STATE ESTIMATION OF POWER DISTRIBUTION SYSTEM

A DISSERTATION

submitted in partial fulfillment of the
requirements for the award of the degree
of
MASTER OF ENGINEERING
in
WATER RESOURCES DEVELOPMENT
(Hydro-Electric System Engineering & Management)

By

BISWARUP SADHUKHAN

WATER RESOURCES DEVELOPMENT TRAINING CENTRE
INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE
ROORKEE - 247 667 (INDIA)
DECEMBER, 2001

CANDIDATE' S DECLARATION

I hereby declare that the work, which is being presented in this dissertation titled **"STATE ESTIMATION OF POWER DISTRIBUTION SYSTEM"** in the partial fulfillment of the requirement for the award of Degree of Master of Engineering in Hydro-electric System Engineering and Management,. submitted in the Department of Water Resources Development Training Center, Indian Institute of Technology, Roorkee, is an authentic record of my own work carried out during a period of July 2001 to Dec 2001 under the supervision of **Dr. Biswarup Das**, *Assistant Professor, Electrical Engineering Department, and* **Prof. Devadutta Das**, *Professor & Head, Water Resources Devevopment Training Centre* Indian Institute of Technology , Roorkee.

The matter presented in this Dissertation has not been submitted by me for award of any other degree.

Dated: December 14, 2001
Place : Roorkee

Biswarup Sadhukhan
(BISWARUP SADHUKHAN)
Trainee Officer,
45th Batch

This is to certify that the statement made by the candidate is correct to the best of our knowledge.

Biswarup Das
(Dr. Biswarup Das)
Assistant Professor,
Electrical Engineering Department,
I.I.T. Roorkee, Roorkee
India

Devadutta Das
(Prof. Devadutta Das)
Professor & Head,
WRDTC
I.I.T. Roorkee, Roorkee
India

ACKNOWLEDGEMENT

I am greatly indebted to my guides Dr. Biswarup Das, Assistant Professor, Electrical Engineering Department and Prof. Devadutta. Das, Professor & Head, WRDTC, IIT, Roorkee, for their kind support and guidance during my dissertation work. Their co-operation and in-depth knowledge have made my work possible.

I am also thankful to O. C. computer Laboratory, for providing me the best facilities, which enabled my work to roll faster.

Last but not least, I am also grateful to all WRDTC faculty and my classmates whose presence have made a workable atmosphere in the laboratory, conducive for my work.

Biswarup Sadhukhan

(BISWARUP SADHUKHAN)
Trainee Officer

ABSTRACT

To implement different control decision function successfully in a modern distribution automation system, large number of system data are required from the field. Because of cost consideration, adequate numbers of meters can not be placed in the system to monitor the different system variables. Hence, state estimation technique is going to play a very crucial role in the coming days for successful implementation of "distribution automation" system. In this dissertation work, a distribution system state estimation algorithm is developed based on weighted least square approach. The state estimation problem has been formulated in rectangular co-ordinates. The algorithm is generalized enough to take into account all type of actual and pseudo-measurements. However, in this work, only measurement of power (both active and reactive) flows and magnitude of current flows over the feeders have been considered. The developed algorithm has been tested with two different system, namely, i) 18-bus system and ii) 31-bus system.

CONTENTS

CHAPTER -1

INTRODUCTION

Due to growth in population and Industry, the demand of electric power is ever - increasing. To supply this increased load demand, total power generation in India has grown rapidly during the past few decades. However, to supply quality power to the customers, it is not enough to merely increase the generation only. Adequate transmission and distribution facilities should also be created and maintained properly. To properly maintain the transmission and distribution system, it is very important to monitor these systems continuously. For this purpose, SCADA (supervisory control and data acquisition systems) have been introduced worldwide to continuously monitor and control the transmission systems. Similarly, for continuous monitoring and control of the distribution system, the concept of distribution automation system has been introduced about a decade ago [1].

In a distribution automation system, a number of meters, sensors and remote terminal units (RTU) are placed at appropriate positions in the power distribution system to measure different quantities such as voltage, current flow, power flow etc. The measured data are then sent back to a central computer station over a dedicated communication channel. These sent data are subsequently analyzed by appropriate analysis software package to assess the present health of the system. If any anomaly in the operation of the power distribution system is detected, proper remedial and control decisions are taken by the software analysis package at the central computer station. These remedial and control decisions are subsequently sent to the field over the same or other communication channel for proper field implementation.

The most common different control decision function, which are currently being employed in any modern distribution automation system are as follows:

- Accurate fault location in a distribution system
- Volt-var control
- Feeder reconfiguration to minimize the loss in the system remote monitoring
- Demand side-management
- Feeder load balancing

For successful implementation on the above control decision function, the information about various quantities in the power distribution system is necessary. For example, the volt- var control, feeder load balancing and feeder reconfiguration algorithms require the information of present system loading condition as input. Similarly, demand side management module utilizes the information of present system bus voltages and loading conditions to determine the appropriate load management policy.

The most obvious method for obtaining the above required data seems to be the direct measurement of those data by placing appropriate meters at appropriate places. However, since the distribution system in generally very large and dispersed over a wide geographical area, such an option would require a large number of meters, at an exorbitant cost. Thus, to keep the cost of meter placement low, only a limited number of meters are to be placed at some strategic locations of the power distribution system to measure some of the system variables and the rest of the system variables need to be estimated from these measurements. As any system variable can be expressed in terms of the bus voltage (magnitude and angles) of the system, the above estimation problem, reduces to the problem of estimating the bus voltage magnitude and angles from the available measurements.

The problem of estimation of bus voltages magnitudes and angles, known as the "state estimation (SE)" problem, has been studied extensively and implemented successfully in many parts of the world for electric power transmission system during the last few decades. An excellent survey of various SE techniques can be found in [2-4]. A comprehensive treatise on SE problem has been given in [5]. However, only in the last decade, due to the availability of technology to automatically monitor and control a distribution system, the interest for distribution system state estimation has grown. Consequently, several researchers have addressed this problem in the literature. In [6], a three-phase SE algorithm based on normal equation method for distribution system has been proposed. In this paper, the rectangular form of SE problem based on current, instead of power, has been introduced. A distribution system SE that minimizes the number of remote measurements is presented in [7]. An iterative process based on Kirchoff's current law was used to obtain the distribution system SE. The distribution system SE problem with zero injection constraint has been solved in [8]. A branch current based distribution system SE has been proposed in [9]. A distribution SE using stochastic load model has been proposed in [10]. A highly efficient algorithm for treatment of current measurements in branch-current based distribution SE has been suggested in [11].

In this dissertation, the state estimation problem for power distribution system has been solved based on normal equation method. The SE algorithm has been formulated in rectangular co-ordinates. The developed algorithm is generalized enough to take into account different types of measurements such as branch current flow, branch power flow (both real and reactive), bus voltage magnitude etc.

This thesis report is organized as follows. Chapter 2 describes the formulation of SE problem in rectangular form. Chapter 3 presents the main results of this work. Chapter 4 delineates the main conclusion of this work and gives a brief suggestion for further extension of this work.

CHAPTER-2

STATE ESTIMATION TECHNIQUE

2.1 FUNDAMENTAL EQUATION OF STATE ESTIMATION PROBLEM

In state estimation, the model used to relate the measurements to the state variable in as follows:

$$Z = h\,(X) + N \qquad \text{...(2.1)}$$

Where,

Z = Vector of measurements
X = Vector of state variables
N = Vector of measurement noise
h = Non-linear functions relating state variables to the measurements

The aim of the state estimation algorithm is to calculate (X) such that the Euclidean norm of N will be minimized. Thus, the "Cost",

$$C = \|N\|^2 = N^T N = [Z-h\,(X)]^T\,[Z-h(X)] \qquad \text{...(2.2)}$$

should be minimized.

Thus, in the absence of any constraints (and no constraint has been considered in this work), the state estimation problem is essentially an unconstrained optimization problem.

The condition for minimizing the objective function 'C' in eq. (2.2) is as:

$$\frac{\partial C}{\partial X} = \frac{\partial}{\partial X}\left[\{Z-h(X)\}^T\{Z-h(X)\}\right] = 0 \qquad \text{...(2.3)}$$

Now, as h(X) is a non-linear function of X, it can be expanded by Taylor's series as,

$$h(X) \approx h(X_o) + \left.\frac{\partial h(X)}{\partial X}\right|_{x=0} \Delta X + \left.\frac{\partial^2 h(X)}{\partial X}\right|_{x=0} \frac{\Delta X^2}{2} + \ldots\ldots\ldots\ldots \qquad \text{...(2.4)}$$

where X_0 is the initial estimate of X.

Neglecting the higher order terms, we have,

$$h(X) = h(X_0) + \left.\frac{\partial h(X)}{\partial X}\right|_{x=0} \Delta X = h(X_0) + J_0\,\Delta X \qquad \text{...(2.5)}$$

Where, J_0 is the Jacobian matrix evaluated as $X = X_0$

From eqs. (2.3) and (2.5), we have,

$$\frac{\partial C}{\partial X} = \frac{\partial}{\partial X}\left[\{Z - h(X_0) - J_o \Delta X\}^T \{Z - h(X_o) - J_0 \Delta X\}\right] = 0 \qquad \ldots(2.6)$$

Solving eq. (2.6) one gets [12],

$$\Delta X = (J_o^T J_0)^{-1} J_o^T \{Z - h(X_0)\} \qquad \ldots(2.7)$$

Generally, only Euclidean norm is not used but the effects of measurement errors are also included. Thus, the cost function is normally $C = N^T R N$, where R is a diagonal matrix having $R_{ii} = \sigma_{ii}^2$, the variance of the ith measurement. Proceeding as before, the correction vector for the state variables can be obtained as [12].

$$(\Delta X)^T = (J_0^T R^{-1} J_o)^{-1} (J_o^T R^{-1}) [Z - h(X_o)] \qquad \ldots(2.8)$$

Once the correction vector is obtained, a new estimate $X = X_0 + \Delta X$ is obtained for the next iteration and again the new correction vector (ΔX) is obtained till the convergence is obtained. Thus, the state estimation algorithm is an iterative procedure, which, in a finite number of steps, calculates the vector X to a degree of accuracy. The iterative algorithm can be summarized as follows:

$$X^{n+1} = X^n + [J_n^T R^{-1} J_n]^{-1} \; [J_n^T R^{-1}][Z - h(X_n)] \qquad \ldots(2.9)$$

$$\text{Until } \max\, [X_k^{n+1} - X_k^n] < \varepsilon, \text{ for all } k \qquad \ldots(2.10)$$

Where, X^n and X^{n+1} are the vector X at nth and (n+1)th iterations, J_n is the Jacobian matrix evaluated with X^n, $h(X_n)$ is the value of the non-linear function evaluated with X^n and X_k^{n+1} is the kth element of vector X at (n+1)th iteration. The matrix J_n is also called a measurement Jacobian matrix. Obviously, the elements of the matrix J_n depend on the types of measurements considered for estimating the states. This is because, as the type of measurement changes, the function h(X) relating that particular type of measurement to the stage variables also changes. In this work, only three types of measurements, namely, feeder real power flow, feeder reactive power flow and the feeder current magnitude have been considered. As state variables, the real and imaginary parts of the bus voltages have been considered. Thus, in this work, rectangular form of state estimation problem has been adopted, which is described in detail in the next section.

2.2 RECTANGULAR FORM OF STATE ESTIMATION

In rectangular form, the individual bus voltages are represented as $\bar{V}_i = e_i + jf_i$, for i = 1,2, ...n, where n is the number of buses in the system. e_i and f_i are the real and imaginary parts of the bus voltages respectively. Thus, in this formulation, the state variables are e_i and f_i, for i = 1,2,....n and hence, the total number of states to be estimated is 2(n-1) (excluding the slack bus, of which the voltage is completely known).

The measurements P_{ij}, Q_{ij} and $|I_{ij}|$ can be expressed as follows:

$$P_{ij} = (e_i - e_j)(e_i g_{ij} + f_i b_{ij}) + (f_i - f_j)(f_i g_{ij} - e_i b_{ij}) \qquad \ldots(2.11)$$

$$Q_{ij} = (e_i - e_j)(f_i g_{ij} - e_i b_{ij}) - (f_i - f_j)(f_i b_{ij} + e_i g_{ij}) \qquad \ldots(2.12)$$

$$|I_{ij}| = \sqrt{\{(e_i - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 + \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2} \qquad \ldots(2.13)$$

Where,

P_{ij} = real power flow on the feeder between bus i and bus j

Q_{ij} = reactive power flow on the feeder between bus i and bus j

$|I_{ij}|$ = Current magnitude on the feeder between bus i and bus j

The derivation of eqs. (2.11) - (2.13) are given in Appendix A in detail. From these equations, the various elements of the Jacobian matrix can be calculated as follows:

The Jacobian elements with respect to state variables e_i, e_j, f_i, f_j are :

$$\begin{aligned}\partial P_{ij} / \partial e_i &= (e_i - e_j) g_{ij} + (e_i g_{ij} + f_i b_{ij}) + (f_i - f_j)(-b_{ij}) \\ &= e_i g_{ij} - e_j g_{ij} + e_i g_{ij} + f_i b_{ij} - f_i b_{ij} + f_j b_{ij} \\ &= 2e_i g_{ij} - e_j g_{ij} + f_j b_{ij} \qquad \ldots(2.14)\end{aligned}$$

$$\partial P_{ij}/\partial e_j = -(e_i g_{ij} + f_i b_{ij}) \qquad \ldots(2.15)$$

$$\begin{aligned}\partial P_{ij} / \partial f_i &= (e_i - e_j)\ b_{ij} + (f_i - f_j) g_{ij} + (f_i g_{ij} - e_i b_{ij}) \\ &= e_i b_{ij} - e_j b_{ij} + f_i g_{ij} - f_j g_{ij} + f_i g_{ij} - e_i b_{ij} \\ &= 2f_i g_{ij} - e_j b_{ij} - f_j g_{ij} \qquad \ldots(2.16)\end{aligned}$$

$$\partial P_{ij}/\partial f_j = -(f_i g_{ij} - e_i b_{ij}) \qquad \ldots(2.17)$$

$$\begin{aligned}\partial Q_{ij} / \partial e_i &= (e_i - e_j)(-b_{ij}) + (f_i g_{ij} - e_i b_{ij}) - (f_i - f_j)(g_{ij}) \\ &= -e_i b_{ij} + e_j b_{ij} + f_i g_{ij} - e_i b_{ij} - f_i g_{ij} + f_i g_{ij} \\ &= -2e_i b_{ij} + e_j b_{ij} + f_j g_{ij} \qquad \ldots(2.18)\end{aligned}$$

$$\partial Q_{ij} / \partial e_j = -f_i g_{ij} + e_i b_{ij} \qquad \ldots(2.19)$$

$$\partial Q_{ij} / \partial f_i = (e_i - e_j) g_{ij} - (f_i b_{ij} + e_i g_{ij}) - (f_i - f_j) b_{ij}$$

$$= e_i g_{ij} - e_j g_{ij} - f_i b_{ij} - e_i g_{ij} - f_i b_{ij} + f_j b_{ij}$$

$$= - 2 f_i b_{ij} - e_j g_{ij} + f_j b_{ij} \qquad \ldots(2.20)$$

$$\partial Q_{ij} / \partial f_j = (f_i b_{ij} + e_i g_{ij}) \qquad \ldots(2.21)$$

Jacobian terms of current magnitude measurement can be expressed as,

$$\frac{\partial |I_{ij}|}{\partial e_i} = \frac{1}{2} \frac{2\{(e_i - e_i)g_{ij} - (f_i - f_j)b_{ij}\}g_{ij} + 2\{(f_i - f_j)g_{ij} + (e_i + e_j)b_{ij}\}b_{ij}}{\sqrt{\{(e_l - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 + \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2}}$$

$$= \frac{\{(e_i - e_i)g_{ij} - (f_i - f_j)b_{ij}\}g_{ij} + \{(f_i - f_j)g_{ij} + (e_i + e_j)b_{ij}\}b_{ij}}{\sqrt{\{(e_l - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 + \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2}}$$

$$= \frac{(e_i - e_i)g_{ij}^2 - (f_i - f_j)b_{ij}g_{ij} + (f_i - f_j)g_{ij}b_{ij} + (e_i + e_j)b_{ij}^2}{\sqrt{\{(e_l - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 + \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2}}$$

$$= \frac{(e_i - e_j)(g^2{}_{ij} + b_{ij}{}^2)}{\sqrt{\{(e_i - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 + \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2}} \qquad \ldots(2.22)$$

Similarly,

$$\frac{\partial |I_{ij}|}{\partial e_j} = \frac{-(e_i - e_j)(g^2{}_{ij} - b^2{}_{ij})}{\sqrt{\{(e_i - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 + \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2}} \qquad \ldots(2.23)$$

$$\frac{\partial |I_{ij}|}{\partial f_i} = \frac{(f_i - f_j)(g^2{}_{ij} + b^2{}_{ij})}{\sqrt{\{(e_i - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 + \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2}} \qquad \ldots(2.24)$$

$$\frac{\partial |I_{ij}|}{\partial f_j} = \frac{-(f_i - f_j)(g_{ij}{}^2 + b_{ij}{}^2)}{\sqrt{\{(e_i - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2}} \qquad \ldots(2.25)$$

Thus the structure of the combined measurement Jacobian matrix can be expressed as,

State variables

$$J = \text{Measurements} \quad \begin{array}{c|cccc} & e_i & e_j & f_i & f_j \\ \hline P_{ij} & \dfrac{\partial P_{ij}}{\partial e_i} & \dfrac{\partial P_{ij}}{\partial e_j} & \dfrac{\partial P_{ij}}{\partial f_i} & \dfrac{\partial P_{ij}}{\partial f_j} \\ Q_{ij} & \dfrac{\partial Q_{ij}}{\partial e_i} & \dfrac{\partial Q_{ij}}{\partial e_j} & \dfrac{\partial Q_{ij}}{\partial f_i} & \dfrac{\partial Q_{ij}}{\partial f_j} \\ |I_{ij}| & \dfrac{\partial |I_{ij}|}{\partial e_i} & \dfrac{\partial |I_{ij}|}{\partial e_j} & \dfrac{\partial |I_{ij}|}{\partial f_i} & \dfrac{\partial |I_{ij}|}{\partial f_j} \end{array}$$

Thus, in every iteration, the Jacobian matrix is re-calculated with the help of the eqs. (2.14) - (2.25) and subsequently, the iterative procedure in eq. (2.9) is followed till the convergence criteria in eq. (2.10) is satisfied. The step-by-step algorithm for the state estimation procedure is given below.

Step 1 : Read the input data (line resistance and reactance, the actual and Pseudo measurements, matrix R of co-variance of measurements, tolerance for convergence).

Step 2 : Assume initial bus voltage (magnitudes and angles).

Step 3 : Compute the calculated value of different measurement variables.

Step 4 : Compute the measurement error $[Z_i-h_i(X)]$, where, Z_i = i-th actual measurement and $h_i(X)$ = calculated value of i-th measurement.

Step 5 : Compute the Jacobian Matrix ([J]).

Step 6 : Compute the correction vector $[\Delta X^T] = [J^T R^{-1} J]^{-1} * J^T R^{-1} * [Z_i-h_i(X)]$.

Step 7 : Compute the absolute value of the highest element of ΔX, $\Delta X_{max.}$

Step 8 : if $\Delta X_{max} <$ tolerance, stop iteration and print results.

Step 9 : if $\Delta X_{max} >$ tolerance, update the state variable $X_i^{k+1} = X_i^k + \Delta X_i^k$ and go back to step 3.

In the next chapter, results for state estimation on two sample distribution systems are presented.

CHAPTER 3

RESULTS AND DISCUSSIONS

The validity of the state estimation procedure as developed in Chapter 2, has been checked on two sample distribution system namely, i) 18 bus system and ii) 31-bus system. The data for these two systems are given in Appendix B, in Tables B.1 and B.2 respectively. The schematic diagram of these two systems are given in Figs. C.1 and C.2 in Appendix C respectively. As explained in Chapter 2, power flows (both real and reactive) and current flows over the feeders have been considered as measurements. In the absence of field measurements, load flow analysis has been carried out for both these systems using the load data given in Appendix B and the solutions for power and current flows in the feeder obtained from load flows analysis have been taken as true measurements. To account for the measurement errors, following procedure has been undertaken.

A large number of random numbers, say N, have been generated between a given range, say ± p, where $0<p<1.0$. Subsequently, each time measurement (i.e. either power flow or current flow obtained from load flow solution) has been multiplied by these N numbers. These resulting N numbers have been considered as errors from true measurements (henceforth denoted by e_i, $i = 1, 2, \ldots N$) and subsequently, the standard deviation for m-th measurement has been calculated as $\sigma_m = \sqrt{\frac{\sum_{i=1}^{N} e_{im}^2}{N}}$, where e_{im} is the i-th error term for the m-th measurement. In this work, values of N and p have been taken as 10000 and 0.2 respectively. The calculated standard deviations are also given in Appendix B, in Tables B.3 and B.4 respectively. Lastly any value from the above N numbers has been picked up arbitrarily and each true measurement has been multiplied by the quantity (1-y), where y is the arbitrarily picked up random number. The resulting quantities have been considered as actual measurements.

Using the actual measurements and the corresponding standard deviations obtained as outlined above, the state-estimation problem has been solved for both the systems. The results are given as below.

(a) **18 Bus System.**

For the 18-bus system, various case studies have been carried out to find the effectiveness of the state estimation algorithm described in the previous chapter. In the first case study, only power flow (both real and reactive) measurements over all the feeders have been considered. As there are 17 feeders in a completely radial 18-bus distribution system, total number of measurements available is 34. It is to be noted that total number of state variables to be estimated for this system is also 34. Hence, the state estimation problem is solvable. In this case study, only true measurements (i.e. the feeder power flows obtained from the load flow study) but no actual measurements (as described at the beginning of this chapter) have been considered. The results are shown in Table 3.1. In this table, the solution of bus voltage magnitudes and angle obtained from both load flow and state estimation studies are shown. It is observed that when all the measurements are true measurements, the state estimation solution matches very closely with those of the load flow solution (the load flow solution has been taken as the true solution). Once the solutions for the state variables are found from state estimation technique, the feeder current and power flows for all the feeders can also be estimated very easily. Subsequently, applying KCL at each node, the load power (both real and reactive) at that node can also be estimated. The results of load estimation are also shown in Table 3.1. Comparison of estimated loads with the system loads (given in Table B.1 in Appendix B) reveals that the estimated loads also match very closely with the actual loads in the system.

Table 3.1 : 18-bus system, with true P, Q measurements only.

Bus i	Load Flow Solution		State-estimation results			
	V_{real} (kV)	$V_{imaginary}$ (kV)	V_{real} (kV)	$V_{imaginary}$ (kV)	P_{cal} (kW)	Q_{cal} (kVAR)
1	23.000000	0.000000	23.000000	0.000000	000.000	000.000
2	22.928640	-0.188339	22.928640	-0.188339	000.005	000.001
3	22.892155	-0.188170	22.892155	-0.188170	500.034	200.011
4	22.868084	-0.188081	22.868084	-0.188081	499.903	199.963
5	22.850353	-0.188045	22.850353	-0.188045	500.057	200.021
6	22.838968	-0.188061	22.838968	-0.188061	500.024	200.008
7	22.833931	-0.188129	22.833931	-0.188129	399.983	149.993
8	22.873589	-0.189326	22.873587	-0.189326	450.013	150.004
9	22.857174	-0.189409	22.857173	-0.189409	499.918	199.967
10	22.845089	-0.189571	22.845088	-0.189571	400.033	150.011
11	22.835016	-0.189706	22.835014	-0.189706	400.001	150.000
						Contd..

12	22.844925	-0.189876	22.844923	-0.189876	400.102	150.039
13	22.822302	-0.190344	22.822302	-0.190344	499.911	199.965
14	22.812304	-0.190699	22.812302	-0.190699	399.984	149.993
15	22.892958	-0.190139	22.892958	-0.190139	599.924	199.970
16	22.875111	-0.191039	22.875110	-0.191039	600.098	200.036
17	22.866186	-0.191490	22.866186	-0.191499	599.957	199.984
18	22.806337	-0.191000	22.806336	-0.191000	600.041	200.0147

In the second case study, instead of the true measurements, only actual measurements have been considered. The actual measurements have been obtained following the procedure outlined at the beginning of this chapter. The results are shown in Table 3.2. Comparison of the results in Table 3.1 and 3.2 shows that due to the presence of errors in measurements, the quality of estimation deteriorates.

Table 3.2 : 18-bus system, with actual P, Q measurement only

Bus i	Load Flow Solution		State-estimation results			
	V_{real} (kV)	$V_{imaginary}$ (kV)	V_{real} (kV)	$V_{imaginary}$ (kV)	P_{cal} (kW)	Q_{cal} (kVAR)
1	23.000000	0.000000	23.000000	0.000000	000.000	000.000
2	22.928640	-0.188339	22.921505	-0.207173	-000.010	-8.135
3	22.892155	-0.188170	22.881359	-0.206954	549.456	219.784
4	22.868084	-0.188081	22.854868	-0.206835	549.974	219.987
5	22.850353	-0.188045	22.835354	-0.206778	549.720	219.886
6	22.838968	-0.188061	22.822881	-0.206485	550.065	220.027
7	22.833931	-0.188129	22.817277	-0.206856	439.942	164.975
8	22.873589	-0.189326	22.860927	-0.208209	494.088	164.632
9	22.857174	-0.189409	22.842861	-0.208285	549.919	219.967
10	22.845089	-0.189571	22.829559	-0.208453	439.889	164.955
11	22.835016	-0.189706	22.818470	-0.208592	440.019	165.006
12	22.844925	-0.189876	22.829380	-0.208788	439.651	164.862
13	22.822302	-0.190344	22.804477	-0.209283	549.976	219.987
14	22.812304	-0.190699	22.793470	-0.209664	439.864	164.944
15	22.892958	-0.190139	22.882240	-0.209121	659.643	219.856
16	22.875111	-0.191039	22.862598	-0.210096	660.011	220.003
17	22.866186	-0.191490	22.852776	-0.210583	659.913	219.965
18	22.806337	-0.191000	22.786902	-0.209990	659.983	219.994

To improve the quality of estimation, the number of measurements used for state estimation has been increased. This has been achieved by considering the magnitude of current flow over the feeders as additional measurements. Thus, the total number of measurements has been increased to 51 from 34 in previous case studies and consequently, the state estimation problem has now been made a case of over-determined state estimation problem. For this purpose, only actual values of the current magnitude measurements have been considered. The same procedure outlined at the beginning of this chapter has been again used for obtaining the actual value of current magnitude measurements. The results are shown in Table 3.3. Comparison of results of Table 3.2 and 3.3 shows that the quality of state estimation improves when an exact state estimation problem is converted into an over-determined state estimation problem by considering additional measurements.

Table 3.3 : 18-bus system, with actual P, Q & I measurements

Bus i	Load Flow Solution		State-estimation results			
	V_{real} (kV)	$V_{imaginary}$ (kV)	V_{real} (kV)	$V_{imaginary}$ (kV)	P_{cal} (kW)	Q_{cal} (kVAR)
1	23.000000	0,000000	23.000000	0.000000	000.000	000.000
2	22.928640	-0.188339	22.928665	-0.188344	44.005	000.775
3	22.892155	-0.188170	22.892133	-0.188190	501.297	200.088
4	22.868084	-0.188081	22.868038	-0.188109	501.238	200.064
5	22.850353	-0.188045	22.850297	-0.188076	501.390	200.121
6	22.838968	-0.188061	22.838917	-0.188090	501.142	200.017
7	22.833931	-0.188129	22.833898	-0.188152	398.386	149.925
8	22.873589	-0.189326	22.873882	-0.189326	443.075	149.701
9	22.857174	-0.189409	22.857490	-0.189241	501.320	200.078
10	22.845089	-0.189571	22.845449	-0.189316	398.335	149.902
11	22.835016	-0.189706	22.835413	-0.189464	398.359	149.912
12	22.844925	-0.189876	22.845369	-0.189587	298.294	149.895
13	22.822302	-0.190344	22.822873	-0.189741	501.394	200.063
14	22.812304	-0.190699	22.812971	-0.190167	398.195	149.820
15	22.892958	-0.190139	22.893465	-0.190489	590.884	199.659
16	22.875111	-0.191039	22.875859	-0.189976	591.079	199.709
17	22.866186	-0.191490	22.867055	-0.191201	590.943	199.648
18	22.806337	-0.191000	22.807086	-0.1907762	591.004	199.670

b) 31-bus system:

The same case studies as in the case of 18-bus system have also been carried out for the 31-bus system. The results for state estimation with only true P and Q measurements are shown in Table 3.4. Again, from Table 3.4 and Table B.2 in Appendix B it is found that when only true measurements are considered, even for an exact state estimation problem, the solution obtained is very close to the true solution. The results of state estimation considering only actual P, Q measurements are shown in Table 3.5. As before, from Tables 3.4 and 3.5 it is found that the errors in the measurements deteriorate the quality of the state estimation.

Table 3.4 : 31-bus system, with true P, Q measurement only

Bus i	Load Flow Solution		State-estimation results			
	V_{real} (kV)	$V_{imaginary}$ (kV)	V_{real} (kV)	$V_{imaginary}$ (kV)	P_{cal} (kW)	Q_{cal} (kVAR)
1	23.00000	0.00000	23.00000	0.000000	0.00000	0.00000
2	22.31724	0.113224	22.31724	0.113224	-0.00592	-0.00040
3	22.14691	0.163232	22.14691	0.163232	521.9824	173.9968
4	21.80192	0.001054	21.80192	0.001054	0.014463	0.012657
5	21.40749	-0.15131	21.40749	-0.151310	935.9647	311.9791
6	21.06191	-0.28378	21.06191	-0.283780	-0.00562	-0.00185
7	20.55888	-0.36677	20.55888	-0.366770	0.020681	0.012081
8	20.20064	-0.42477	20.20064	-0.424770	-0.015580	-0.00788
9	19.84240	-0.48278	19.84240	-0.482780	0.000293	-0.00149
10	19.60931	-0.52001	19.60931	-0.520010	188.9890	62.99442
11	19.39202	-0.55461	19.39202	-0.554610	0.000308	-0.00049
12	19.17473	-0.58922	19.17473	-0.589220	335.9750	111.9866
13	18.98618	-0.61917	18.98618	-0.619170	656.9177	218.9559
14	18.85439	-0.64004	18.85439	-0.640040	782.9176	260.9546
15	18.79074	-0.65009	18.79074	-0.650090	728.9244	242.9587
16	19.75644	-0.51582	19.75644	-0.515820	476.9704	158.9848
17	19.67086	-0.53000	19.67086	-0.530000	548.9578	182.9768
18	19.63103	-0.53659	19.63103	-0.536590	476.9652	158.9808
19	20.47088	-0.40122	20.47088	-0.401220	431.9854	143.9940
20	20.40658	-0.42637	20.40658	-0.426370	671.9506	223.9690
21	20.36677	-0.43320	20.36677	-0.433200	494.9822	164.9904
22	20.54758	-0.371230	20.54758	-0.371230	206.9888	68.99359
23	21.66630	-0.064390	21.66630	-0.064390	521.9703	173.9759
24	21.54492	-0.122870	21.54492	-0.122870	1916.994	639.0141
25	21.41534	-0.174960	21.41533	-0.174960	-0.01950	-0.01662
26	21.28575	-0.227040	21.28575	-0.227040	1115.97700	371.9896
27	21.21484	-0.255480	21.21484	-0.255480	549.0085	183.0080
28	21.1536	-0.266540	21.15360	-0.266540	791.9716	263.9861
29	22.28331	0.122377	22.28331	0.122377	882.0001	293.9960
30	22.15327	0.096648	22.15327	0.096648	882.0120	294.0101
31	22.08815	0.083783	22.08815	0.083783	881.9774	293.9906

Table 3.5 : 31-bus system, with actual P, Q measurement only

Bus i	Load Flow Solution		State-estimation results			
	V_{real} (kV)	$V_{imaginary}$ (kV)	V_{real} (kV)	$V_{imaginary}$ (kV)	P_{cal} (kW)	Q_{cal} (kVAR)
1	23.00000	0.00000	23.00000	0.000000	0.000000	0.00000
2	22.31724	0.113224	22.24896	0.124546	-57.0653	-9.87350
3	22.14691	0.163232	22.06099	0.179625	560.9688	190.6898
4	21.80192	0.001054	21.68017	0.000242	-19.7594	-19.5358
5	21.40749	-0.15131	21.24385	-0.16828	1014.996	330.5159
6	21.06191	-0.28378	20.86067	-0.31479	-11.6162	-10.0968
7	20.55888	-0.36677	20.30186	-0.40606	-19.1722	-10.8036
8	20.20064	-0.42477	19.90269	-0.46981	-10.2264	-5.76050
9	19.84240	-0.48278	19.50258	-0.53353	-10.6572	-6.00451
10	19.60931	-0.52001	19.24161	-0.57441	203.1840	66.64315
11	19.39202	-0.55461	18.99792	-0.61239	-4.17726	-2.35307
12	19.17473	-0.58922	18.75383	-0.65035	365.2680	120.76100
13	18.98618	-0.61917	18.54166	-0.68320	719.3239	238.9986
14	18.85439	-0.64004	18.39314	-0.70607	859.5366	286.1089
15	18.79074	-0.65009	18.32133	-0.71709	801.4256	267.0341
16	19.75644	-0.51582	19.40627	-0.57022	523.8301	174.1506
17	19.67086	-0.53000	19.31039	-0.58578	603.2019	200.9082
18	19.63103	-0.53659	19.26574	-0.59301	524.5218	174.7999
19	20.47088	-0.40122	20.20376	-0.44422	474.3608	157.6782
20	20.40658	-0.42637	20.13202	-0.47207	738.7122	245.9867
21	20.36677	-0.43320	20.08762	-0.47958	544.3416	181.4107
22	20.54758	-0.37123	20.28926	-0.41100	227.6743	75.88149
23	21.66630	-0.06439	21.53015	-0.07215	570.9976	188.2402
24	21.54492	-0.12287	21.39576	-0.13683	2106.054	700.3078
25	21.41534	-0.17496	21.25218	-0.19442	-1.613410	-1.4026
26	21.28575	-0.22704	21.10849	-0.25202	1225.915	407.7460
27	21.21484	-0.25548	21.02980	-0.28346	603.3700	200.8487
28	21.15360	-0.26654	20.96183	-0.29564	870.9138	290.2406
29	22.28331	0.122377	22.21152	0.134625	969.6556	323.3710
30	22.15327	0.096648	22.06803	0.106150	969.0424	322.7514
31	22.08815	0.083783	21.99613	0.091912	969.9018	323.2359

CHAPTER-4

CONCLUSION

In this dissertation, rectangular formulation of weighted least square method is used for distribution system state estimation. The algorithm is generalized enough to considered all type of measurements. Test results have shown that the proposed rectangular form formulation is a suitable choice for the distribution system state estimation. The main conclusions of this work are as follows :

i) If all the measurements are true measurements, then the state estimation solution is very close to the true solution.

ii) In the presence of errors in the measurements, the quality of the state estimation deteriorates.

iii) The quality of the solution of the state estimation problem can be improved by increasing the dimension of the measurement vector.

Scope of future work

i) Techniques for bad data detection need to be incorporated in the developed algorithm.

ii) Instead of rectangular formulation of the state estimation problem, current based formulation may be investigated, where the gain matrix becomes a constant matrix resulting in faster execution of the algorithm.

iii) Techniques for obtaining pseudo-measurements may be investigated.

REFERENCES

1. Kazuaki Kato, Hiromi Nagasaka, Akimichi-Okimato, Toshihito Kunieda and Toshihiko Nakamura, "Distribution Automation Systems for High Quality Power Supply", IEEE Transaction on Power Delivery, Vol. 6, No. 3, July 1991, pp : 1196-1203.
2. F. F. Wu, "Power System State Estimation: A Survey", Electrical Power & Energy System, Vol. 12, No. 2, April 1990.
3. M. B. Do Coutto Filho, A. M. Liete da Silva and D. M. Falcao, "Bibliography on Power System State Estimation (1968-1989)", IEEE Transaction on Power System, Vol. 5, No. 3, August 1990, pp : 950-957.
4. A. Bose and K. Clements, "Real Time Modeling of Power Networks", IEEE Proceedings, pp. 1607-1622, December 1987.
5. A. Monticelli, "State Estimation in Electric Power System, A Generalized Approach", Kluwer Academic Publisher, 1999
6. C. N. Lu, J. H. Teng, and W. H. E. Liu, " Distribution System State Estimation", IEEE Transaction on Power System, Vol. 10, No. 1, February 1995, pp : 229-240.
7. I. Roytelman and S. M. Shahidehpour, "State Estimation for Electric Power Distribution System in Quasi-real Time Condition", IEEE Transaction on Power Delivery, Vol. 8, No. 4, October 1993, pp: 2009-2015.
8. W. M. Lin and J. H. Teng, "State Estimation for Distribution System with Zero-injection Constraints", IEEE Transaction on Power System, Vol. 11, No. 1, February 1996, pp : 518-524.
9. M. E. Baran and A. W. Kelley, "A Branch Current Based State Estimation for Distribution System", IEEE Transaction on Power System, Vol. 10, No. 1, February 1995, pp: 483-491.
10. Ke Li., "State Estimation for Power Distribution System and Measurement Impacts", IEEE Transaction on Power System, Vol. 11, No. 2, May 1996, pp: 911-916.
11. W. M. Lin, J. H. Teng and S. J. Chen, "A Highly Efficient Algorithm in Treating Current Measurements for the Branch Current Based Distribution State Estimation", IEEE Transaction on Power Delivery, Vol. 16, No. 3, July 2001, pp: 433-439.
12. Y. Wallach, "Calculation & Programs for Power System Networks", Prentice-Hall, 1986.

APPENDIX A

For power flow measurement P_{ij} and Q_{ij} at the line connected between bus i and j can be expressed as;

$$
\begin{aligned}
P_{ij} + jQ_{ij} &= V_{ij}\, I_{ij}^{*} \\
&= (e_i + jf_i)\,[\,(V_i - V_j)\,]^{*} \\
&= (e_i + jf_i)\,[\{(e_i + jf_i) - (e_j + jf_j)\}\,(g_{ij} + jb_{ij})\,]^{*} \\
&= (e_i + jf_i)\,[\{(e_i - e_j) + j\,(f_i - f_j)\}\,(g_{ij} + jb_{ij})\,]^{*} \\
&= (e_i + jf_i)\,[\{(e_i - e_j)\,g_{ij} - (f_i - f_j)\,b_{ij}\} + j\{(f_i - f_j)\;g_{ij} + (e_i - e_j)\,b_{ij}\}\,]^{*} \\
&= (e_i + jf_i)\,[\{(e_i - e_j)\,g_{ij} - (f_i - f_j)\,b_{ij}\} - j\{(f_i - f_j)\;g_{ij} + (e_i - e_j)\,b_{ij}\}\,]
\end{aligned}
$$

Separating real & imaginary parts we get,

$$
\begin{aligned}
P_{ij} &= e_i\,[\,(e_i - e_i)\,g_{ij} - (f_i - f_i)b_{ij}\,] + f_i\,[(f_i - f_j)\,g_{ij} + (e_i - e_i)\,b_{ij}\,] \\
&= (e_i - e_j)\,e_i\,g_{ij} - (f_i - f_j)\,e_i\,b_{ij} + (f_i - f_j)\,f_i\;g_{ij} + (e_i - e_j)\,f_i\;b_{ij} \\
&= (e_i - e_j)\,(e_i\,g_{ij} + f_i\,b_{ij}) + (f_i - f_j)\,(f_i\,g_{ij} - e_i\,b_{ij})
\end{aligned}
$$

$$
\begin{aligned}
Q_{ij} &= f_i\,[\,(e_i - e_j)\,g_{ij} - (fi - f_i)\,b_{ij}\,] - e_i\,[\,(f_i - f_j)\,g_{ij} + (e_i - e_j)b_{ij}\,] \\
&= (e_i - e_j)\,f_i\;g_{ij} - (f_i - f_j)\,f_i\;b_{ij} - (fi - fj)\,e_i\,g_{ij} - (e_i - e_j)\,e_i\,b_{ij} \\
&= (e_i - e_j)\,(f_i g_{ij} - e_i\,b_{ij}) - (f_i - f_j)\,(f_i b_{ij} + e_i\,g_{ij})
\end{aligned}
$$

Current magnitude can be expressed as follows

$$
\begin{aligned}
I_{ij} &= (V_i - V_j)\,Y_{ij} \\
&= \{\,(e_i + jf_i) - (e_j + jf_j)\,\}\,(\,g_{ij} + j\,b_{ij}) \\
&= \{\,(e_i - e_j) + j\,(f_i - f_j)\,\}\;(g_{ij} + j\,b_{ij})
\end{aligned}
$$

$$
\begin{aligned}
I_{ij} = \{\,&(e_i - e_j)\,g_{ij} - (f_i - f_j)\,b_{ij}\,\} + j \\
&\{\,(f_i - f_j)\,g_{ij} + (e_i - e_j)\,b_{ij}\,\}
\end{aligned}
$$

$$
|I_{ij}| = \sqrt{\{(e_i - e_j)g_{ij} - (f_i - f_j)b_{ij}\}^2 + \{(f_i - f_j)g_{ij} + (e_i - e_j)b_{ij}\}^2}
$$

APPENDIX B

Table B.1 : System network and load data for 18-bus system

Bus i	Bus j	Branch impedance		Max load at Bus j	
		R_{ij} (ohm)	X_{ij} (ohm)	P (KW)	Q (KVAR)
1	2	0.00	0.55	000.0	000.0
2	3	0.30	0.12	500.0	200.0
3	4	0.25	0.10	500.0	200.0
4	5	0.25	0.10	500.0	200.0
5	6	0.25	0.10	500.0	200.0
6	7	0.25	0.10	400.0	150.0
2	8	0.30	0.12	450.0	150.0
8	9	0.25	0.10	500.0	200.0
9	10	0.30	0.12	400.0	150.0
10	11	0.50	0.20	400.0	150.0
8	12	0.30	0.12	400.0	150.0
12	13	0.30	0.12	500.0	200.0
13	14	0.20	0.08	400.0	150.0
2	15	0.40	0.16	600.0	200.0
15	16	0.30	0.12	600.0	200.0
16	17	0.30	0.12	600.0	200.0
14	18	0.20	0.08	600.0	200.0

Base voltage (substation voltage) = 23.0 KV.

Table B.2 : System network and load data for 31-bus system

Bus i	Bus j	Branch impedance		Max load at Bus j	
		R_{ij} (ohm)	X_{ij} (ohm)	P (KW)	Q (KVAR)
1	2	0.896	0.155	000.0	000.0
2	3	0.279	0.155	522.0	174.0
3	4	0.444	0.439	000.0	000.0
4	5	0.864	0.751	936.0	312.0
5	6	0.864	0.751	000.0	000.0
6	7	1.374	0.774	000.0	000.0
7	8	1.374	0.774	000.0	000.0
8	9	1.374	0.774	000.0	000.0
9	10	1.374	0.774	189.0	63.0
10	11	1.374	0.774	000.0	000.0
11	12	1.374	0.774	336.0	112.0
12	13	1.374	0.774	657.0	219.0
13	14	1.374	0.774	783.0	261.0
14	15	1.374	0.774	729.0	243.0
9	16	0.864	0.751	477.0	159.0
16	17	1.374	0.774	549.0	183.0
17	18	1.374	0.774	477.0	159.0
7	19	0.864	0.751	432.0	144.0
19	20	0.864	0.751	672.0	224.0

Contd..

20	21	1.374	0.774	495.0	165.0
7	22	0.864	0.751	270.0	69.0
4	23	0.444	0.439	522.0	174.0
23	24	0.444	0.439	1917.0	639.0
24	25	0.864	0.751	000.0	000.0
25	26	0.864	0.751	1116.0	372.0
26	27	0.864	0.751	549.0	183.0
27	28	1.374	0.774	792.0	264.0
2	29	0.279	0.015	882.0	294.0
29	30	1.374	0.774	882.0	294.0
30	31	1.374	0.774	882.0	294.0

Base voltage (substation voltage) = 23.0 KV.

Table B.3 : Variance of measurements for 18-bus system

Bus i	Bus j	For P_{flow} (kw) measurement	For Q_{flow} (kvar) measurement	For I_{flow} magnitude (amp) measurement
1	2	2.089155	0.299915	2.389017
2	3	0.195196	0.030586	0.227174
3	4	0.122045	0.019018	0.142385
4	5	0.066164	0.010212	0.077254
5	6	0.027312	0.004131	0.031854
6	7	0.005391	0.000758	0.006236
2	8	0.452299	0.061914	0.517384
8	9	0.057051	0.008440	0.066213
9	10	0.021587	0.003036	0.024930
10	11	0.005393	0.000758	0.006235
8	12	0.122147	0.016586	0.140262
12	13	0.075986	0.010218	0.087372
13	14	0.033718	0.004131	0.038439
2	15	0.109593	0.012187	0.122532
15	16	0.048591	0.005401	0.054494
16	17	0.012134	0.001348	0.013629
14	18	0.012131	0.001348	0.013701

Table B.4 : Variance of measurements for 31-bus system

Bus i	Bus j	For P_{flow} (kw)	For Q_{flow} (kvar)	For I_{flow} magnitude(amp)
1	2	9.201213	1.119613	10.320835
2	3	5.999910	0.771731	7.192129
3	4	5.442919	0.714767	6.440847
4	5	1.931523	0.252546	2.430705
5	6	1.431350	0.181885	1.862089
6	7	1.391972	0.169806	1.862089
7	8	0.672736	0.081345	0.943489
8	9	0.651010	0.077111	0.943891
9	10	0.265984	0.031265	0.435029
10	11	0.226010	0.026269	0.346822
11	12	0.221367	0.025381	0.346822
12	13	0.163160	0.018495	0.261117
13	14	0.078219	0.008785	0.127544
14	15	0.018007	0.002009	0.029752
9	16	0.077152	0.008714	0.064812
16	17	0.035798	0.004004	0.011687
17	18	0.007691	0.000857	0.002008
7	19	0.087175	0.009869	0.009869
19	20	0.046181	0.005181	0.005181
20	21	0.008281	0.000922	0.000922
7	22	0.001444	0.000161	0.000161
4	23	0.833281	0.097888	0.097888
23	24	0.659843	0.076315	0.076315
24	25	0.208588	0.024098	0.024098
25	26	0.206438	0.023465	0.023465
26	27	0.061104	0.006867	0.006867
27	28	0.021239	0.002369	0.002369
2	29	0.238753	0.026622	0.281853
29	30	0.106245	0.011917	0.011917
30	31	0.026344	0.002938	0.002938

APPENDIX C

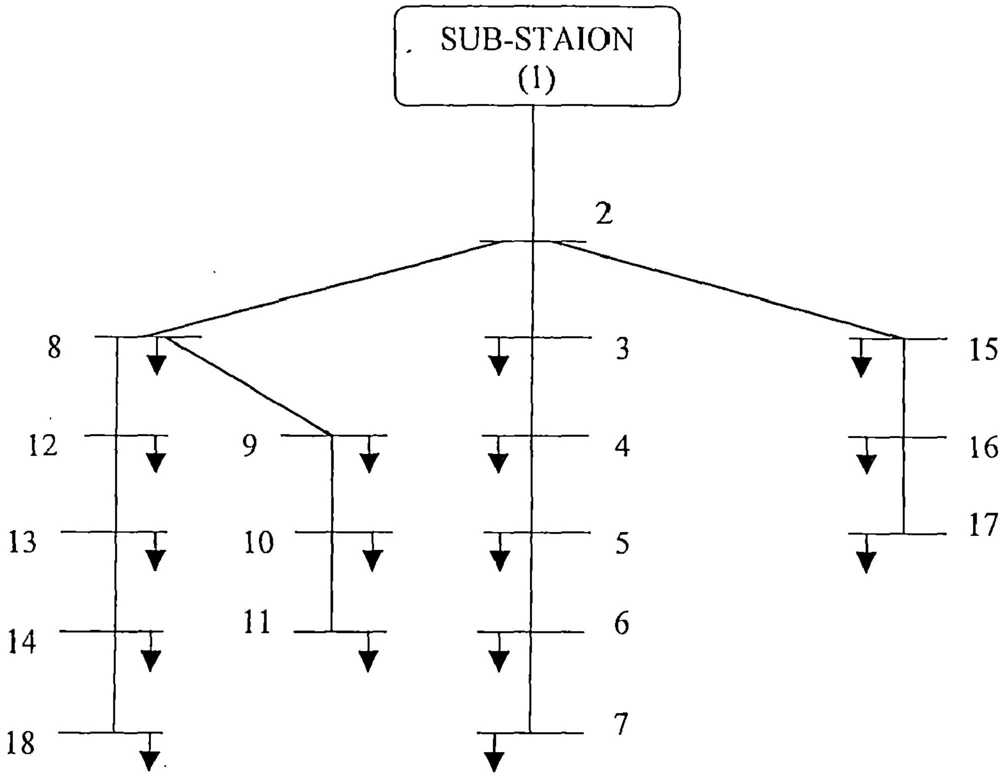

Figure C.1 : Schematic diagram of 18-bus distribution system

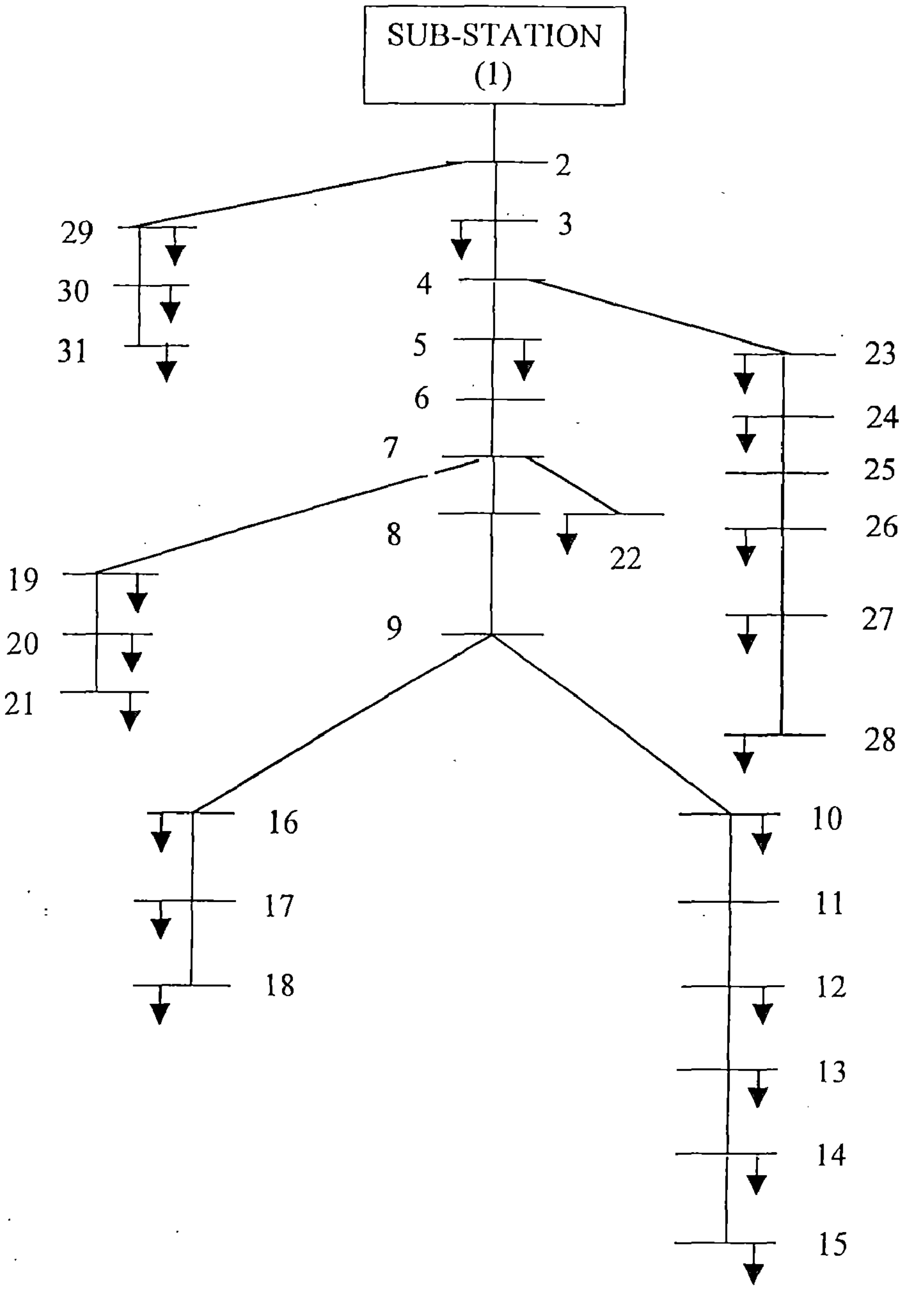

Figure C.2 : Schematic diagram of 31-bus distribution system

APPENDIX D

SOFTWARE FOR STATE ESTIMATION OF POWER DISTRIBUTION SYSTEM

```
//STATE ESTIMATION.CPP
//This program finds the state variables using Normal Equation method
#include<fstream.h>
#include<iostream.h>
#include<iomanip.h>
#include<math.h>
#include<string.h>
//*********Input & Output file****************************************
 ifstream infile("18b_inc.dat") ;
 ofstream outfile("18b_outc.dat") ;
//********** Global variables****************************************
 const int mm = 64 ;
 int nm , ns , nbus , nline , s, p , enda , endb , nodea ,nodeb , end ;
 int lfrom[mm] , lto[mm] , list[mm ], next[mm] , ifar[mm] , branch ;
 float r[mm] , invr[mm] , base_kva , base_kv , base_imp , tol ;
 float  srev[mm] , simv[mm] , orev[mm] , oimv[mm] , pflow[mm] , qflow[mm];
 float drev[mm] , dimv[mm] , dif[mm] , rfi[mm] , ifi[mm] , iiflow[mm] ;
 float  meas[mm] , cmeas[mm] ,  d[mm] , rev[mm] , imv[mm] ;
 float prod[mm] , delv[mm] , mdelv[mm] , th[mm][mm] , pl[mm] , ql[mm] ;
 float h[mm][mm] , hr[mm][mm] , rli[mm] , ili[mm] , irflow[mm] ;
 int i, j , k , id , l , r1, s1 ;
 float rsum , isum ;
//**************structure**************************************************
 struct feeder                    //structure specifier
 {
 float res ;
 float reac ;
 float g ;
 float b ;
 }feed_data[mm] ;                 //array of structure
//**************function declaration***********************************
 void read_data( ) ;
 void cal_admit( ) ;
 void invr_covar( )  ;
 void output_read_data( ) ;
 void cal_measurement( ) ;
```

```
  void cal_error( ) ;
  void cal_jacobian( ) ;
  void cal_transpose_jacobian( ) ;
  void cal_h_transpose_r_inverse( ) ;
  void cal_h_transpose_r_inverse_h( ) ;
  void cal_inverse( ) ;
  void cal_h_transpose_r_inverse_error( ) ;
  void cal_delv( ) ;
  void final_voltage( ) ;
  //*************main program starts***************************************
  void main( )
  {
//*********calling of function********************************************
 read_data( ) ;  //reading of number of bus,line,measurement,
                      //state variables & slack bus,line connection,
                      //initial bus voltage,tolerance,measured data,
                      //co-variance element,system data.

   for(i=1 ; i<=nline ; i++)
   {
    meas[i] = pflow[i]/base_kva ;
    meas[i+nline] = qflow[i]/base_kva ;
    meas[i+2*nline] = sqrt(pow((irflow[i]*base_kv/base_kva),2)+
                         pow((iiflow[i]*base_kv/base_kva),2)) ;
   }
  // linknet program for network topology
  for(i=1 ; i<=nbus ; i++)
  list[i] = 0 ;
  for(i=1 ; i<=nline ; i++)
   {
    endb = 2*i ;
    enda = endb-1 ;
    nodea = lfrom[i] ; nodeb = lto[i] ;
    next[enda] = list[nodea] ;
    list[nodea] = enda ;
    ifar[enda] = nodeb ;
    next[endb] = list[nodeb] ;
    list[nodeb] = endb ;
    ifar[endb] = nodea ;
   }

 for(i=1 ;  i<=nbus ;  i++)       //reading of initial bus voltage
```

```
  {
   srev[i] = 1.0 ;
        simv[i] = 0.0 ;
  }
 cal_admit( ) ;              //computation of admittance
 invr_covar( ) ;             //inversion of co-variance matrix
 output_read_data( ) ;          //write for read data
//***********iteration starts*****************************************
 p = 0 ;
 do{
 p = p+1 ;
 id = 0 ;
 cal_measurement( ) ;         //calculated line measurement
 cal_error( ) ;             //calculation of error matrix [z-f(x)]
 cal_jacobian( ) ;            //calculation of Jacobian matrix element

  for(i=1 ; i<=nm ; i++)
  for(j=1 ; j<=ns/2 ; j++)
  h[i][j] = h[i][j+1] ;
  for(i=1 ; i<=nm ; i++)
  for(j = (ns/2+1) ; j<=ns ; j++)
  h[i][j] = h[i][j+2] ;

 cal_transpose_jacobian( ) ;    //Transpose of jacobian matrix
 cal_h_transpose_r_inverse() ;  //Multiplication of h transpose and
                                             // r inverse matrix
 cal_h_transpose_r_inverse_h( ) ; //calulation of product of h transpose,
                                             //r inverse and h
 cal_inverse( ) ;             //calculation of inverse of hrh matrix
 cal_h_transpose_r_inverse_error( ) ; //calculation of product of
                                               //h transpose,r inverse and [z-f(x)]
 cal_delv( ) ;                //calculation ofdelrev and delimv

 for(i=1 ; i<= nbus ; i++)
 {
 if(i!=s)
  {
       rev[i] = srev[i] + delv[i-1] ;
       imv[i] = simv[i] + delv[i+nbus-2] ; .
  }
        else
         {
```

```
            rev[s] = 1.0 ;
            imv[s] = 0.0 ;
           }
   }

  for(j=1; j<=ns; j++)
  mdelv[j] = sqrt(pow(delv[j],2)) ;

   for(i=1; i<=ns; i++)
   {
    if(mdelv[i]>tol)
    id = 1 ;
   }
           if(id==1)
           {
           for(i=1; i<= nbus; i++)
            {
              srev[i] = rev[i] ;
              simv[i] = imv[i] ;
             }
            }
         }

  while(id==1) ;
 final_voltage( ) ;
//**************** end of main  programe********************************
 for(i=1; i<=nbus; i++)
   {
        drev[i]=orev[i]-rev[i]*base_kv ;
        dimv[i]=oimv[i]-imv[i]*base_kv ;
   }

  for(j=1; j<=nbus; j++)
  {
   drev[j] = sqrt(pow(drev[j],2)) ;
   dimv[j] = sqrt(pow(dimv[j],2)) ;
  }

for(i=1; i<=nbus; i++)
{
 if(i!=s)
  {
```

```
    dif[i-1]=drev[i];
    dif[i-2+nbus]=dimv[i];
   }
  }

 float min = 1.00 ;
 for(i=1; i<=ns; i++)
 {
  if(dif[i]<min)
  min = dif[i] ;
 }
 outfile<<"mimmum volltage error = "<<min<<endl;
 outfile<<"********************************************"<<endl ;

 for(i=1; i<=nline; i++)
 {
  rfi[i] = (rev[lfrom[i]]-rev[lto[i]])*feed_data[i].g-
          (imv[lfrom[i]]-imv[lto[i]])*feed_data[i].b ;
  ifi[i] = (imv[lfrom[i]]-imv[lto[i]])*feed_data[i].g+
          (rev[lfrom[i]]-rev[lto[i]])*feed_data[i].b ;
 }
 outfile<<"\n feeder current" ;
 outfile<<endl ;
 outfile <<setw(6)<<"line no"<<setw(20)<<"connected between nodes"
          <<setw(20)<<"Re current"<<setw(20)<<"Im current"<<endl ;
 outfile <<setw(6)<<"------ "<<setw(20)<<"---------"<<setw(20)
          <<"------"<<setw(20)<<"-------"<<endl ;

 for(i=1; i<=nline; i++)
 outfile<<setw(6)<<i<<setw(5)<<lfrom[i]<<setw(5)<<lto[i]
          <<setprecision(6)<<setw(25)<<rfi[i]*base_kva/base_kv
          <<setprecision(6)<<setw(25)<<ifi[i]*base_kva/base_kv<<endl ;
 outfile<<endl ;

 for(i=nbus; i>=2; i--)
 {
 end = list[i] ;
 rsum = 0.0 ; isum = 0.0 ;
 do{
 branch = (end+1)/2 ;
 j = ifar[end] ;
 end = next[end] ;
```

```
 if(j>i)
 {
  rsum = rsum+rfi[branch] ;
  isum = isum+ifi[branch] ;
 }
  else
    {
          rli[i] = rfi[branch]-rsum ;
          ili[i] = ifi[branch]-isum ;
         }
 }
 while(end!=0) ;
 }

outfile<<"\n load current" ;
outfile<<endl ;
outfile <<setw(6)<<"bus no"<<setw(20)<<"Re current"<<setw(20)
         <<"Im current"<<endl ;
outfile <<setw(6)<<"------ "<<setw(20)<<"------"<<setw(20)
         <<"-------"<<endl ;

for(i=1; i<=nbus; i++)
outfile<<setw(6)<<i<<setprecision(6)<<setw(25)
        <<rli[i]*base_kva/base_kv<<setprecision(6)
        <<setw(25)<<ili[i]*base_kva/base_kv<<endl ;
outfile<<endl ;

for(i=1; i<=nbus; i++)
{
 pl[i] = rev[i]*rli[i]+imv[i]*ili[i] ;
 ql[i] = imv[i]*rli[i]-rev[i]*ili[i] ;
}

outfile<<"\n load " ;
outfile<<endl ;
outfile <<setw(6)<<"bus no"<<setw(20)<<"Re power"<<setw(20)
        <<"Im power"<<endl ;
outfile <<setw(6)<<"------ "<<setw(20)<<"------"<<setw(20)
        <<"-------"<<endl ;

for(i=1; i<=nbus; i++)
outfile<<setw(6)<<i<<setprecision(6)<<setw(25)<<pl[i]*base_kva
```

```
        <<setprecision(6)<<setw(25)<<ql[i]*base_kva<<endl ;
 outfile<<endl ;
 }
 //**************function body*****************************************

void read_data( )         // function for reading system data
{
 int i, j , a , b ;
 float c , d ;

 infile>>a ;   nbus = a ;        // no. of bus
 infile>>a ;   nline = a ;       // no. of line
 infile>>a ;   nm = a ;          // no. of measurements
 infile>>a ;   ns = a ;          // no. of state variables
 infile>>c ;   base_kva = c ;    // system KVA
 infile>>c ;   base_kv = c ;     // system_KV
 base_imp = ((base_kv*base_kv)/base_kva)*1000.0 ;

 for(i=1; i<=nline; i++)              // reading feeder data
  {
   infile>>a>>b>>c>>d ;
   lfrom[i] = a ;
   lto[i] = b ;
   feed_data[i].res = c/base_imp ;       // resistance of feeder
   feed_data[i].reac = d/base_imp ;      // reactance of feeder
 }
 infile>>a ;   s = a ;                  // slack bus no.
 infile>>c;    tol = c ;                 // tolerance

 for(i = 1 ; i <= nline ; i++)           // reading the measurements
   infile>>pflow[i] ;
for(i = 1 ; i <= nline ; i++)
   infile>>qflow[i] ;
for(i = 1 ; i <= nline ; i++)
   infile>>irflow[i] ;
for(i = 1 ; i <= nline ; i++)
   infile>>iiflow[i] ;

  for(i=1 ; i<=nm ; i++)
  infile >> r[i] ;               // reading the co-variance matrix
 for(i=1 ;i<=nbus ;i++)
 infile>>orev[i]>>oimv[i] ;
```

```
}                          // end of `read_data' function
//*************************************************************************
void output_read_data( )           //function for output for read data
{
int i , j ;
outfile<<"output of read data"<<endl ;
outfile<<"number of bus="<<nbus<<endl ;          //no.of buses
outfile<<"number of line="<<nline<<endl ;       //no. of line
outfile<<"number of measurement="<<nm<<endl ;   //no. of measurement
outfile<<"number of state variables="<<ns<<endl ;//no.of state variables
outfile<<"base_kva="<<base_kva<<endl ;          //base kva
outfile<<"base_kv="<<base_kv<<endl<<endl ;      //base kv

outfile<<"line connection"<<endl ;
for(i=1 ; i<=nline ; i++)                              //line connection
outfile<<"line"<<i<<"connected between nodes"
          <<lfrom[i]<<"and"<<lto[i]<<endl ;
 outfile<<endl<<endl;

 outfile<<setw(20)<<"connection"<<setw(24)<<"resistance(p.u)"
          <<setw(16)<<"Reactance(p.u)"<<endl ;
 outfile<<setw(20)<<"----------"<<setw(24)<<"----------"
           <<setw(16)<<"----------"<<endl ;
 outfile<<endl ;
for(i=1; i<=nline; i++)
outfile<<setw(13)<<lfrom[i]<<setw(5)<<lto[i]<<setprecision(4)
          <<setw(21)<<feed_data[i].res<<setprecision(4)  //resistance
          <<setw(20)<<feed_data[i].reac<<endl ;         //reactance

outfile<<endl ;
outfile<<"slack bus no="<<s<<endl ;                //slack bus no.
outfile<<"tolerance="<<setprecision(4)<<tol<<endl ;  //tolerance
outfile<<endl ;

outfile<<"measurement data(p.u)"<<endl ;          //measurement data
for(i = 1; i <= nm; i++)
outfile<<setprecision(4)<<meas[i]<<endl ;

outfile<<"inverse of co-variance matrix diagonal element"<<endl ;
for(i=1 ;i<=nm ;i++)
outfile<<invr[i]<<endl ;
```

```
 outfile<<setw(20)<<"connection"<<setw(24)<<"g(p.u)"<<setw(16)
           <<"b(p.u)"<<endl ;
 outfile<<setw(20)<<"------"<<setw(24)<<"-----"<<setw(16)
           <<"-----"<<endl ;
 for(i=1 ; i<=nline ; i++)
 outfile<<setw(13)<<lfrom[i]<<setw(5)<<lto[i]<<setprecision(4)
           <<setw(21)<<feed_data[i].g<<setprecision(4)<<setw(17)
           <<feed_data[i].b<<endl ;
 }                     // end of `output_read_data' function
 //**********************************************************************
void cal_admit( )             //function for admittance calculation
{
 for(int i=1; i<=nline; i++)
 {
  feed_data[i].g=(feed_data[i].res)/
                  (pow(feed_data[i].res,2)+pow(feed_data[i].reac,2)) ;
  feed_data[i].b=(-feed_data[i].reac)/
                 (pow(feed_data[i].res,2)+pow(feed_data[i].reac,2)) ;
 }
}                                    //end of admittance computation
//**********************************************************************
void invr_covar( )              //inversion of co-vatiance matrix
{
for(i=1 ; i<=nm ; i++)
invr[i] = 1/r[i] ;
}                               //end of co-variance matrix
//**********************************************************************
 void cal_measurement( )        //computation of measurement
 {
 for(int i=1 ; i<=nline ; i++)
 {
 cmeas[i] = (srev[lfrom[i]]-srev[lto[i]])*
             ((srev[lfrom[i]])*(feed_data[i].g)+
             (simv[lfrom[i]])*(feed_data[i].b))
             +(simv[lfrom[i]]-simv[lto[i]])*
             ((simv[lfrom[i]])*(feed_data[i].g)-
             (srev[lfrom[i]])*(feed_data[i].b)) ;

 cmeas[i+nline] = (srev[lfrom[i]]-srev[lto[i]])*
                   ((simv[lfrom[i]])*(feed_data[i].g)-
                   (srev[lfrom[i]])*(feed_data[i].b))-
                   (simv[lfrom[i]]-simv[lto[i]])*
```

```
                ((srev[lfrom[i]])*(feed_data[i].g)+
                (simv[lfrom[i]])*(feed_data[i].b)) ;


 cmeas[i+2*nline] = sqrt(pow(((srev[lfrom[i]]-
           srev[lto[i]])*feed_data[i].g-
                    (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].b),2)+
                    pow(((srev[lfrom[i]]-srev[lto[i]])*feed_data[i].b-
                     (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].g),2));
 }
}                            //end of measurement function
//*************************************************************************
void cal_error( )
{
for(int i=1 ; i<=nm ; i++)
d[i] = (meas[i] - cmeas[i]) ;
}                          //end of error function
//*************************************************************************
void cal_jacobian( )            //Jacobian function
{ int i , j ;
 for(i=1; i<=nline; i++)
 {
  for(j=1; j<=nbus; j++)
      {
     if(j==lfrom[i])
             {
                h[i][j] = 2*(srev[lfrom[i]]*feed_data[i].g)-
                        (srev[lto[i]]*feed_data[i].g)+(simv[lto[i]]*
                        feed_data[i].b);
                }
else
    if(j==lto[i])
                 {
                   h[i][j] = -(srev[lfrom[i]]*feed_data[i].g)-
                            (simv[lfrom[i]]*feed_data[i].b);
                   }
else
    {
      h[i][j]=0.0 ;
      }
  }
}
```

```
  for(i=1; i<=nline; i++)
  {
   for(j=1; j<=nbus; j++)
   {
    if(j==lfrom[i])
               {
                   h[i][j+nbus] = 2*(simv[lfrom[i]]*feed_data[i].g)-
                                    (simv[lto[i]]*feed_data[i].g)-
                                    (srev[lto[i]]*feed_data[i].b) ;
               }
 else
     if(j==lto[i])
              {
                        h[i][j+nbus] = -(simv[lfrom[i]]*feed_data[i].g)+
                                         (srev[lfrom[i]]*feed_data[i].b) ;
                   }
 else
     {
       h[i][j+nbus] = 0.0 ;
       }
 }
}

 for(i=1; i<=nline; i++)
 {
 for(j=1; j<=nbus; j++)
 {
 if(j==lfrom[i])
 {
      h[i+nline][j] = -2*(srev[lfrom[i]]*feed_data[i].b)+
                         (srev[lto[i]]*feed_data[i].b)+(simv[lto[i]]*
                         feed_data[i].g) ;
  }
       else
     if(j==lto[i])
          {
           h[i+nline][j] = -(simv[lfrom[i]]*feed_data[i].g)+
                               (srev[lfrom[i]]*feed_data[i].b) ;
                   }
        else
         {
          h[i+nline][j] = 0.0 ;
```

```
          }
       }
  }

  for(i=1; i<=nline; i++)
  {
   for(j=1; j<=nbus; j++)
   {
   if(j==lfrom[i])
   {
       h[i+nline][j+nbus] = -2*(simv[lfrom[i]]*feed_data[i].b)+
                               (simv[lto[i]]*feed_data[i].b)-
                               (srev[lto[i]]*feed_data[i].g) ;
   }
   else
      if(j==lto[i])
        {
           h[i+nline][j+nbus] = ((simv[lfrom[i]]*feed_data[i].b)+
                               (srev[lfrom[i]]*feed_data[i].g)) ;
        }
   else
   {
       h[i+nline][j+nbus] = 0.0 ;
   }
  }
 }
  for(i=1;i<=nline;i++)
 {
for(j=1;j<=nbus;j++)
{
if(j==lfrom[i])
 {
h[i+2*nline][j] = (srev[lfrom[i]]-srev[lto[i]])*
                  (pow(feed_data[i].g,2)+pow(feed_data[i].b,2))/
                  (sqrt(pow(((srev[lfrom[i]]-
          srev[lto[i]])*feed_data[i].g-
                  (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].b),2)+
                   pow(((srev[lfrom[i]]-srev[lto[i]])*feed_data[i].b-
                  (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].g),2))) ;
 }

else
```

```
   if(j==lto[i])
    {
  h[i+2*nline][j] = -(srev[lfrom[i]]-srev[lto[i]])*
                          (pow(feed_data[i].g,2)+pow(feed_data[i].b,2))/
                         (sqrt(pow(((srev[lfrom[i]]-
              srev[lto[i]])*feed_data[i].g-
                         (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].b),2)+
                         pow(((srev[lfrom[i]]-srev[lto[i]])*feed_data[i].b-
                         (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].g),2))) ;
     }
   else
        {
        h[i+2*nline][j]=0.0 ;
    }
  }
 }

for(i=1;i<=nline;i++)
{
 for(j=1;j<=nbus;j++)
 {
  if(j==lfrom[i])
   {
  h[i+2*nline][j+nbus]= (simv[lfrom[i]]-simv[lto[i]])*
                              (pow(feed_data[i].g,2)+pow(feed_data[i].b,2))/
                        (sqrt(pow(((srev[lfrom[i]]-
              srev[lto[i]])*feed_data[i].g-
                        (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].b),2)+
                         pow(((srev[lfrom[i]]-srev[lto[i]])*feed_data[i].b-
                        (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].g),2))) ;
  }
else
 if(j==lto[i])
 {
 h[i+2*nline][j+nbus] = -(simv[lfrom[i]]-simv[lto[i]])*
                            (pow(feed_data[i].g,2)+pow(feed_data[i].b,2))/
                       (sqrt(pow(((srev[lfrom[i]]-
             srev[lto[i]])*feed_data[i].g-
                       (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].b),2)+
                        pow(((srev[lfrom[i]]-srev[lto[i]])*feed_data[i].b-
                       (simv[lfrom[i]]-simv[lto[i]])*feed_data[i].g),2))) ;
}
```

```
  else
  {
    h[i+2*nline][j+nbus] = 0.0;
  }
 }
}
 }                               //end of jacobian function
//************************************************************************
  void cal_transpose_jacobian( )
  {
   for( i=1; i<=nm; i++)
   {
    for( j=1; j<=ns; j++)
     {
          th[j][i] = h[i][j] ;
     }
    }
  }                         //end of transpose of jacobian matrix
//************************************************************************
  void cal_h_transpose_r_inverse( )
  {
  . for( j=1; j<=ns; j++)
          {
           for( k=1; k<=nm; k++)
            {
            hr[j][k] = (th[j][k]*invr[k]) ;
            }
          }
   }                  // end of h_transpose_r_inverse matrix
//************************************************************************
void cal_h_transpose_r_inverse_h( )
{
  for(j=1; j<=ns; j++)
   {
        for(k=1; k<=ns; k++)
         {
          th[j][k] = 0.0 ;
           for( i=1; i<=nm; i++)
            {
             th[j][k] += hr[j][i]*h[i][k] ;
            }
         }
```

```
   }
  }                          //end ofcompotation of product of
                                           // h transpose, r inverse and h matrix
//****************************************************************************
  void cal_inverse( )              //function for inversion
  {
   //order of the matrix is ns(no. of state variables)
              for(i=1; i<=ns; i++)
                     {
                     for(j=1; j<=ns; j++)
                            {
                            for(k=1; k<=ns; k++)
                                   {
                                   if((j!=i)&&(k!=i))
                                   {
                            th[j][k] -= th[j][i]*th[i][k]/th[i][i] ;
                                   }
                                   }
                            }
                            th[i][i] = -(1.0/th[i][i]) ;
                            for(l=1; l<=ns; l++)
                                   {
                                          if(l==i) continue ;
                                          th[l][i] *= th[i][i] ;
                                          th[i][l] *= th[i][i] ;
                                   }
                     }
                     for(r1=1; r1<=ns; r1++)
                     {
                     for(s1=1; s1<=ns; s1++)
                     {
                     th[r1][s1]= -th[r1][s1] ;
                     }
                     }
         }                   //end of inverse function
//*************************************************************************
 void cal_h_transpose_r_inverse_error( )
 {
  for( j=1; j<=ns; j++)
   {
       prod[j] = 0.0 ;
       for(int i=1; i<=nm; i++)
```

```
            {
              prod[j] += hr[j][i]*d[i] ;
            }
      }
    }                       // end of product function
  //***********************************************************************
   void cal_delv( )
   {
    for( j=1; j<=ns; j++)
     {
      delv[j] = 0.0 ;
      for( k=1; k<=ns; k++)
           {
            delv[j] += th[j][k]*prod[k] ;
           }
    }
   }                         //end of delv function
//***********************************************************************
  void final_voltage( )
  {
  outfile<<"\n FINAL VOLTAGE" ;
  outfile<<endl ;
  outfile <<setw(6)<<"Bus No."<<setw(15)<<"Re Vol"<<setw(15)
            <<"Im Vol"<<endl ;
 outfile <<setw(6)<<"------ "<<setw(15)<<"------"<<setw(15)
 <<"-------"<<endl ;
for( i=1;i<=nbus;i++)
outfile<<setw(6)<<i<<setprecision(6)<<setw(15)<<rev[i]*base_kv<<
setprecision(6)<<setw(20)<<imv[i]*base_kv<<endl ;
outfile<<endl ;
outfile<<"number of iteration = "<<p<<endl ;
outfile<<"**************************************************"<<endl;
}                            //end of output function
//*********************************END***********************************
```

WATER RESOURCES DEVELOPMENT TRAINING CENTRE
INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE

NO.WR/D-2/2001/ 5017 Dated: Dec.15,2001

NOTICE

The following Viva-Voce examination of M.E.(WRD) 45th Batch 2000-2001 will take place on 18.12.2001.

Name : Mr. Biswarup Sadhukhan

Internal Guide : 1. Prof. Devadutta Das
2. Dr. B.Das.

Topic of the Dissertation : State Estimation of Power Distribution System

Date and Time : 18.12.2001 at 10.00 A.M.

Place : Seminar Room, WRDTC

Asstt. Supdt. (~~Estt.~~)

Copy to:

1. External Examiner Dr. J. D. Sharma, Prof. E.E. Deptt. I.I.T. Roorkee
2. Prof. & Head , WRDTC
3. Chairman, Academic Programme Committee
4. All faculty members, WRDTC
5. Prof. Devidutta Das, Professor & Head, WRDTC
6. Dr. B.Das, Professor, Electrical Engg. Deptt., UOR
7. Asstt.Registrar (Acd.), IIT Roorkee
8. Mr. S.B.Sharma, SLT, WRDTC
9. Mr. Chamkaur Singh Addl. L.T., WRDTC
10. Mr.Biswarup Sadhukhan, Trainee Officer, 45th WRD batch.

Asstt. Supdt. (Estt.)