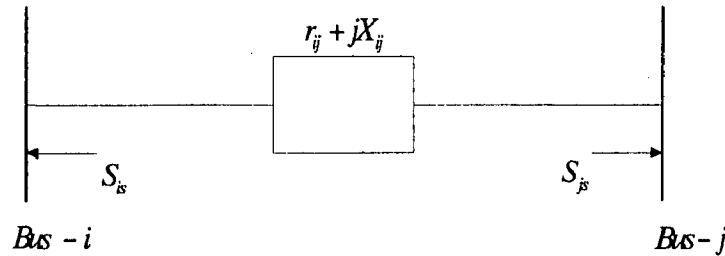


Fig.6.7: Injection model of UPFC

6.4 LOAD FLOW CALCULATION OF POWER SYSTEM WITH UPFC

In view of fact that power systems nowadays are becoming more openly accessible; maneuverability of their power flow continues to be a general concern in the coming decade. Earlier power transmission was mainly mechanically controlled by means of, say, on-load tap changing of transformers and switching in and out of circuits, which is obviously not flexible enough to cope with the future development. UPFC promptly becomes a promising device as it incorporates all three attributes for controlling the power flow. Load flow computation for power system is clearly in need for performing other essential functions such as power system analysis and planning. In [20], a useful power injection modeling of UPFC is developed and employed for performing the load flow control studies. Power flow (or load flow) analysis involves the calculation of power flows and voltages of a power system for a given set of bus bar loads, active power generation schedule and specified bus bar voltage magnitude conditions. Such calculations are widely used in the analysis and design of steady state operation as well as dynamic performance of the system. The power flow problem is formulated as a set of nonlinear equations. Power flow analysis is of great importance in planning and designing the future expansion of power system as well as in determining the best operation of existing system. The power flow problem consists of a given transmission network where all lines are represented by π -equivalent circuit and transformers by an ideal voltage transformer in series with an impedance.

Many calculation methods have been proposed to solve this problem. Among them, Newton-Raphson method and fast-decoupled load flow method are two very successful methods. In general, the decoupled power flow methods are only valid for weakly loaded network with large X/R ratio network. For system conditions with large angles across lines (heavily loaded network) and with special control devices (FACTS devices such as UPFC) that strongly influence active and reactive power flows, N-R method may be required. Therefore, when the AC power flow calculation is needed in systems with FACTS devices, N-R method should be used.

The UPFC model is incorporated into an existing Newton-Raphson load flow

algorithm .The modified Jacobian matrix and power mismatch equations are deduced based on the injection model of UPFC to control active and reactive powers and voltages magnitude in any combination.

6.4.1 MODIFIED NONLINEAR POWER FLOW EQUATION

The effect of UPFC on power system can be modeled as injected power flow at two related buses as shown above, thus they have no effect on node admittance matrix.

At each bus four quantities are associated, namely active power P, reactive power Q, voltage magnitude V and voltage angle δ . However, only two quantities are specified, at each bus, prior to the load flow. Depending upon the quantities specified there are three types of buses in the system:

- (1) Voltage Controlled (PV) Bus: Active power and voltage are specified. Limits to inject reactive power are specified depending upon the capacity of individual device. This type of bus corresponds to generator, synchronous condenser and SVC.
- (2) Load (PQ) Bus: Active and reactive power are specified. This type of bus generally corresponds to a load centre. Both active and reactive power is assumed to be constant irrespective of voltage variations.
- (3) Slack Bus: Voltage magnitude and phase angle are specified. Because the system losses are not known in advance of power flow calculation, the total injected power cannot be specified at all the buses. It is usual to choose one of the available generator-bus as slack-bus, and its voltage magnitude and angle are specified.

The function of the load flow is to find voltage and angle at all the buses, active and reactive power generation schedule. The problem is formulated as a set of nonlinear equations which can be solved by number of numerical methods including Newton-Raphson method which is very efficient and reliable. The load flow equations at bus i for the system having n buses and without UPFC, can be expressed as below

$$P_{is} = P_{Gi} - P_{Li} = V_i \sum_{j=1}^n V_j (G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) \quad (6.18)$$

$$Q_{is} = Q_{Gi} - Q_{Li} = V_i \sum_{j=1}^n V_j (G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) \quad (6.19)$$

Where P_{Gi} and Q_{Gi} are the real and reactive power generated at bus i . P_{Li} and Q_{Li} are real and reactive power load at bus i .

For each PQ and PV bus, there is an active power mismatch equation and for each PQ node, there is reactive power mismatch equation. These equations can be formulated as follow

$$\Delta P_i = P_{is} - V_i \sum_{j=1}^n V_j (G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) \quad i = 1, 2, \dots, n-1 \quad (6.20)$$

$$\Delta Q_i = Q_{is} - V_i \sum_{j=1}^n V_j (G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) \quad i=1, 2, \dots, q \quad (6.21)$$

Where P_{is} and Q_{is} are injected bus-generated powers. q is the number of network PQ buses. Here, the n^{th} bus is supposed to be the slack bus. G_{ij} and B_{ij} are the ij^{th} elements of Y_{bus} matrix.

If the series transformer of UPFC is installed in branch l (connected between bus- k to bus- p), the power mismatch equations at bus- k and bus- p will be modified as

$$\Delta P_k = P_{ks} - P_{kg} - V_k \sum_{j=1}^n V_j (G_{kj} \cos \delta_{kj} + B_{kj} \sin \delta_{kj}) \quad (6.22)$$

$$\Delta Q_k = Q_{ks} - Q_{kg} - V_k \sum_{j=1}^n V_j (G_{kj} \sin \delta_{kj} - B_{kj} \cos \delta_{kj}) \quad (6.23)$$

$$\Delta P_p = P_{ps} - P_{pg} - V_p \sum_{j=1}^n V_j (G_{pj} \cos \delta_{pj} + B_{pj} \sin \delta_{pj}) \quad (6.24)$$

$$\Delta Q_p = Q_{ps} - Q_{pg} - V_p \sum_{j=1}^n V_j (G_{pj} \sin \delta_{pj} - B_{pj} \cos \delta_{pj}) \quad (6.25)$$

The injected active power at buses (P_{kg} and P_{pg}), and reactive powers (Q_{kg} and Q_{pg}) having a UPFC are calculated using equations (eq.6.14-eq.6.17). Thus, the relationship are obtained for small variations in V and δ , by forming the total differentials,

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J_1] \begin{bmatrix} \Delta \delta \\ \Delta V/V \end{bmatrix} + [J_2] \begin{bmatrix} \Delta \delta \\ \Delta V/V \end{bmatrix} \quad (6.26)$$

$$[J] = [J_1] + [J_2] \quad (6.27)$$

Where J_1 is the normal N-R power flow Jacobian matrix and J_2 is the partial derivative matrices of injected power with respect to the variables. When bus- k and bus- p are PQ

buses, the matrix J_2 may have 16 nonzero elements (eq. (6.28) to eq. (6.43)). If bus- k or bus- p is a PV bus, the corresponding elements of row and column will not exist. When more than one UPFC are installed in the network, their effects are added to matrix J_2 . In this situation the non-zero elements may be more than 16. Now we can see that the power flow can be solved by N-R method in the normal way except the small differences in J matrix and power mismatch equations. The elements of J_2 are given below

$$\frac{\partial P_{kg}}{\partial \delta_k} = -2V_{s1}V_l g_{kp} \sin(\phi_{s1} - \delta_k) - 2V_{s2}V_k g_{ko} \sin(\phi_{s2} - \delta_k) \quad (6.28)$$

$$\frac{\partial P_{kg}}{\partial \delta_p} = V_{s1}V_p [g_{kp} \sin(\phi_{s1} - \delta_p) - b_{kp} \cos(\phi_{s1} - \delta_p)] \quad (6.29)$$

$$\frac{\partial P_{pg}}{\partial \delta_p} = V_{s1}V_p [g_{kp} \sin(\phi_{s1} - \delta_p) + b_{kp} \cos(\phi_{s1} - \delta_p)] \quad (6.30)$$

$$\frac{\partial P_{pg}}{\partial \delta_k} = 0 \quad (6.31)$$

$$\frac{\partial P_{kg}}{\partial V_k} = -2V_{s1}g_{kp} \cos(\phi_{s1} - \delta_k) - 2V_{s2}g_{ko} \cos(\phi_{s2} - \delta_k) \quad (6.32)$$

$$\frac{\partial P_{kg}}{\partial V_p} = V_{s1} [g_{kp} \cos(\phi_{s1} - \delta_p) + b_{kp} \sin(\phi_{s1} - \delta_p)] \quad (6.33)$$

$$\frac{\partial P_{pg}}{\partial V_p} = V_{s1} [g_{kp} \cos(\phi_{s1} - \delta_p) - b_{kp} \sin(\phi_{s1} - \delta_p)] \quad (6.34)$$

$$\frac{\partial P_{pg}}{\partial V_k} = 0 \quad (6.35)$$

$$\frac{\partial Q_{kg}}{\partial \delta_k} = V_{s1}V_k [-g_{kp} \cos(\phi_{s1} - \delta_k) + b_{kp} \sin(\phi_{s1} - \delta_k)] \quad (6.36)$$

$$\frac{\partial Q_{kg}}{\partial \delta_p} = 0 \quad (6.37)$$

$$\frac{\partial Q_{pg}}{\partial \delta_p} = -V_{s1}V_p [-g_{kp} \cos(\phi_{s1} - \delta_p) + b_{kp} \sin(\phi_{s1} - \delta_p)] \quad (6.38)$$

$$\frac{\partial Q_{pg}}{\partial \delta_k} = 0 \quad (6.39)$$

$$\frac{\partial Q_{kg}}{\partial V_k} = I_q + V_{sl} [g_{kp} \sin(\phi_{sl} - \delta_k) + b_{kp} \cos(\phi_{sl} - \delta_k)] \quad (6.40)$$

$$\frac{\partial Q_{kg}}{\partial V_p} = 0 \quad (6.41)$$

$$\frac{\partial Q_{pg}}{\partial V_p} = -V_{sl} [g_{kp} \sin(\phi_{sl} - \delta_p) + b_{kp} \cos(\phi_{sl} - \delta_p)] \quad (6.42)$$

$$\frac{\partial Q_{pg}}{\partial V_k} = 0 \quad (6.43)$$

With help of Eqn. (6.28-6.43) the power flow Jacobian matrix can be modified and power flow equations can be solved as conventional N-R method.

6.4.2. ALGORITHM FOR SOLVING LOAD FLOW EQUATIONS

The algorithm for modified load flow solution is given below:

1. From the given input data, form system admittance matrix.
2. Give initial value of the magnitude and phase angle of all bus voltages and $k = 0$.
3. Calculate all needed node power mismatches by Eqn. (6.20) and (6.21) and J_1 .
4. Superimpose modification into power mismatches by Eqn.s (6.22 to 6.25) and calculate J_2 .
5. Is the absolute value of the maximum mismatch less than tolerance?
6. If yes, the load flow solution obtained, stop.
7. If tolerance is not satisfied then amend injected powers and jacobian matrix by adding appropriate partial derivatives, find change in bus voltages and load angle and update bus voltages and load.
8. $k = k+1$, go to step 3.

6.4.3. RESULTS OF LOAD FLOW

The load flow calculation has been carried out on the IEEE 14 bus test data, by modifying an existing N-R load flow program to allow UPFC model. In the modified load flow computation an accuracy tolerance of less than 10^{-4} pu in respect of the

maximum absolute mismatch of nodal power injections are adapted, Load flow with and without UPFC has been carried out. For IEEE 14-bus system, solution converged in 5 iterations with tolerance of 0.0001 without UPFC while with UPFC it takes 6 iterations with tolerance of 0.00003. The results have been displayed with and without UPFC. The parameters of UPFC were set as $(V_{sl}, \varphi_{sl} Iq) = (.1, .5, .1)$.

Without UPFC

<i>Bus no.</i>	<i>Bus-Type</i>	V_m	<i>Angle</i>
1	<i>Bus-Type</i>	1.060	0
2	1	1.045	-4.807
3	2	1.010	-12.44
4	2	1.021	-9.996
5	0	1.024	-8.425
6	0	1.070	-13.872
7	2	1.062	-13.022
8	0	1.090	-13.022
9	2	1.055	-14.596
10	0	1.050	-14.753
11	0	1.057	-14.444
12	0	1.055	-14.728
13	0	1.050	-14.808
14	0	1.035	-15.689

Total System Loss = 0.13440+j 0.43226 p.u.

Line flows

<i>From</i>	<i>bus</i>	<i>Line flow (pu)</i>	<i>Line flow (pu)</i>
1	2	1.0587 + 0.0740i	-1.0392 - 0.0724i
1	5	0.4935 + 0.0566i	-0.4815 - 0.0601i
2	3	0.6197 + 0.0772i	-0.6025 - 0.0498i
2	4	0.3425 - 0.0310i	-0.3361 + 0.0143i

2	5	0.2600 - 0.0121i	-0.2564 - 0.0136i
3	4	-0.3395 - 0.0121i	0.3473 + 0.0190i
4	5	-0.3531 + 0.0936i	0.3548 - 0.0882i
4	7	0.2088 - 0.1629i	-0.2088 + 0.1769i
4	9	0.1198 - 0.0713i	-0.1198 + 0.0816i
5	6	0.3571 - 0.1106i	-0.3571 + 0.1444i
6	11	0.0159 + 0.0084i	-0.0159 - 0.0084i
6	12	0.0774 + 0.0221i	-0.0767 - 0.0206i
6	13	0.1778 + 0.0605i	-0.1757 - 0.0564i
7	8	0.0000 - 0.0000i	-0.0000 + 0.0000i
7	9	0.2135 - 0.0611i	-0.2135 + 0.0660i
9	10	0.1096 + 0.0690i	-0.1092 - 0.0677i
9	14	0.1980 + 0.0583i	-0.1932 - 0.0481i
10	11	0.0192 + 0.0097i	-0.0191 - 0.0096i
12	13	0.0157 + 0.0046i	-0.0156 - 0.0046i
13	14	0.0563 + 0.0030i	-0.0558 - 0.0019i

With UPFC in line 12(between nodes 6 and 12)

<i>Bus no.</i>	<i>Bus-Type</i>	V_m	<i>Angle</i>
1	1	1.0600	0
2	2	1.0414	-3.1767
3	2	1.0040	-9.6529
4	2	1.0316	-6.4722
5	2	1.0328	-5.6002
6	0	1.0837	-10.0089
7	0	1.0703	-8.7011

8	0	1.0703	-8.7011
9	0	1.0791	-9.8470
10	0	1.0728	-10.1531
11	0	1.0750	-10.2066
12	0	1.0185	-12.7197
13	0	1.0517	-11.7033
14	0	1.03308	-12.4159

Total System Loss = $0.12665 + j 0.39886$ p.u.

Line flows

<i>From</i>	<i>bus</i>	<i>Line flow (pu)</i>	<i>Line flow (pu)</i>
1	2	$1.0406 - 0.0090i$	$-1.0220 + 0.0077i$
1	5	$0.4874 + 0.0068i$	$-0.4759 - 0.0133i$
2	3	$0.6158 + 0.0609i$	$-0.5990 - 0.0362i$
2	4	$0.3361 - 0.0613i$	$-0.3300 + 0.0435i$
2	5	$0.2531 - 0.0447i$	$-0.2497 + 0.0179i$
3	4	$-0.3433 - 0.0251i$	$0.3512 + 0.0319i$
4	5	$-0.3575 + 0.0869i$	$0.3592 - 0.0815i$
4	7	$0.2054 - 0.1870i$	$-0.2054 + 0.2021i$
4	9	$0.1178 - 0.0846i$	$-0.1178 + 0.0956i$
5	6	$0.3414 - 0.1954i$	$-0.3414 + 0.2320i$
6	11	$0.0349 + 0.0307i$	$-0.0347 - 0.0304i$
6	12	$0.2755 + 0.1488i$	$-0.2653 - 0.1274i$
6	13	$0.3148 + 0.1104i$	$-0.3085 - 0.0981i$
7	8	$-0.0000 + 0.0000i$	$0.0000 - 0.0000i$
7	9	$0.2100 - 0.0834i$	$-0.2100 + 0.0883i$
9	10	$0.0907 + 0.0464i$	$-0.0904 - 0.0456i$
9	14	$0.2114 + 0.0532i$	$-0.2062 - 0.0421i$

10	11	0.0003 - 0.0124i	-0.0003 + 0.0124i
12	13	-0.1265 - 0.0285i	0.1301 + 0.0317i
13	14	0.0433 + 0.0084i	-0.0430 - 0.0078i

From the above results it can be concluded that the voltages improve while the overall system losses reduce along with control of line power by placing a UPFC in the system.

6.5 FORMULATION OF STATE ESTIMATION PROBLEM FOR POWER SYSTEM EMBEDDED WITH UPFC.

The basic problem remains the same as discussed in previous chapter except for additional state variables and constraints which must be included in the formulation to reflect presence of UPFC in the power system. The Jacobian matrix of the system should also be modified.

As discussed in section 6.3, control parameters of UPFC namely the magnitude and the angle of inserted voltage (V_{s1} , φ_{s1}) in line- k and the magnitude of the current (I_q), are included in the state vector set. Objective again is to minimize the WLS error between measured and calculated data, subject to additional UPFC constraints, defined below:

Equality constraints

Neglecting UPFC losses, during steady state operation UPFC neither absorbs nor injects real power into the system [26]. Thus the sum of real powers injected to the end buses of the line where UPFC is inserted should equal zero.

From Eqns. (6.14) and (6.16)

$$P_{ig} + P_{jg} = 0$$

Inequality Constraints:

UPFC control parameter limits: The voltage magnitude (V_T) and phase angle (ϕ_T) of series voltage of UPFC must lie within limit. Mathematically, it can be written as,

$$0.5 \geq \bar{V}_{s1max} \geq 0$$

$$\bar{I}_{q\max} \geq 0$$

$$0 \leq \bar{V}_{s1} \leq \bar{V}_{s1\max}$$

$$\bar{I}_{q\min} \leq \bar{I}_q \leq \bar{I}_{q\max}$$

$$0 \leq \phi_{s1} \leq 2\pi$$

It is to be noted here that all the equality and inequality constraints defined in chapter 5 are applicable together with above constraints.

6.5.1 ESTIMATION ALGORITHM

The steps followed have been given as under:

Step 1: Get the system data, measurements and define the zero injection buses together with boundary limits on the state variables.

Step 2: Select an initial erroneous state vector, tolerance limit and set the iteration count.

Step 3: Calculate the objective function and say it $f(v)$ old.

Step 4: Calculate P_i , Q_i , and $P_{ig}+P_{jg}$ corresponding to equality constraints.

Step 5: Find $\nabla(h(v))$ by differentiating zero injection Eqns with respect to state variables using load flow equations.

Step 6: Calculate updated state variables by Eqn. (5.28).

Step 7: Enforce the boundary limits by passing the state variables through a symmetrical ramp activation function

Step 8: Find i^{op} by differentiating the objective function with respect to state variables.

Step 9: Find Δv by Eqn. (5.19) and Update v .

Step 10: Find the mismatch vector between measurements and calculated values and get its weighted squared sum to find out the new objective function value and find the difference between $f(v)$ new and $f(v)$ old. If this difference is less than tolerance go next step, else go to step 3 after increasing the iteration count.

Step 11: Display the results and Stop.

6.6: RESULTS AND DISCUSSIONS

The IEEE 14 bus system is used for simulation. The true values were obtained by

the result of the load flow calculation, and the measurement values were made by adding errors to those true values. The measurement set base value for the IEEE 14 bus system is shown in Fig.6.9 and table 6.2. Bus no 5 and 7 are characterized as zero injection buses and UPFC is inserted between bus no. 6 and bus no.12. The energy mismatch delta E was used for the convergence criteria with the tolerance 10^{-02} and the time step used was $\Delta t=10^{-04}$.

Table:6.2

Measurements	Type	Buses	P(pu)	Q(pu)
z_1	Injection	1	1.52	-0.058
z_2	Injection	2	0.18	-0.250
z_3	Injection	3	-0.9522	-0.16
z_4	Injection	4	-0.1583	-0.0039
z_5	Injection	6	-0.1954	-0.234
z_6	Injection	8	0.1496	0.0017
z_7	Injection	9	0.0883	0.0210
z_8	Injection	10	-0.0057	-0.048
z_9	Injection	11	-0.0963	-0.1603
z_{10}	Injection	12	-0.5602	0.1396
z_{11}	Injection	13	-0.1978	-0.058
z_{12}	Injection	14	-0.1173	-0.0508
z_{13}	Line flow	1-2	1.0434	-0.0090
z_{14}	Line flow	1-5	0.4886	0.0068
z_{15}	Line flow	2-3	0.6171	0.0609
z_{16}	Line flow	2-4	0.3372	-0.0613
z_{17}	Line flow	2-5	0.2537	-0.0447
z_{18}	Line flow	3-4	-0.3436	-0.0251
z_{19}	Line flow	4-5	-0.3594	0.0869
z_{20}	Line flow	4-7	0.2105	-0.1870
z_{21}	Line flow	4-9	0.1218	-0.0846

z_{22}	Line flow	5-6	0.3675	-0.1954
z_{23}	Line flow	6-11	0.0347	0.0307
z_{24}	Line flow	6-12	0.2758	0.1488
z_{25}	Line flow	6-13	0.3149	0.1104
z_{26}	Line flow	7-8	0.0000	0.0000
z_{27}	Line flow	7-9	0.2100	-0.0834
z_{28}	Line flow	9-10	0.0914	0.0464
z_{29}	Line flow	9-14	0.2118	0.0532
z_{30}	Line flow	10-11	0.0001	-0.0124
z_{31}	Line flow	12-13	-0.1267	-0.0285
z_{32}	Line flow	13-14	0.0434	0.0084

The estimated states are shown below:

Bus No.	V	δ	Bus No.	V	δ
1	1.06	0	2	1.043	-3.731
3	1.002	-9.309	4	1.032	-6.615
5	1.034	-5.046	6	1.085	-10.68
7	1.071	-8.08	8	1.07	-8.922
9	1.079	-9.481	10	1.073	-10.553
11	1.072	-10.167	12	1.019	-12.296
13	1.051	-11.443	14	1.043	-12.258

The convergence characteristics of the energy function with respect to number of iterations is shown in Fig.6.10.

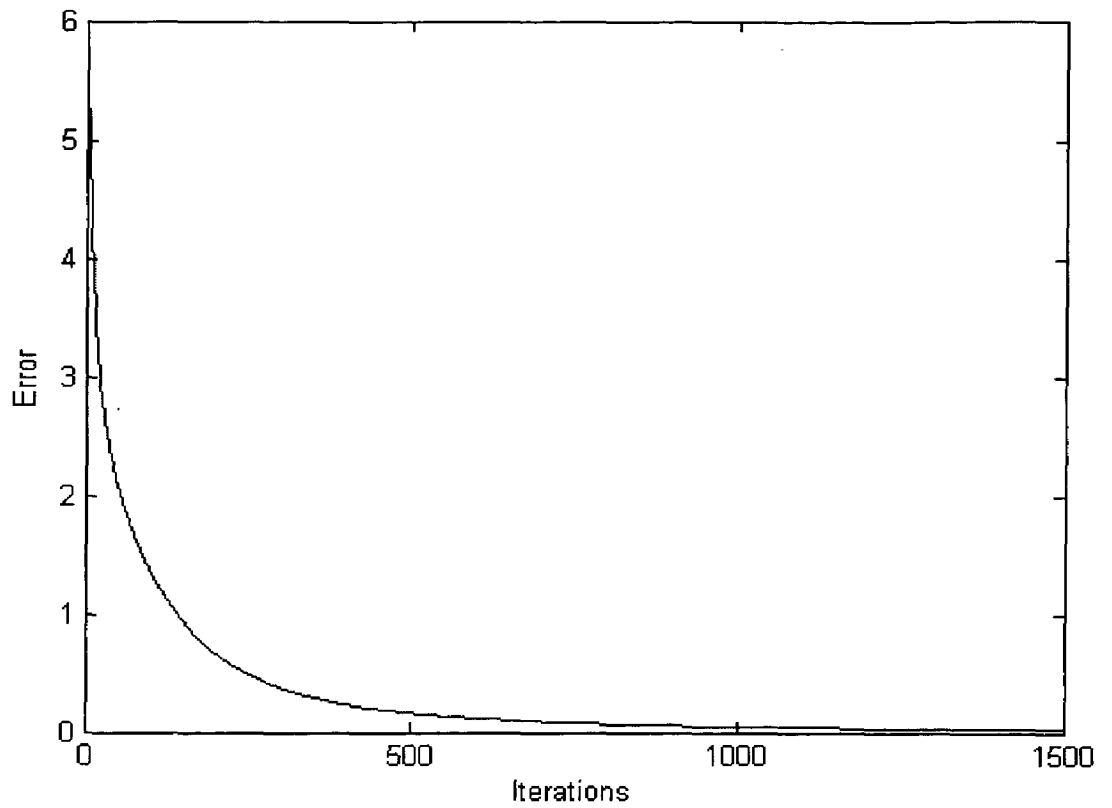


Figure 6.10: Convergence of energy function

This part of the dissertation presents an algorithm for state estimation of power systems embedded with FACTS devices. While only the Unified Power Flow Controller (UPFC) is used in the development, other types of controllers can easily be integrated into the developed prototype with minor effort. This algorithm also estimates the controller parameters along with system state during normal operation.

CHAPTER 7

IMPLEMENTATION OF MODIFIED HOPFIELD NEURAL NETWORK ON NEURAL NETWORK PROCESSOR FOR SOLVING NONLINEAR PROGRAMMING PROBLEM

The modified Hopfield neural network algorithm discussed in previous chapter for solving nonlinear programming problems is implemented on a dedicated neural network processor, emphasizing the specific architecture and programming paradigm. For the demonstration purpose a simple nonlinear programming problem is solved by the network. The internal parameters of the network are obtained using the valid-subspace technique. The implementation is aimed for rapid real time neural network applications. The concept has been tested on the AAC (Accurate Automation Corporation) Multiple Instruction Multiple Data (MIMD) Neural Network Processor (NNP) hardware.

7.1 INTRODUCTION

Constrained optimization problems have a fundamental role in many areas of sciences and engineering, where a set of design parameter is optimized subject to inequality and/or equality constraints. Basically, all of the neural networks [58, 59] used in constrained optimization contain some penalty parameters. The stable equilibrium points of these networks, corresponding to solution of the optimization problem, are obtained only when the penalty parameters are sufficiently large [60]; while the modified Hopfield network [25] does not depend on the penalty parameter. The constraints of the problem are not included in the network energy function rather they are handled by valid subspace technique.

Software implementations on general-purpose computers can be too sluggish for time-critical applications. Here, dedicated ANN hardware and the ANN programming paradigm are described as an effective compromise of performance and cost, enabling a computational platform for time-critical neural network processing tasks. The majority of ANN implementations today tend to be software simulators hosted on general-purpose computers. Some special neural network platforms were researched and published [54, 58

and 59].

Accurate Automation Corporation's Neural Network Tools (NNPTools) allows rapid prototyping of neural network solutions and integration into real world applications. Each processor runs at 140 million connections/sec with 8K neurons. An expanded version of the system performs a total of a billion plus connections/sec. Unlike classical SIMD NN architectures, which are really general purpose array processors, this MIMD system architecture was custom designed for NN applications.

In this chapter a new implementation paradigm for solving nonlinear programming using Hopfield neural network is presented which is particularly suited for real time applications.

7.2: THE MODIFIED HOPFIELD NETWORK

The algorithm of the network is described in section (5.3) of chapter (5). A brief description is reproduced here for reference:

Step (i): Minimization of E^{conf} (constraint satisfactions) Corresponding, to the projection of $v(t)$ in the valid subspace defined by:

$$v(t) = T^{\text{conf}} \cdot v(t) + i^{\text{conf}} \quad (7.1)$$

This operation corresponds to an indirect minimization of $E^{\text{conf}}(t)$.

Step (ii): Application of a nonlinear 'symmetric ramp' activation function constraining $v(t)$ in a hypercube

Step (iii): Minimization of E^{op} , which consists of updating $v(t)$ in the opposite direction to the gradient of E^{op} (defined by T^{op} and i^{op}) corresponding to network equilibrium points, which are the solutions for the constrained optimization problems.

Each iteration has two distinct stages. First, as described in Step (iii) v is updated using the gradient of the term E^{op} alone. Second, after each updating, v is directly projected in the valid subspace. In the next section, the parameters T^{conf} , i^{conf} , T^{op} and i^{op} are defined.

7.3: FORMULATION OF THE NONLINEAR OPTIMIZATION PROBLEM

Consider the following general nonlinear optimization problem:

$$\text{Minimize } E^{\text{op}}(v) = f(v) \quad (7.2)$$

$$\text{Subject to } E^{conf}(v): h_i(v) \leq 0, \quad i \in (1..p) \quad (7.3)$$

$$v^{min} \leq v \leq v^{max}$$

Where $v, v^{min}, v^{max} \in \mathbb{R}^n$;

A projection matrix to the system is given as follows [25]:

$$T^{conf} = [I - \nabla(h(v))^T \cdot (\nabla(h(v)) \cdot \nabla(h(v))^T)^{-1} \cdot \nabla(h(v))] \cdot v + s \quad (7.4)$$

where

$$\nabla h(x) = \begin{bmatrix} \frac{\partial h_1(x)}{\partial x_1} & \frac{\partial h_1(x)}{\partial x_2} & \dots & \frac{\partial h_1(x)}{\partial x_N} \\ \frac{\partial h_2(x)}{\partial x_1} & \frac{\partial h_2(x)}{\partial x_2} & \dots & \frac{\partial h_2(x)}{\partial x_N} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial h_p(x)}{\partial x_1} & \frac{\partial h_p(x)}{\partial x_2} & \dots & \frac{\partial h_p(x)}{\partial x_N} \end{bmatrix} \quad (7.5)$$

Inserting the value of Eqn. (6.4) in the expression of the valid subspace in Eqn. (7.1):

$$v = v - \nabla h(v)^T \cdot ((\nabla h(v)) \cdot \nabla h(v)^T)^{-1} \cdot h(v) \quad (7.6)$$

The equilibrium points of the network can be calculated by assuming the following values of T^{op} and i^{op} ,

$$i^{op} = \left[\frac{\partial f(v)}{\partial v_1}, \frac{\partial f(v)}{\partial v_2}, \dots, \frac{\partial f(v)}{\partial v_N} \right] \quad (7.7)$$

$$T^{op} = 0 \quad (7.8)$$

7.4: COMPUTATIONAL PARADIGM

ANNs represent a computational paradigm; whose characteristics include intrinsic parallelism, local processing in (artificial) neurons, distributed memory, and learning and recall modes.

An interesting aspect of the ANN computational paradigm is to represent complex problems with a small number of primitives which suggests advantages of hosting ANNs on dedicated Neural Network Hardware (NNH) to maximize performance at a given cost target. The number of weights grows exponentially with the number of neurons. The intrinsic parallelism of ANNs facilitates parallel processing as a key feature of NNH, i.e.

the decomposition of neural processing into concurrently executed sub processes. Parallel processing exploits concurrency in a computational process and comprises both parallel hardware and parallel software.

7.4.1 Programming Paradigm and Language

The way we describe computations for computers is known as programming paradigm. It is a technique which supports writing code for a specific set of problems, using a programming language that is designed to describe that specific set of problems. A good language also includes standard libraries and programming tools like editor and compiler.

An ANN language should therefore resemble the ANN paradigm, i.e. should match its description and definition. The ANN paradigm therefore provides the opportunity to design a powerful language that exploits the capabilities of the NNH for good run-time performance. The mere availability of parallel hardware itself does not guarantee parallel processing. This directly corresponds to the sophistication of parallel software exploiting the parallel architecture in NNH. Parallel processing at the highest level is carried out by multiple programs, while at a medium level it is limited to concurrent tasks within a single program. The lowest level refers to concurrence of multiple instructions, or even concurrence within an instruction [61].

7.5 NEURAL NETWORK PROCESSOR NNP®

The above concepts are illustrated with the commercially available Neural Network Processor (NNP) by Accurate Automation Corp. [56]. Its design concept is reviewed, and the implementation of the modified hopfield neural network is proposed.

7.5.1 Architecture: The NNP® architecture provides features for fast processing of neural connections and transfer function. Up to eight NNPs can be integrated in a parallel multiprocessor environment, resulting in a multiple instruction stream/multiple data stream (MIMD) platform. The block diagram of a single NNP® is shown in Fig. 7.1.

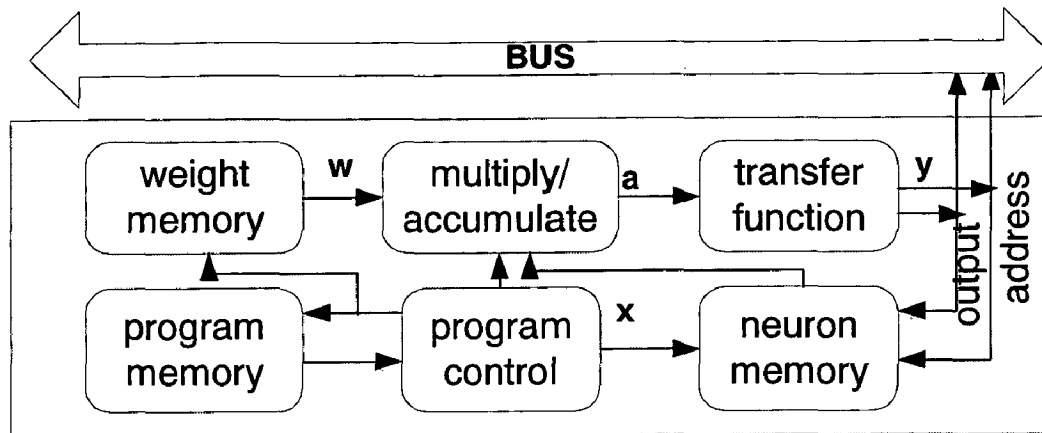


Figure 7.1: Block diagram of a single NNP (10)

A multiply-accumulate unit, lookup tables for transfer functions, and 16-bit integer arithmetic emphasize the above feature. The figure illustrates dataflow of activation values a , neuron input x and weight value w . The neuron output y , i.e. the transfer function, is not computed explicitly but is read from a pre-loaded lookup table. Weights are usually learned off-line, but NNP can also be utilized for on-chip learning. Efficiency is further increased by instruction pipelining, enabling the completion of one instruction per clock cycle. Thus the NNP is single instruction stream/multiple data stream (SIMD) processor. The AAC MIMD Neural Network Processor is designed to implement multiple interconnected neural networks of differing architecture simultaneously using 16-bit twos-complement binary fixed-point arithmetic with up to 8k total neurons and 32k connection weights per module, It is capable of running at 140,000,000 connections (byte wide multiply/additions) per second per module for a total of one billion plus connections per second in an 8 processor array, It Supports two I/O buses, an Interprocessor Bus which can also be used for on-line I/O in parallel with the computational process, and a memory I/O Bus through which the various processor memories may be mapped into the memory space of a supporting microprocessor or DSP for downloading programs, connection weights, etc, and Each processor in an NNP array is controlled by an instruction set, supported by an NNP Module Simulator, an Assembler, a Neural Network Compiler, and the Accurate Automation Neural Network Toolbox [56].

7.4.2.2 Programming Paradigm

The NNP programming paradigm emphasizes layered, well-structured, ANN architectures. Accordingly, its programming language comprises only nine powerful commands, tailored for the efficient computations of appropriate ANN primitives, in particular connections and nonlinear transfer function. For example ‘mula’ (multiply and accumulate), mull (multiply and load in accumulator) and ‘lbt’ (compute transfer function). The cornerstone of an NNP assembler program is the multiply-accumulate command, which multiplies one weight value with one input value and adds the product to the existing activation value.

7.6: EXAMPLE

The modified Hopfield network have been used to solve the constrained nonlinear optimization problem defined by

$$\text{Min } f(v) = .4v_1 + .5(5v_1^2 + 8v_2^2 + 4v_3^2) - 3v_1v_2 - 3v_2v_3 \quad (7.9)$$

$$\text{Subject to: } v_1 + v_2 + v_3 = 1 \quad (7.10)$$

$$0 \leq v_1 \leq .8 \quad (c1)$$

$$0 \leq v_2 \leq .4 \quad (c2)$$

$$0 \leq v_3 \leq .5 \quad (c3)$$

For this problem, with one equality constraints and with bounded variables, the solution vector (equilibrium point) obtained by the modified Hopfield network on NNPTools is given by :

$v^* = [0.2568; 0.3444; 0.3891]$, with $f(v^*) = 0.3776$. These results are near to optimal solution provided by $v^* = [0.2576; 0.3332; 0.4092]$ and $f(v^*) = 0.3683$ on Matlab. The initial value of v was randomly generated between 0 and 1. The small difference in the two values is due to truncation in case of NNP implementation because 16 bit registers are used to store the data in binary form. The algorithm on the NNP was written in assembler with a constant initial weight matrix

$$T_{\text{conf}} = \begin{bmatrix} .6667 & -.3333 & -.3333 \\ -.3333 & .6667 & -.3333 \\ -.3333 & -.3333 & .6667 \end{bmatrix}$$

The following figure shows the input and output neuron values when the program is executed on NNPTools for 20 iterations. A complex problem may require that the weight

matrix should be updated for each iteration as per Eqn. (7.4) and hence an interfacing is required between the host computer and the NNP.

Offset	Hex	Value
inputt	07C0	0.484375
0001	0440	0.265625
0002	0360	0.210938
outputt	041C	0.256838
0007	0583	0.344482
0008	063A	0.389160

Figure 7.2: Neuron Memory from NNPTools

The host updates weights and presents data to the NNP. The weight updation algorithm together with some complex calculation can be written in C++ for the host computer. In that case the NNP takes care of neuron updation in the forward path..

A small portion of the assembler code for updating a neuron is shown below:

```

; neuron 0
    mull inputt+0,0.666700
    muala inputt+1,-0.333300
    muala inputt+2,-0.333300
    muala inputt+3,.333333
    lbtft outputt+0,tsf

```

The hardware implementation of the computing engine is suitable for online applications with a fast speed.. By using an MIMD parallel processing architecture one can update multiple neurons in parallel with efficiency approaching 100% as the size of the neural network increases, to achieve the desired efficiency of real-world applications. The inherent advantage of the proposed scheme is its speed, which is particularly suited for real time implementations applications.

CHAPTER-8

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

The state estimator plays an important role of a purifier, creating a complete and reliable database for security monitoring, security analysis and the various controls of a power system. State estimator should estimate the system states as quickly as possible, but conventional computer based methods are almost reaching a limit in terms of speed. On the other hand neural networks, having much potential for hardware implementation, along with their inherent parallel architecture are being used in various areas of science and engineering. This work proposes a neural network method for solving state estimation problem which can be implemented on hardware. The common weighted least square method does not enforce the equality and limit constraints explicitly. However, the constraints contain reliable information about physical restrictions and equipment limits and can be used to increase the quality of state estimation result. Further an increasing concern about environmental aspect and optimal use of transmission capacities emphasize the use of FACTS devices, which offer several advantages, in the system, so there is requirement of such estimators which not only estimate the voltage magnitude and phase angle but also FACTS device control parameters. In this work FACTS devices are also included in the state estimation algorithm. However, the work has been carried out only for the static system conditions it can be extended to dynamic system also. In conclusion, the findings of the work have been briefed as follows:

1. The topological observability problem is formulated as an integer programming problem, and Hopfield neural network is used with simulated annealing to judge whether network is observable or not, further the algorithm also provide information about where the meters should be placed in order to get an observable system when number of meters and buses are known in advance. The inequality constraints of the topological observability problem are not included in the objective function by using penalty factors which is a difficult task. They are handled in a different way by assigning a dedicated neural network in the original

Hopfield model.

2. The state estimation problem is formulated as a general nonlinear programming problem with equality constraints and limits on variables, and modified Hopfield neural network method is used for estimating the system state. The software implementation of the algorithm is slower than the conventional methods but when it will be implemented on hardware using any reconfigurable technology such as FPGAs, a faster speed is assured. One of the major advantages is that no training and testing of the neural network is required under human supervision.
3. UPFC have been incorporated in the system with the help of Power Injections Models (PIM). UPFC has been incorporated into an existing Newton-Raphson load flow algorithm. This approach keeps the conventional NRLF intact so that the UPFC is readily being incorporated into traditional NRLF by including the modified terms in the iterative procedure. For each UPFC just modification of few elements of the Jacobian matrix and hence additional burden incurred is minimized. The other FACTS devices can also be modeled in the same manner and incorporated in load flow.
4. State estimation with the modified Hopfield neural network method is carried out in presence of UPFC, along with the additional constraints which arises thereafter.
5. The modified Hopfield neural network algorithm is implemented on dedicated neural network processor (NNP) architecture, to solve a simple nonlinear programming problem to demonstrate the feasibility of hardware implementation of the algorithm.

SUGGESTIONS FOR FUTURE WORK

As development is an unending process. End of one work opens doors for further work. As a consequence of the investigations carried out in this thesis on state estimation of power systems, following aspects are identified for future research work in this area.

1. Topology processing and bad data analysis is not carried out in this work and hence the algorithm can be modified to include these concepts.
2. The Hopfield neural network algorithm for topological observability determination can be implemented on any digital reconfigurable technology.
3. The state estimation algorithm, discussed in chapter (5), which is suitable for

hardware implementation, can be implemented on dedicated processors or hardware platforms to validate its applicability in real time state estimation.

4. Present study has considered the static condition of FACTS devices from steady state operation point of view. Inclusion of dynamic state of these devices can also be explored. Instead of UPCF other FACTS devices can also be incorporated in the state estimation algorithm.

REFERENCES

- [1] Wu F.F., Liu W.H.E., Holtan L., Gjelsvik A., Aam S., **“Comparison of Different Methods for State Estimation,”** IEEE Transaction On Power Systems, Vol. 3, No. 4, November 1988, pp 1798-1806.
- [2] F.C. Schweppe, J. Wildes, D.B. Rom, **“Power System Static-State Estimation, Parts I, II and III,”** IEEE Transactions on Power Apparatus and Systems, Vol. PAS-89, No. 1, January 1970, pp. 120-135.
- [3] A. Garcia, A. Monticelli, and P. Abreu, **“Fast Decoupled State Estimation and Bad Data Processing,”** IEEE Transaction Power Apparatus and systems. Vol. PAS-98, pp. 1645–1652, Oct. 1979.
- [4] A. Simoes Costa, and V. H. Quintana, **“A Robust Numerical Technique for Power System State Estimation,”** IEEE Transaction Power Apparatus and systems. Vol. PAS-100, pp. 691–698, Feb. 1981.
- [5] F. C. Aschmoneit, N. M. Peterson, and E. C. Adrian, **“An Orthogonal Row Processing Algorithm for Power System Sequential State Estimation,”** IEEE Transaction Power Apparatus and systems. Vol. PAS-100, pp. 3791–3800, Aug. 1981.
- [6] J. W. Wang and V. H. Quintana, **“A Decoupled Orthogonal Row Processing Algorithm for Power System State Estimation,”** IEEE Transaction Power Apparatus and systems. Vol. PAS-103, pp. 2337–2344, Aug. 1984.
- [7] A. Monticelli, C. A. F. Murati, and F. F. Wu, **“A Hybrid State Estimator: Solving Normal Equations by Orthogonal Transformations,”** IEEE Transaction Power Apparatus and systems. Vol. PAS-105, pp. 3460–3468, Dec. 1985.
- [8] F. C. Aschmoneit, N. M. Peterson, and E. C. Adrian, **“State Estimation with Equality Constraints,”** in Proc. 10th PICA Conf., Toronto, Canada, May 1977, pp. 427–430.
- [9] A. Simoes-Costa, S. Seleme, and R. Salgado, **“Equality Constraints in Power System State Estimation via Orthogonal Row-Processing Techniques,”** in Proc. IFAC Conf. Electrical Energy Systems., Rio de Janeiro, Brazil, 1985, pp. 43–49.

- [10] A. Gjelsvik, S. Aam, and L. Holten, "**Hachtel's Augmented Matrix method—A rapid method for Improving Numerical Stability in Power System Static State Estimation,**" IEEE Transaction Power Apparatus and systems. Vol. PAS-104, pp. 2987–2993, Nov. 1985.
- [11] W. E. Liu, F. F. Wu, L. Holten, A. Gjelsvik, and S. Aam, "**Computational issues in the Hachtel's Augmented Matrix method for Power System State Estimation,**" in Proc. Power System Computation Conf., Lisbon, Portugal, 1987.
- [12] F. Alvarado and W. Tinney, "**State Estimation Using Augmented Blocked Matrices,**" IEEE Transaction Power Systems, vol. 5, pp. 911–921, Aug. 1990.
- [13] R. R. Nucera and M. L. Gilles, "**A Blocked Sparse Matrix Formulation for the solution of Equality-Constrained State Estimation,**" IEEE Transaction Power Syst., vol. 6, pp. 214–224, Feb. 1991.
- [14] E. Kliokys and N. Singh, "**Minimum Correction Method for Enforcing limits and Equality Constraints in State Estimation Based on Orthogonal Transformations,**" IEEE Transaction Power Systems., vol. 15, pp. 1281–1286, Nov. 2000.
- [15] A. Clements and B.F. Wollenberg. "**An Algorithm for Observability Determination in Power System State Estimation,**" IEEE PES Summer Meeting, Paper A 75 447-3, San Francisco, July 1975.
- [16] V.H. Quintana. A. Shoes-Costa, and A. Mandcl, "**Power System Observability Using a Direct Graph-Theoretic Approach,**" IEEE Transaction Power App. and Systems, Vol. 101. No. 3, pp. 617-626, March 1982.
- [17] Monticelli and F.F. Wu, "**Network Observability: Identification of Observable Islands and Measurement Placement,**" IEEE Transaction on Power Apparatus and Systems. Vol. Pas-104, No. 5, pp. 1035- 1041, May 1985.
- [18] A.Bargiela, et al., "**Observability Determination in Power System State Estimation,**" IEEE Transaction On Power Syst., Vol. PWRS-1, No. 2, p. 108-114, May 1986.
- [19] H. Mori and S. Tsuzuki, "**Power System Topological Observability Analysis Using a Neural Network Model**", Proc. of Second Symposium on Expert Systems Application to Power Systems", pp. 385-391, Seattle, Washington, U.S.A., July 1989.

- [20] H. Mori and S. Tsuzuki, **“Determination of Power System Topological Observability Using the Boltzmann Machine”**, Proc. of IEEE 1990 ISCAS, pp. 2938-2941, New Orleans, LA, U.S.A., May 1990.
- [21] Tank, D.; Hopfield, J., **“Simple ‘Neural’ Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit”** IEEE Transaction on Circuits and systems, Volume: 33, Issue: 5, May 1986 pp. 533 -541.
- [22] G. V. Wilson and G. S. Pawley, **“On the Stability of the TSP Algorithm of Hopfield and Tank”**, Biological Cybernetics, vol. 58, pp. 63-70, 1988.
- [23] B. Kamgar-Parsi, **“Dynamical Stability and Parameter Selection in Neural Optimization”**, Proc. International Joint Conference on Neural Networks, vol. 4, pp. 566-571, 1992.
- [24] S. V. B. Aiyer, M. Niranjan and F. Fallside, **“A Theoretical Investigation into the Performance of the Hopfield Model”**, IEEE Transactions on Neural Networks, vol. 1, no. 2, pp. 204-215, May 1990.
- [25] I.N. Silva, and L.V.R. de Arruda, **“Nonlinear Optimization using a Modified Neural Network”** Proc. International Joint Conference on Neural Networks, IJCNN Vol 2, pp. 1629 - 1633 , May 1998 .
- [26] N.G. Hingorani and L.Gyugyi, **“Understanding FACTS,”** The Institution of Electrical and Electronics Engineers, 1998.
- [27] R. Mohan Mathur, R.K.Varma, **“Thyristor-Based FACTS Controllers for Electrical Transmission Systems,”** A John Wiley & Sons. Inc. Publication, 2002.
- [28] Y. H. Song, J.Y. Liu and P. A. Mehta, **“Power Injection Modeling and Optimal Multiplier Power Flow Algorithm for Steady State Studies of Unified Power Flow Controllers,”** Electrical Power System Research, Vol 52, pp. 51-59, 1999.
- [29] A.Edris, **“FACTS Technology Development: An update,”** IEEE Power Engineering Review, March 2000, pp 4-11.
- [30] L.Gyugyi, **“A Unified Power Flow Control Concept for Flexible AC Transmission Systems”**, IEE Proc., Part-C, Vol.139, No.4, July 1992, pp.323-333
- [31] L. Gyugyi, T.R. Rietman, A. Edris, C.D. Schauder, D.R. Torgerson, S.L.Williams, **“The Unified power Flow Controller: A new Approach To power Transmission Control,”** IEEE transactions on power Delivery , vol. 10,No.2, pp.1085 -1097, April

1995.

- [32] L.L. Freris and A.M. Sasson, "**Investigation of the Load Flow Problem**," Proc.IEE Part C, Vol. 115, 1968, pp. 1459-1470.
- [33] Y. H. Song, J.Y. Liu and P. A. Mehta, "**Power Injection Modeling and Optimal Multiplier Power Flow Algorithm for Steady State Studies of Unified Power Flow Controllers**," Electrical Power System Research, Vol 52, pp. 51-59, 1999.
- [34] Ali Abur, Garng M. Huang "**Power System State Estimation and Optimal Measurement Placement for Distributed Multi-Utility Operation**" Power Systems Engineering Research Center ,PSERC Publication 02-45 November 2002
- [35] T.E. Dy Liacco, "**An Overview of Power System Control Centers**", Energy Control Center Design, IEEE Tutorial Course, TU0010-9PWR, 1977.
- [36] M. Ribbens-Pavella, T. Van Cutsem, P. Rousseaux, "**On-Line Stability and Dynamic Security Assessment of Electric Power Systems**", Proc. Of the IFAC Symposium on Planning and Operation of Electric Energy Systems, Rio de Janeiro, Brazil, July 22- 25, 1985.
- [37] P. Rousseaux, Th. Van Cutsem, T.E. Dy Liacco, "**Whither Dynamic State Estimation?**", Electrical Power & Energy Systems, Vol. 12, No. 2, April 1990, pp. 104-116.
- [38] A. Garcia, A. Monticelli, and P. Abreu, "**Fast Decoupled State Estimation and Bad Data Processing**," IEEE Transaction Power App. and Syst., Vol. PAS-98, pp. 1645-1652, September 1979.
- [39] Smith, K. "**Neural Networks for Combinatorial Optimization: A review of more than a decade of research**", INFORMS Journal on Computing, vol. 11, no. 1, pp. 15-34, 1999.
- [40] M. Verleysen and P. Jaspers, "**An Analog VLSI Implementation of Hopfield's Neural Network**", IEEE Micro, pp. 4
- [41] B. Sheu, E. Chou, R. Tsai and D. Chen, "**VLSI Neural Networks: Design Challenges and Opportunities**", in Computational Intelligence, M. Palaniswami, Y. Attikiouzel, R. J. Marks II, D. Fogel and T. Fukada (eds.), IEEE Press, NY, 1995.
- [42] J. J. Hopfield, "**Neural Networks and Physical Systems with Emergent Collective Computational Abilities**", Proceedings National Academy of Sciences,

vol. 79, pp. 2554-2558, 1982.

- [43] K. A. Clements and B.F. Wollenberg. **“An Algorithm for Observability Determination in Power System State Estimation,”** IEEE PES Summer Meeting, Paper A 75 447-3, San Francisco, July 1975.
- [44] G. R. Krumpholtz, et al., **“Power System Observability: A Practical Algorithm Using Network Topology,”** IEEE Transaction On Power Apparatus and Systems., Vol. PAS-99, No. 4, pp.1534-1542, July1980..
- [45] V. H. Quintana. A. Shoes-Costa, and A. Mandcl, **“Power System Observability Using a Direct Graph-Theoretic Approach,”** IEEE Transaction Power Apparatus and systems. Vol. 101. No. 3, pp. 617-626, March 1982.
- [46] A. Monticelli and F.F. Wu, **“Network Observability: Identification of Observable Islands and Measurement Placement,”** IEEE Transaction on Power Apparatus and systems, Vol. PAS-104, No. 5, pp. 1035- 1041, May 1985.
- [47] A. Bargiela, et al., **“Observability Determination in Power System State Estimation,”** IEEE Transaction On Power System., Vol. PWRS-1, No. 2, pp. 108-114, May 1986.
- [48] H. Mori and S. Tsuzuki, **“Power System Topological Observability Analysis Using a Neural Network Model”**, Proc. of Second Symposium on Expert Systems Application to Power Systems”, pp. 385-391, Seattle, Washington, U.S.A., July 1989.
- [49] H. Mori and S. Tsuzuki, **“Determination of Power System Topological Observability Using the Boltzmann Machine”**, IEEE International Symposium on Circuits and Systems, 1990, Vol.4 pp. 2938 - 2941
- [50] M. B Do Coutto, Filho A. M., Leite Da Dilva ,**“Bibliography on Power System State Estimation (1968 - 1989)”** IEEE Transactions on Power Systems, Vol. 5, no. 3, August 1990.
- [51] David Abramson, Kate Smith, David Duke, **“FPGA Based Implementation of a Hopfield Neural Network for Solving Constraint Satisfaction Problems”** Euromicro Conference, 1998. Proceedings-24., Volume: 2, 25-27 Aug. 1998 pp. 688 - 693.
- [52] Botros, N. M.; Abdul-Aziz, M.; **“Hardware Implementation of an Artificial Neural Network using Field Programmable Gate Arrays (FPGA's)”** IEEE

- Transactions on Industrial Electronics, Vol: 41, Issue: 6, Dec. 1994 pp.665 – 667
- [53] Bazaraa, Shetty, & Sherali, **Nonlinear Programming, Theory & Applications**, Wiley, 1994.
- [54] Silva, I. N, Amaral W. A., Arruda, V. R.; “**Robust Parametric Estimation for Nonlinear Models using Artificial Neural Networks**” IEEE International Conference on Systems, Man, and Cybernetics, Vol 4, pp: 3813 - 3816
- [55] Allen J. Wood and Bruce F. Wollenberg ,”**Power Generation, Operation and Control 2/e**” John Wiley & Sons, Inc., 1996.
- [56] Accurate Automation, “**Neural Network Tool Manual**”, Accurate Automation Corp., 7001 Shallowford Rd., Chattanooga, TN 37421, 1995.
- [57] N.G. Hingorani and L. Gyugyi,”**Understanding FACTS**”, The Institution of Electrical and Electronics Engineers, 1998.
- [58] M. P. Kennedy and L. O. Chua, “**Neural Networks for Nonlinear Programming**,” IEEE Transaction Circuits Systems. vol. 35, no 3. pp 349-358, 1988
- [59] A. Rodriguez-Vazquez et al.,”**Nonlinear Switched Capacitor Neural Network for Optimization Problems**,” IEEE Transaction Circuits Syst., vol. 37, pp. 384-398, March 1990.
- [60] X. Y. Wu et al., “**A High Performance Neural Network for Solving Linear and Quadratic Programming Problems**,” IEEE Transaction Neural Networks, vol. 7, no.3, May 1996.
- [61] Hennesy J.L. and D.A. Patterson, “**Computer Organization and Design: The Hardware/Software Interface**”, Morgan Kaufman, San Mateo, California,
- [63] www.accurate-automation.com

APPENDIX –A

Data for six bus system (at 100 MVA base)

The six bus system is shown on Fig. A1. The relevant data are provided in the following tables.

Table A1

Generator Power Data

Bus No.	Real Power Generation		Reactive Power Limits		Specified voltage(pu)
	Max(MW)	Min(MW)	Max(MVAR)	Min(MVAR)	
2	100	20	50	-40.0	1.050

Table A2

Load bus data

Bus No.	Load	
	Real(MW)	Reactive(MVAR)
5	70	70
6	70	70

Table A3

Line data

Line No.	From Bus	To Bus	Series Impedance(p.u.)		Half Line Charging(pu)
			Resistance	Reactance	
1	1	2	0.1000	0.2000	0.0200
2	1	4	0.0500	0.2000	0.0200
3	1	5	0.0800	0.3000	0.0300
4	2	3	0.0500	0.2500	0.0300
5	2	4	0.0500	0.1000	0.1000
6	2	5	0.1000	0.3000	0.0200
7	2	6	0.0700	0.2000	0.2500

8	3	5	0.1200	0.2600	0.0250
9	3	6	0.0200	0.1000	0.0100
10	4	5	0.2000	0.4000	0.0400
11	5	6	0.1000	0.3000	0.0300

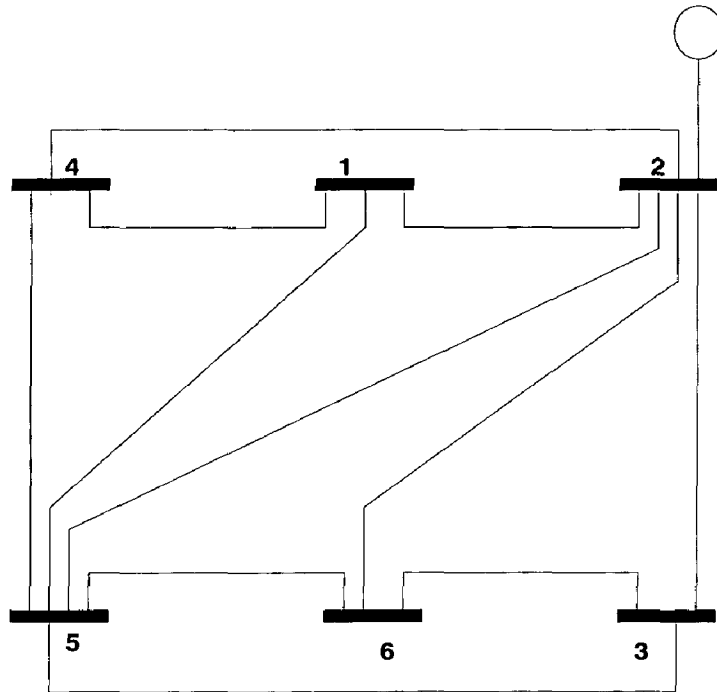


Fig.A1

Data for IEEE 14 bus system (at 100 MVA base)

The IEEE 14 bus system is shown on Fig. A2. The relevant data are provided in the following tables.

Table A4
Generator Power Data

Bus No.	Real Power Generation		Reactive Power Limits		Specified voltage(pu)
	Max(MW)	Min(MW)	Max(MVAR)	Min(MVAR)	
1	250	50	100	-45.0	1.060
2	200	20	50	-40.0	1.045
3	200	000	24	-06.0	1.010
6	250	000	40	0	1.070
8	000	000	24	-06.0	1.090

Table A5
Reactor/Capacitor data

Bus No.	MVA(pu)
8	00.19

Table A6
Load bus data

Bus No.	Load	
	Real(MW)	Reactive(MVAR)
2	21.70	12.70
3	94.20	19.00
4	47.80	-03.90
6	11.20	07.50
9	29.50	16.60

10	09.00	05.80
11	03.50	01.80
12	06.10	01.60
13	13.50	05.80
14	14.90	05.00

Table A7
Transformer Data

Line no	From Bus	To Bus	Series Impedance		Taps
			Resistance	Reactance	
8	4	7	0.00001	0.20912	0.978
9	4	9	0.00001	0.55618	0.969
10	5	6	0.00001	0.25202	0.932

Table A8
Line data

Line No.	From Bus	To Bus	Series Impedance(p.u.)		Half Line Charging(p u)
			Resistance	Reactance	
1	1	2	0.0194	0.0592	0.0528
2	1	5	0.0540	0.2230	0.0492
3	2	3	0.0470	0.1980	0.0438
4	2	4	0.0581	0.1763	0.0340
5	2	5	0.0570	0.1739	0.0346
6	3	4	0.0670	0.1710	0.0128
7	4	5	0.0134	0.0421	.000000
11	6	11	0.0950	0.1989	0.01870
12	6	12	0.1229	0.2558	0.00000

13	6	13	0.0662	0.1303	0.00000
14	7	8	0.0001	0.1762	0.00000
15	7	9	0.0001	0.1100	0.00000
16	9	10	0.0318	0.0845	0.00000
17	9	14	0.1271	0.2704	0.00000
18	10	11	0.0820	0.1921	0.00000
19	12	13	0.2209	0.1999	0.00000
20	13	14	0.1709	0.3480	0.00000

