State estimation and bad data detection in electrical power system

Bernard C. Clewer

Abstract

Author Bernard C. Clewer

Title State estimation and bad data detection in electrical power systems

The thesis studies the subjects of bad data detection and state estimation in electrical power systems which are the processes whereby voltage, power flow and switch status measurements gathered continuously in real-time are used in conjunction with a model of the system to calculate the voltage levels at every node in the system. Traditionally the state estimation process requires two stages. The first stage is the pre-processing of the measurements by a bad data detector in an attempt to remove all the measurements which are grossly in error. The second is the calculation of the voltage levels by a state estimator from the remaining measurements which are likely to contain small random errors.

Conventional state estimation algorithms are very sensitive to measurement errors, especially switch status errors, and unfortunately it is not possible to ensure that all the measurement errors are removed by the bad data detector. The thesis presents a new algorithm for state estimation utilising linear programming which is able to function in the presence of not only bad analogue measurements but also switch status measurement errors, thus removing the need for a bad data detector. The proposed method of state estimation is also able to include in its model of the system the individual busbars and bus-couplers within a substation. This feature enables the state estimation algorithm to process and provide additional network information thus leading to a more useful and reliable data base.

STATE ESTIMATION AND BAD DATA DETECTION IN ELECTRICAL POWER SYSTEMS

by

Bernard C. Clewer B.Sc. (Dunelm)

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY
to the
DEPARTMENT OF ENGINEERING
at the
UNIVERSITY OF DURHAM
in September 1986.

I dedicate this work to my wife
Ingrid

Contents

List of figures

List of tables

Acknowledgements

I would like to acknowledge the financial support of Advanced Computer Services Ltd., Ferranti Computer Systems Ltd. and the Science and Engineering Research Council.

I express my thanks to all the members of the Department of Engineering at Durham University but especially to my supervisor Professor M.J.H. Sterling and to Dr. M.R. Irving for their advice and guidance throughout my research.

Finally I would like to thank Mrs. E. Barlow for her patience in typing the thesis.

Declaration

None of the material contained in the thesis has been previously submitted by me for a degree in this or any other University and the work represents my own original contribution.

Chapter 1
Introduction

1.1 An introduction to Electrical Power Systems

The primary function of an electrical power system is to provide a secure and reliable source of electricity to the consumer. The term consumer refers to both the domestic user who will use in the order of a few tens of kilo Watts of power to an industrial user such as a steel works who may use several mega Watts of power.

The electricity is supplied to the domestic user as an alternating current with a sinusodial voltage wave form. The voltage level is measured with respect to a neutral line and its value varies from country to country but is usually in the range of 120-240 Volts. Industrial users are usually supplied by a three phase supply which consists of three alternating currents with a phase shift of 120 degrees between each of the voltage wave forms. The voltage level is measured between two of the phases and it can be as high as several tens of kilo Volts depending on the user's requirements. It is more economical in terms of capital expenditure to construct a power system as a three phase supply and it is hence only split into three single phases just before it reaches the domestic user. The load imposed on the system by the domestic users is divided evenly amongst the three phases thus it is sufficient for the power system operator to consider only one of the phases of the system and it is usual for diagrams and displays to detail only one phase.

The process of supplying the consumer with electricity can be divided into three distinct functions, namely generation, transmission and distribution. A brief discussion of the role of each functions follows, illustrated by references to both the Central Electricity Generating Board (CEGB) which supplies England and Wales and also to other power system companies where relevant.

Electricity generation involves the conversion of a primary energy source into electrical energy. The most common primary energy sources are fossil fuels, namely coal and oil and in the CEGB this accounts for approximately 85% of the requirements, the remainder being met by nuclear (approximately 12.5%), gas turbine, diesel and hydro-electric generating equipment. The use of fossil and nuclear fuels produces heat which is converted into the rotational energy of the generators by boiling water to create high pressure steam which is then passed through a turbine. Similarly, gas turbine generators pass hot burning gases through a turbine while hydro electric power stations pass water through a turbine, the pressure required to force the water through the

turbine being generated by storing water in a reservoir situated above the turbine. Diesel generators differ from the others in that a conventional internal combustion engine is used to convert the diesel fuel into rotational energy.

The rotational energy generated by utilising a primary energy source is transformed into electrical energy by a generator which exploits the electro-magnetic interactions between a magnetic field and a moving conductor. Modern steam turbine generators usually have a terminal voltage of 23.5kV and a power output of up to 660MW. A large power station may have as many as six generators. The older equipment may operate at lower voltages and ratings. Typical gas turbine generators have a rating in the order of 70MW while hydro electric generators can vary from as little as a few kilowatts to several hundred megawatts.

The CEGB has approximately 130 power stations with a total generating capacity in excess of 55gW, figure 1.1 illustrates the location of the major power stations in the CEGB. The location of a power station is governed by two major factors. Easy transport of the fuel to the power station and the location of a plentiful supply of water for cooling purposes. Thus power stations are usually located on the coast or a large river and in the case of coal-fired stations, near to a coal field.

The large thermal generators whether they be fossil fuel or nuclear units, are usually the most economical to run and thus they are usually run continuously at a fairly steady output level. As would be expected, the output of large generating units cannot change quickly and it may take several hours to synchronise such a generator to the network from a cold start. Gas turbine generators are however expensive to run and are only run for short periods of time, but they can be synchronised to the network within a matter of minutes and are used to meet sharp increases in the load on the system. Pumped storage hydro-electric generating schemes are an alternative method of meeting the sharp increase in the load curve. Pumped storage units are operated as follows. During the periods of low demand electricity is used to pump water from the lower reservoir to an upper reservoir. This utilisation of the power enables some of the large thermal units to remain synchronised with the system during the slack periods. When the load increases the water stored in the upper reservoir is returned to the lower reservoir and the kinetic energy gained in the fall is used to drive a hydro-electric generator.

The control of the generating units is a complex problem which needs to consider the following criteria. The likely load both in the near future (i.e. in the next 30 minutes) and the more distant future (i.e. in the next

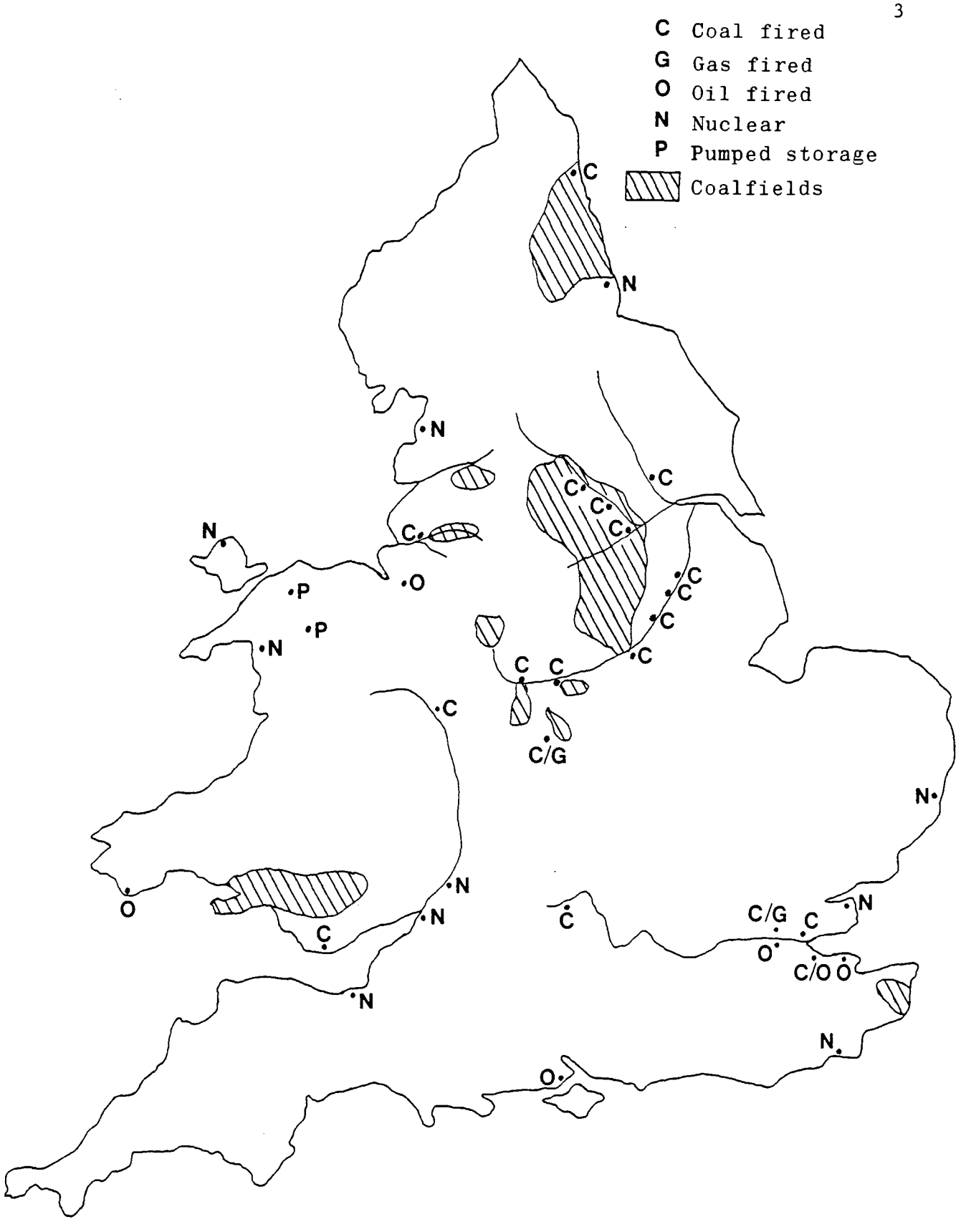


Fig. 1.1: Location of the major power stations in the CEGB.

4-6 hours). The time taken to synchronise a generator if it is not already synchronised. The rate of change of the output of a generator once it is synchronised and in the case of hydro electric schemes the rate of flow of the river or the amount of water stored in the reservoir.

The frequency of the power system provides an easy method of precisely monitoring the balance between the load demand and the power being generated. If the generators are not being supplied with enough energy to supply the load then the rotational kinetic energy of the generators is converted into electrical energy. This happens automatically provided the generator remains synchronised with the network. As the generator loses kinetic energy it slows down, thus the frequency of system falls. Conversely if too much energy is being pumped into the generators, the frequency rises.

The control of the power generation throughout the network is a hierarchical process which commences with a prediction of the load demand at a central control centre and ends with closed loop controllers which regulate the primary energy source supplied to the generators in response to variations in the desired and actual values of frequency and output power. The hierarchical levels in the control sequence include the long term planning of which generators need to be synchronised (unit commitment) based on the long term load forecasts, the short term adjustment of the desired levels of generation (economic dispatch) based on the short term load forecasts and the desired operating frequency and finally the continual adjustment of the generator regulators by the closed loop controllers.

The CEGB divide the unit commitment and economic dispatch problems amongst a national control centre and six area control centres. The national control centre is responsible for determining the overall operating levels throughout the network while the area control centres are responsible for implementing the levels. Figure 1.2 illustrates the location of the control centres in the CEGB.

The role of the transmission section of a power system is to transport the power from the power stations to the load centres of the network. As figure 1.1 illustrates, the power stations are usually located in areas advantageous to the operation of the power station, i.e. near coal fields etc. and not necessarily near large industrial or domestic load centres. The transmission network is usually an inter-connected system of high voltage transmission lines with numerous bulk supply points from which the consumers are supplied. The transmission network is operated at high voltages for economic reasons. High voltages reduce the power flow losses in the transmission lines caused by the line impedance and also reduce the physical dimensions of the conductors

Fig. 1.2: Location of the grid control centres

required to transport a given power flow. The transmission network of the CEGB is operated at 400kV and 275kV and similar operating levels are used both in Europe and America. Trials are in progress to evaluate the use of even higher voltages but the use of higher voltages leads to problems with insulator breakdown. The transmission or 'supergrid' network of the CEGB consists of approximately 10,000km of 400kV lines and 5,000km of 275kV lines, figure 1.3 illustrates the layout of the supergrid network.

Very large industrial complexes may be supplied directly from a bulk supply point of the supergrid network or alternatively concumers are supplied by a distribution network.

The distribution networks are usually supplied by several bulk supply points from the transmission network. The voltage level is transformed to a level of a few 10'skV. The distribution network in England and Wales is maintained by 12 area boards and typically operates at 11kV and 33kV although it is reinforced by a few 132kV transmission lines. The design of the distribution network can be essentially based on two types. A radical type of network whereby the lines radiate outwards from the bulk supply points or a mesh type of network whereby the lines are connected to the supply points at both ends instead of at just one. The domestic user is supplied by a single phase of the distribution network which is typically at 120V or 240V above ground potential while industrial consumers may be supplied with a three phase supply at a voltage level which is suitable for their requirements.

The load demand created by the consumers varies enormously from day time to night time and from season to season. The load demand is also affected by factors such as the cloud covering and television program schedules. Figure 1.4 illustrates some typical CEGB 24 hour load curves for summer and winter and the top half of figure 1.5 compares the load demand of a typical bank holiday to that of the Royal Wedding in 1981. The bottom half of figure 1.5 depicts the system frequency in the CEGB network during the Royal Wedding and illustrates the changes in the system frequency which arise if the generation does not match the load demand. Ideally the system frequency in the UK should be steady at 50Hz while in other countries a value of 60Hz is sometimes used. The role of forecasting the load demand is a difficult task and is often a matter of judgement based upon the load demand for similar days in the past, weather forecasts and the T.V. Times.

The company responsible for the operation of a power system usually has a set of guidelines specifying the operating conditions of the system. These guidelines will specify the required level of operation together with a list of tolerances which should be adhered to under normal and emergency operating

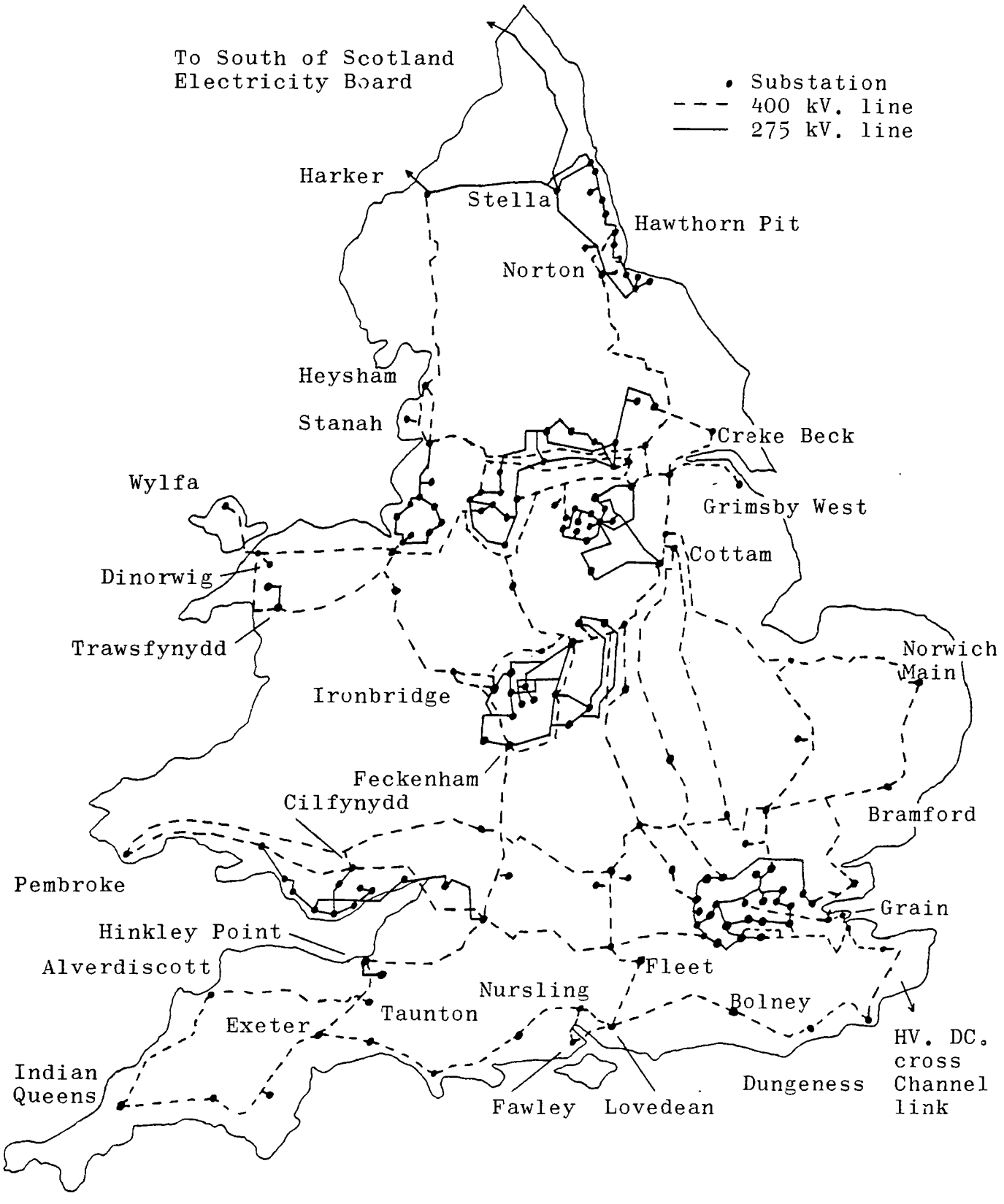
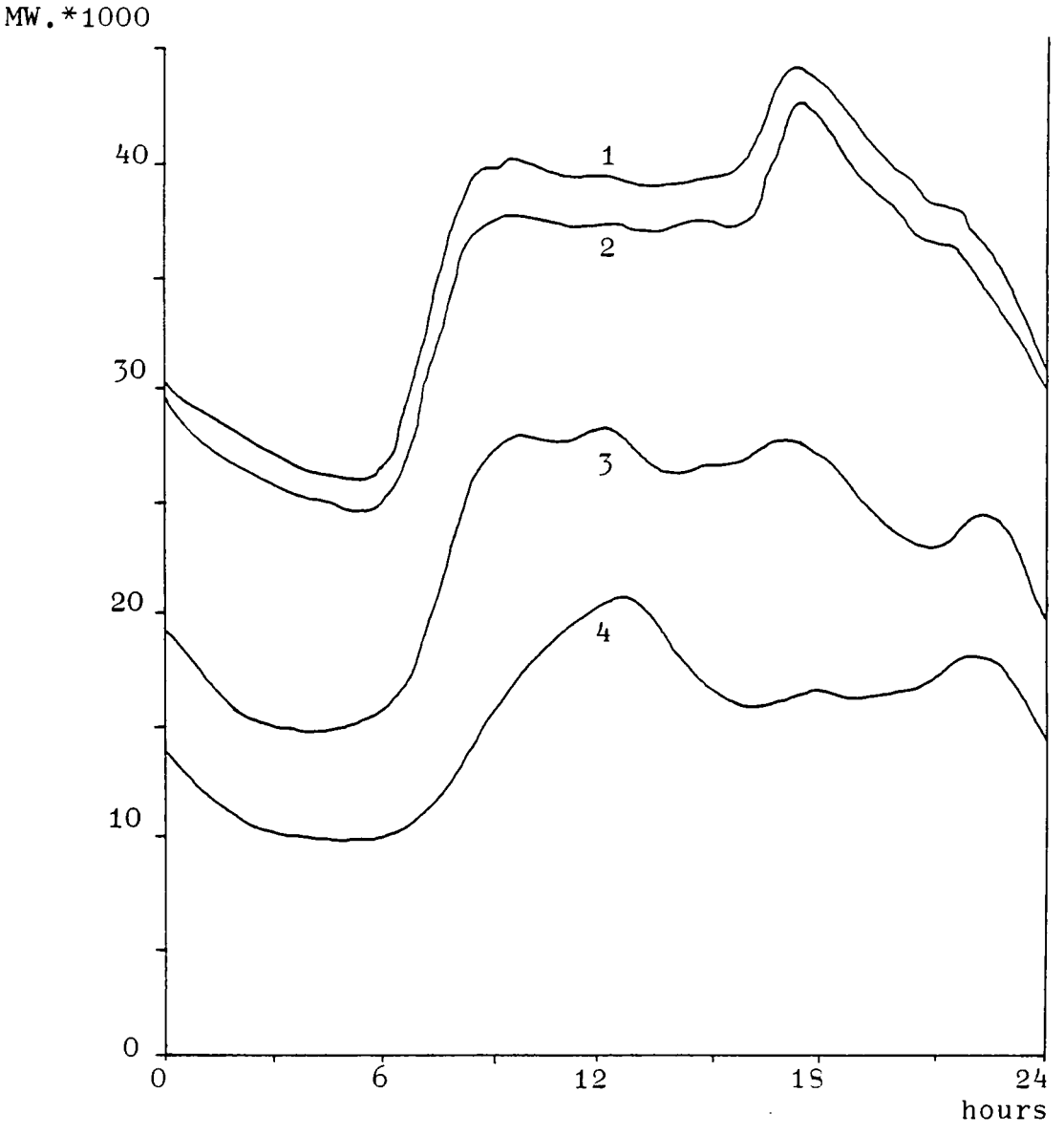


Fig. 1.3: CEGB. Supergrid system



1. Winter peak demand day
2. Typical Winter day
3. Typical Summer day
4. Summer Sunday minimum demand

Fig. 1.4: Comparison of Summer and Winter demands on the CEGB.

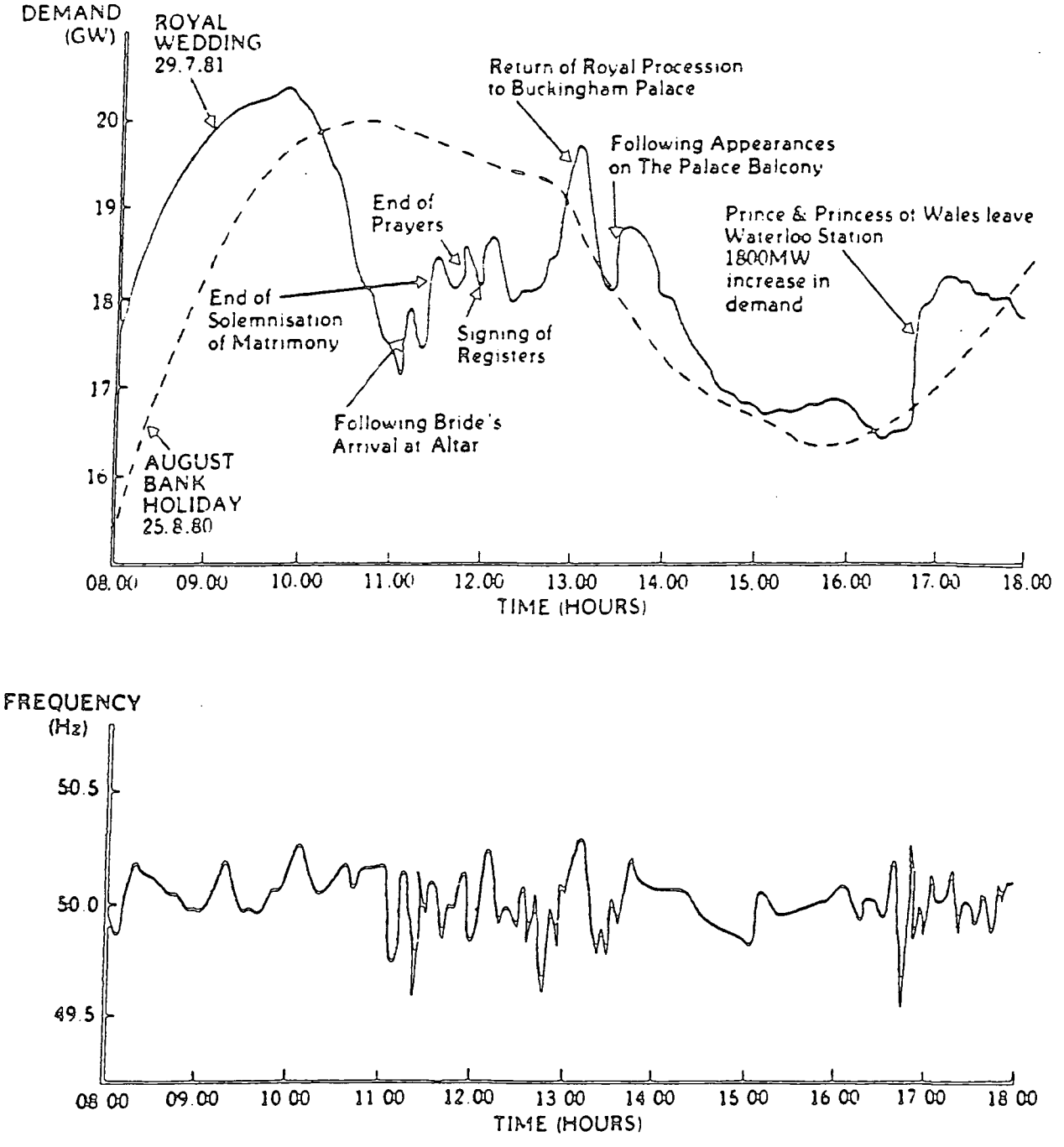


Fig. 1.5: Illustration of the effects of televised events on the system demand and frequency

conditions. The pressures on the power system operators are forever increasing as the consumers expect and sometimes demand a more reliable supply, local and national governing bodies require stricter control of pollution and nature conservation and economic pressures force the company to reduce capital expenditure and operating overheads.

In order to be able to meet the ever more stringent operating conditions with the ever decreasing levels of equipment redundancy the operator requires up to date and accurate information about the state of the entire network. The function of supplying the operator with this information and additional information on the security of system derived by processing the raw measurements can be provided by an on-line digital computer. The computer continuously receives measurements of voltage levels and power flows from selected points in the network. The measurements are then validated to remove those which are in error and values then calculated for all the unmeasured points. Additionally the computer is able to notify the operator of any alarms which already exist and perform calculations to advise the operator if an emergency condition would arise from the loss of any single piece of equipment. The computer may also perform calculations to determine the generator output levels required to satisfy the current load in the most economical way.

The installation and development of a computerised control centre is economically justified by the company by the reduction of both the capital expenditure which would otherwise be required to provide additional equipment needed to maintain a secure supply if such a detailed knowledge of the state of the system were not available, by the reduction of manning levels required to operate the system and by the reduction in fuel costs gained by operating the system more efficiently.

The thesis is concerned with the subject of validating the raw measurements and calculating values for unmeasured points in the system, these functions are usually referred to as bad data detection and state estimation. The following section describes in more detail the contents of the thesis.

1.2 The presentation of the thesis

The thesis is divided into a total of nine chapters and the following paragraphs outline the contents of the remaining chapters.

The second chapter presents a survey of previously published methods of state estimation. The first section of chapter 2 describes the load flow problem and its solution from which the state estimation algorithms were

originally developed. A brief mention of algorithms developed from the state estimation algorithm is also included. The second section of chapter 2 discusses state estimation and the discussion is divided into three sub-sections. The first sub-section discusses the requirements of a state estimator, the second sub-section develops the load flow algorithm into the conventional Newton Raphson least squares state estimation algorithm. The third sub-section discuss the development of the theory into a practical algorithm for use on a computer and continues with a survey of alternative methods of state estimation, including algorithms essentially based on the Newton Raphson least squares method and algorithms which differ from the Newton Raphson method in either the method of solving the state estimation problem or the method of solving the linearised network equations. The final section of chapter 2 discusses methods of bad data detection and possible methods of removing its effects on the state estimates. This third section is further divided into two sub-sections, the first discusses bad data detection methods which are applied to the new measurements before state estimation takes place, the second section discusses methods which may be applied either after or during the state estimation process. This last sub-section also includes a combined method of state estimation and bad data detection usually referred to as bad data suppression.

The third chapter describes the computer simulation of the electrical test networks used to evaluate the performance of the state estimation and bad data detection algorithms discussed in the thesis. The chapter is divided into the following five sections. The first section describes the layout of the major components within a substation and details the electrical connections and switchgear used to inter-connect the components. Section 2 describes how the major components within a sub-section are modelled in computer programs and section 3 details the mathematical equations used in both the simulation program and the state estimation/bad data detection programs. The fourth section outlines the test networks used during the development and testing of the programs and the final section describes the operation of the program used to simulate the behaviour of an electrtical power system.

Chapter 4 presents the theory behind the method of state estimation proposed in the thesis. The proposed algorithm was developed from an existing algorithm named substation data validation which was able to validate the power flows within a substation or group of substations. Thus a description of the theory and implementation of the substation data validation algorithm is presented along with some results illustrating the errors in the estimated power flows when used on a network which included both gross measurement

errors and transmission lines modelled in such a way as to show a voltage drop across the terminating nodes. The second half of chapter 4 details the development of the substation data validation algorithm into a novel state estimation algorithm which is able to operate in the presence of gross measurement errors and invalid switch status measurements. The algorithm is able to include the individual busbars and bus-couplers in its model of the system and is thus able to process and provide additional information on the state of the system. This feature will result in a more reliable data base which contains additional information which may be of use to both the power system operator and other power system analysis programs. The proposed state estimation algorithm also has the following unique feature. The state estimation problem is decomposed into four independent linear sub-problems which are solved in a cyclic, iterative fasion until convergence is achieved. As a consequence of dividing the state estimation problem into four sub-problems, the algorithm proposed in the thesis is referred to as the four stage decomposed state estimation algorithm.

Chapter 5 continues by describing the implementation of the proposed algorithm and details two alternative methods of solving the over-determined set of linear equations formed in each of the four sub-estimation problems. The two methods are a least squares method, which may be used as a weighted least squares method if required and a linear programming method based on the Revised Simplex algorithm. The chapter lists some typical solution times for the linear programming method and presents some results of attempts to reduce the CPU times of the linear programming method by either reducing the number of iterations or reducing the number of linear equations needed to implement the state estimation algorithm.

The sixth chapter presents the results of the evaluation of the proposed four stage decomposed state estimation algorithm on the four test networks under different operating conditions. The performance of the algorithm is compared with that of the conventional Newton Raphson based least squares state estimation algorithm. The results include typical solution times for each of the four test networks and a discussion of the effects on the state estimates of adding noise or gross errors to the measurements. The chapter also describes the results when the least squares method of solving an over-determined set of linear equations replaced the linear programming method and the ability of the algorithm to estimate the power flows in the bus-couplers.

Chapters 7 and 8 describe two different approaches to try and improve the solution times of the four stage decomposed state estimation algorithm

proposed in the thesis. Chapter 7 formulates the active and reactive sub-estimation problems in terms of a network flow problem. The solution of a network flow problem is a form of linear programming which mathematically exploits the structure of the matrices used to describe the problem. The chapter introduces the subject of network flow problems, describes a method of solving the problem, the adaptation of the active and reactive sub-estimation problems to a network flow problem and finally presents the results of a program implementing the algorithm.

Chapter 8 describes an approach to distribute the four stage decomposed state estimation algorithm over two or more processors operating in parallel. The network is divided into two or more areas and the states of each area are estimated separately in conjunction with a limited amount of information from the adjacent areas. An introduction to multi-area state estimation is presented followed by a description of the adaptation and implementation of the four stage decomposed state estimation algorithm to a multi-area form. The chapter conludes with a discussion of the results from the program implementing the multi-area algorithm.

The final chapter summarises the conclusions to be drawn from the research and suggests possible paths to follow to improve the performance of the four stage decomposed state estimation algorithm.

The appendices contain additional information not directly concerned with the algorithm proposed in the thesis but which may be of interest to the reader. Appendix 1 lists the network parameters such as transmission line resistance not shown on the figures in chapter 3. Appendix 2 details the mathematics behind the modelling of a generator, the simultaneous solution of the resultant generator differential equations with the non-linear network equations using implicit trapezoidal integration and finally the linearisation of the subsequent non-linear problem using partial differentiation to create a Jacobian matrix. Appendix 3 lists the steady state voltages and power flows at the start of a simulation run for the three larger test networks, the values for the five substation test network are listed in table 6.1. Appendix 4 details the mathematics of the Revised Simplex method and the least squares method used in the solution of the linear equations formed by the four stage decomposed state estimation algorithm. Appendix 5 lists a typical set of noisy measurement values stored on a disc file in order to repeatedly reproduce a test environment. The final appendix lists some timing results of a brief trial into the effects of reducing the size of the linear programming problem by slightly modifying the formulation of the sub-estimation problems.

Chapter 2
State estimation and bad data detection

This chapter presents a survey of state estimation and bad data detection techniques which are of both theoretical and practical interest. A vast amount of work in these areas has already been published and it is intended only to present the basic concepts of state estimation and bad data techniques and to refer the reader to the relevant literature for further details on any specific points.

The chapter is divided into three main sections. The first discusses the load flow problem and its solution, upon which the state estimation techniques are based. A brief discussion of other on-line analysis programs commonly used in power system control centres is also included. The second section discusses state estimation in greater detail and the third section considers bad data detection and correction techniques.

2.1 History

The scale and complexity of the non-linear equations describing the behaviour of a power system precludes the analysis of all but the smallest of networks by hand calculations.

Prior to, and for some time after the advent of digital computers, moderate sized networks could be studied using an analogue device known as a network analyser. However the construction of such a device prohibited the analysis of some of the networks found in the United States of America which consist of over 2000 nodes and 4000 circuits. The first digital load flow solution techniques were published in 1956[20,52,141] and subsequently a vast amount of information has been published on related topics.

A brief review of the different methods of obtaining a load flow solution has been published by Stott[129] and a large bibliography on the subject has been included in a thesis by Al-Shakarchi[2].

The solution of the load flow problem enabled a power system operator to specify the load and generation pattern throughout the network and obtain the nodal voltages and line flows corresponding to that operating condition. As the load flow solution forms the basis from which not only state estimation but several other power system analysis algorithms were developed descriptions of the load flow problem and two of the most commonly used solution methods are presented below.

2.1.1 The load flow problem

The load flow problem arises from the need of a power system operator to know the required operating conditions of the system to meet a given load demand. Given the total power injection at n-1 of the n nodes of the network and the voltage magnitude and phase angle of the n^{th} node then it is possible to calculate the voltage levels of the first n-1 nodes and the total power injection of the n^{th} node. The n^{th} node is usually referred to as the slack bus since the total power injection at the slack bus is determined during the load flow solution. The resulting injection is such that the sum of the injections at all the nodes plus the transmission line losses is zero. The following points should be noted. When considering the load flow problem the term node and bus are synonymous and refer to a point in the power system to which two or more power system elements (eg transmission lines, generators etc.) are connected. The voltage magnitudes are evaluated with respect to ground which is assumed to have a voltage magnitude of zero. Strictly speaking the ground is an additional node and it is possible to evaluate the voltage magnitudes with respect to any node. However this unnecessarily complicates the problem and is of no practical use. The voltage phase angles are evaluated with respect to the slack bus which is usually assigned an arbitrary value of zero, again the voltage phase angles could be evaluated with respect to any node. Each node in the load flow problem is usually categorised into one of the following two types. A PQ bus is one at which the total power injection is known, a PV bus is one at which the active power injection is known and the voltage magnitude is maintained at a fixed value by adjusting the reactive power injection. Thus all the nodes which do not have any generation or reactive power compensation are PQ buses while the remaining may be classed as either depending on the network conditions being analysed. The slackbus is a PV bus at which the active power injection is initially unknown.

Once the load flow solution has been obtained the line flows may be calculated from the nodal voltages and the line parameters if they are required. The following sections outline two of the most commonly used methods of solving the load flow problem. However there are numerous variations and alternative frames of reference to these methods and the reader is referred to the literature for further details.[2,69,124,125,129]

It should be noted that both the methods outlined require the formation of the nodal (or bus) admittance matrix. This matrix is formed by considering the admittance values of all the transmission lines and the shunt admittances

between the nodes and ground. The bus admittance matrix, Y is square and its dimension is equal to the number of nodes. An off diagonal element of the matrix, y_{ij} is equal to the negative of the line admittance if a transmission line connects nodes i and j, it is equal to zero otherwise. The diagnonal elements, y_{ii} are evaluated by summing the admittance value of all the lines connected to node i together with all the shunt admittances connected to node i. The term shunt admittance includes both the shunt admittances used to represent the line charging effects of a transmission line and those of genuine shunt compensators used for voltage control purposes. The book published by Stagg and El-Abihad[124] provides a detailed mathematical description of the formation of an admittance matrix and the solution of the load flow problem using several different techniques.

2.1.2 Gause Seidel load flow solution

The Gauss Seidel load flow solution is an iterative process in which the bus voltage at every node is evaluated from the total current flowing at the node, (the bus current), together with the bus voltages at all the other nodes and the nodal admittance matrix. The new set of bus voltages may then be used to calculate the resulting power injection at every node. If the difference between the calculated power injection and the specified power injection, known as the power mismatch, or the bus mismatch, is greater than a desired tolerance, then the solution has not converged and a new set of bus currents are evaluated from the updated bus voltages and the specified power injections. The iterative process is usually started by assuming a flat voltage profile, ie the voltage magnitude is 1.0 per unit at every bus except the slackbus and the voltage phase angle is zero at all the buses. The initial bus currents can then be evaluated using the specified power injections.

It is more convenient to monitor the change in the bus voltages, however the rate of change in the bus voltages does not correlate well with the power mismatches hence it is often used as an indication of convergence and the power mismatches are evaluated to determine whether the solution has actually converged.

The equations for the Gauss-Seidel load flow solution are derived from the following two basic equations which govern the power flows through a network.

$$S_{in} = P_{in} - j\, Q_{in} = E^{*} I \qquad (2.1)$$

where

S_{in} = vector of complex power injection at every node.

P_{in}/Q_{in} = vectors of active/reactive components of the complex power injection.

E = vector of complex bus voltages.

E^* = complex conjugate of E.

I = vector of complex bus currents.

Given the specified power injections and the initial voltages then equation 2.1 may be rearranged to evaluate the initial bus currents. The nodal admittance matrix may be used to define the bus currents in terms of the bus voltages as

$$I = YE \tag{2.2}$$

where

Y = nodal admittance matrix.

Expanding equation 2.2 gives the bus current for bus p as

$$I_p = \sum_{q=1}^{n} y_{pq}E_q = \tag{2.3}$$

$$y_{pp}E_p + \sum_{\substack{q=1 \\ q\neq p}}^{n} y_{pq}E_q$$

where

n = total number of buses.

Rearranging equation 2.3 defines the bus voltage at bus p as

$$E_p = \frac{1}{y_{pp}}\left[I_p - \sum_{\substack{q=1 \\ q\neq p}}^{n} y_{pq}E_q\right] \tag{2.4}$$

The set of bus voltages given by equation 2.4 may then be used in conjunction with the specified power injections and equation 2.1 to evaluate a new set of bus currents. It should be noted that all of the quantities have a real and imaginery component and the problem is usually separated into two halves, one for the real part and one for the imaginery part. The reader is referred to the references mentioned in section 2.1.1 for further information on the

implementation and modifications to the Gauss-Seidel load flow solution.

2.1.3 The Newton Raphson load flow solution

The polar power mismatch version of the Newton Raphson load flow solution is the most widely used load flow solution.[129] Nowadays the method is often divided into two smaller problems by making use of the weak interaction between active power flow / voltage phase angle sub-system and the reactive power flow / voltage magnitude sub-system. Accurate solutions can be obtained using this method whilst saving on the storage requirements and execution times. The division of the load flow solution method is usually referred to as the decoupled load flow and the term fast decoupled load flow is used if elements of the partial differential equations required by the Newton Raphson method are evaluated once at the start of the solution.[131,132]

The generalised Newton Raphson method is an iterative algorithm for solving a set of n simultaneous non-linear equations in an equal number of independent variables, x_i, i=1,n. Re-arranging each of the non-linear equations so that the right hand side is zero and then writing each as a function of the variables enables the set of non-linear equations to be written in matrix form as

$$F(x) = 0 \tag{2.5}$$

where

x = vector of length n of the independent variables.

The change in the value of each of the non-linear equations arising from a small change in the values of the variables x may be found by taking the partial differential of each equation, thus

$$\Delta F(x) = J \Delta x \tag{2.6}$$

where

J = m*n Jacobian matrix, defined as $J_{ij} = \delta f_i / \delta x_j$.

$\Delta F(x)$ = vector of length n of the changes in the values of the functions.

Δx = vector of length n of the changes in the values of the variables.

In order to determine the solution to equation 2.5 it is necessary to assume an initial approximate set of values for the variables x_i. The values of x_i

are then substituted into equation 2.5 to obtain the initial right hand side values. To obtain a more accurate solution it is necessary to calculate a correction value for each of the variables x_i. The required correction is such that it would reduce the right hand side values of equation 2.5 to zero. The correction value is obtained by solving for Δx_i in the linear set of equations defined by equation 2.6. The values of the left hand side of equation 2.6, $\Delta F(x)$ are set to the initial right hand side values obtained from equation 2.5. The solution of equation 2.6 is obtained by pre-multiplying both sides of the equation by the inverse of the Jacobian matrix J. The correction values are then added to the variables x_i to obtain a more accurate approximation to the solution of equation 2.5. The iterative process of evaluating a correction value is then repeated until the change in the magnitudes of the correction values is below a pre-set tolerance.

The Newton Raphson method is applied to the load flow problem by rearranging the power system performance equations 2.1 and 2.2 to express the complex power injection in terms of the complex bus voltages and the admittance matrix. Thus

$$S_{in} = P_{in} - j\ Q_{in} = E^{*}YE \tag{2.7}$$

The solution to the load flow problem may then be obtained as follows. Assume approximate values for the complex bus voltages, evaluate the mis-match on the power flow injections by calculating the power flow injections corresponding to the initial bus voltages using equation 2.7 and then subtract these values from the specified power flow injections. A correction value for each of the bus voltages is then required to reduce the mis-match to zero. The correction values are calculated by obtaining the partial differentials of equation 2.7. The resulting Jacobian matrix equation is usually written in polar co-ordinates and partitioned as below.

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \tag{2.8}$$

where

$\Delta P/\Delta Q$ = vectors of the active/reactive power flow injection mis-matches.

H/N/M/L = sub-matrices of the Jacobian matrix.

$\Delta V/\Delta\theta$ = vectors of the voltage magnitude/voltage phase angle correction values.

The solution of the above matrix equation is a time consuming process and requires a considerable amount of memory. However the low cost of semi conductor memory and modern virtual memory machines have eliminated the storage problem although much work has been done in the past to reduce both the solution times and the storage requirements of the method. A common approximation applied to the method is to decouple the problem, this enables the values of the Jacobian sub-matrices M and N in equation 2.8 to be neglected. A second approximation used in conjunction with the one above is to evaluate the elements of the Jacobian sub-matrices H and L once at the start of the solution using the initial values of the bus voltages instead of at every iteration using the most recent values of the voltages. The application of these two approximations results in a solution method known as the fast decoupled load flow.[130] The reader is referred to the relevant literature for further information on this and other implementations of the Newton Raphson load flow solution.[2,16,124,125,128,129,130,136]

2.1.4 Power system monitoring algorithms

The basic load flow solution is widely used as an off-line power system analysis tool especially for planning purposes. Enhancements to the basic solution method such as employing sparse matrix techniques, assuming the Jacobian matrix remains constant for two or more iterations etc. allow networks with over 2000 nodes to be solved using moderate computational resources. Since the late fifties the load flow problem has been developed first into a state estimation algorithm and then to an optimal load flow algorithm and a contingency analysis algorithm. Optimal load flow and contingency analysis are both important on-line control functions of a power system control centre and much work has been published on each of the subjects. However they are beyond the scope of this thesis and hence only a brief description of each, with references to the literature, is presented in the following paragraphs. The development of the load flow algorithm into a state estimation algorithm is discussed in the following section.

Optimal load flow,[5,22,58,135] as the name suggests, optimises the load flow solution with respect to a specified criteria. Usually this criteria is the minimisation of the cost of generating the power to supply the present load demand. Inputs to the problem are the present load demand at every node, a set of cost coefficients for all the generators currently synchronised with the system and a set of constraints such as line flow limits, minimum and maximum generator limits and nodal voltage limits. The optimal load flow

algorithm then solves the load flow problem to meet the load demand while minimising the total cost of generating the power. In addition to minimising the cost the algorithm has to consider the constraints placed on the solution. The optimal load flow algorithm differs from the economic dispatch algorithm in that the power flow losses in the transmission lines are calculated and taken into consideration during the optimal load flow solution while in the economic dispatch algorithm the transmission line losses are calculated from the present generator set points using a set of coefficients known as penalty factors. The penalty factors are evaluated by studying a typical load flow solution. However it should be noted that the two names are often used synonymously.

The contingency analysis function[45,57,62,131,132,139] enables the power system operator to perform a 'what if' study on the present network. The algorithm obtains a load flow solution for the present operating conditions of the network, known as the base-case solution and then allows the operator to specify either a single or in some cases a multiple line outages. Using the base-case solution as a starting point the algorithm computes the new operating conditions and informs the operator if any bus voltage limits would be exceeded or if any of the remaining lines would be overloaded. The methods of solving the problem are usually based on a matrix modification technique and the bus admittance matrix is not re-computed from a flat start. The general trend in modern control centres is to automate the process whereby the lines most likely to cause an overload are found by using a fast, approximate d.c. load flow solution and then performing a more detailed study on those lines using the a.c. contingency analysis function.

2.2 State estimation

2.2.1 Requirements

During the sixties techniques were developed which enabled measured values to be made available for power system analysis programs. However the basic load flow solution techniques required as inputs the active and reactive power injections at all but one of the nodes in the system and at the remaining node the voltage magnitude and phase angle. It was not practicable in those days to measure the power injection at very node and even in modern control centres it is unlikely that there is an injection measurement available for every node. Furthermore the basic load flow solution technique could not make use

of any voltage magnitude measurements at other nodes in the network or of any power flow measurements made on any of the transmission lines. Finally the resulting solution did not always agree with the measured voltages and line power flows. This last point arises for two main reasons. Firstly, the measured values telemetered back to the control centre were subject to numerous small errors arising from transducer mis-calibration, skew on the measurements (caused by values arriving simultaneously at the control centre referring to different instances in time) and small random errors introduced by rounding errors etc. in the equipment. Secondly, a few of the telemetered measurement values were subject to large (or gross) errors arising from transducer failure, telemetry failure and errors introduced by voltage spikes on the power supplies etc. The above points thus laid the foundation for the need to develop a method of utilising additional measurements and resolving any discrepancies between the measured values. The technique developed from this requirement is known as state estimation, and it involves a statistical treatment of the measurement values. The mathematics of statistics has been known for many decades and is widely available.[8,31,68,143]

The state variables of a power system are defined as the voltage magnitude and voltage phase angle at every node in the system. Given these values it is then possible to calculate the power flow at either end of a transmission line using the following equation.

$$P_{pq} - j\,Q_{pq} = (V_p)^*(V_p - V_q)y_{pq} + (V_q)^* V_q y'_{pq} \qquad (2.9)$$

where

P_{pq} = active power flow from node p to node q.
Q_{pq} = reactive power flow from node p to node q.
y_{pq} = line admittance.
y'_{pq} = one half of the total line charging shunt admittance.

The power flow from node q to node p (often known as the reverse power flow) is evaluated by interchanging the subscripts p and q in equation 2.9, thus the power flow loss in a line may then be found by subtracting the values of the forward and reverse power flows. The power flow injection at any node is evaluated by summing the power flows of all the lines connected to that node.

The state estimator program requires as input all the available measurements, the line parameters from which the admittance matrix is constructed and a mathematical representation of the equations defining the power flows in terms of the states. The state estimator then performs a

statistical treatment of the measurements to obtain a set of state estimates which are likely to reflect the true state of the network. The theory behind the solution of the state estimation problem is presented in the next section.

2.2.2 Theory

A power system of n nodes is said to have 2n-1 state variables, namely n voltage magnitude values and n-1 voltage phase angle values evaluated with respect to a reference value at the n^{th} node. Thus at least 2n-1 measurements are required to define the 2n-1 states. If these 2n-1 measurements consist of the active and reactive power flow injections at n-1 nodes of the system and the voltage magnitude at the n^{th} node then the problem is reduced to a load flow problem as described in section 2.1.1. If there are more than 2n-1 measurements then a statistical treatment of the measurements is required to obtain values for the 2n-1 state variables.

A series of non-linear equations may be derived from equation 2.9 which relate a measurement vector Z, of length m where m is the number of measurements, to the state vector X, these equations maybe written as

$$Z = h(X) + W \tag{2.10}$$

where

Z = vector of length m containing the measurement values.

X = state vector of length 2n-1.

h = set of non-linear functions.

W = random noise vector of length m representing the measurement noise.

The solution of the state estimation problem requires that the statistical properties of the noise vector are known. It is usually assumed that the distribution of the noise is Gaussian[125] such that the mean or the expectation, E is zero ie

$$E\{W\} = 0 \tag{2.11}$$

where

E = expectation.

The measurements are also assumed to be unbiased and uncorrelated which enables the expectation to be equated to the standard deviation of the noise

on the measurements by using the equation

$$E\{WW^t\} = R \tag{2.12}$$

where

R = m*m co-variance matrix defined as

$$R_{ii} = (\sigma_i)^2, \qquad i = 1, \ldots, m \tag{2.13}$$
$$R_{ij} = 0, \qquad i = 1, \ldots, m,\ j = 1, \ldots, m,\ i \neq j.$$

where

σ_i = standard variation of the noise on the i^{th} measurement.

The values of σ_i are usually found by field trials and they depend on the accuracy of both the measurement transducer and the telemetry equipment.

The state estimation problem is then to calculate a vector of state estimates X' of the state vector X from the noisy set of measurements Z. The performance of the state estimator can be judged by evaluating the estimate co-variance matrix R_e as shown below

$$R_e = E\{(X-X')(X-X')^t\} \tag{2.14}$$

where

R_e = estimate covariance matrix.
X' = vector of state estimates.
X = true state vector.

This maybe achieved in terms of a least squares estimation by minimising a quadratic function based on the difference between the actual measurement vector Z and the measurement vector required to produce the state vector. The latter measurement vector is defined by substituting the state variables X' into the set of non-linear functions h(.). The objective function is then defined by the equation

$$\text{Min. w.r.t. } X' \qquad J = \sum_{i=1}^{m} \left[\frac{(z_i - h_i(x'_i))^2}{\sigma_i^2} \right] \tag{2.15}$$

or in matrix form as

$$\text{Min. w.r.t. } X' \quad J = (Z - h(X'))^t R^{-1} (Z - h(X')) \tag{2.16}$$

where

J = value of the objective function.

The minimisation of the objective function may be found if the following condition is satisfied

$$\frac{\delta J(X')}{\delta X'} = -2H^t(X') R^{-1} [Z - h(X')] = 0 \tag{2.17}$$

where

$H(X')$ = $\frac{\delta h(X')}{\delta X'}$, a m*2n-1 Jacobian matrix.

m = number of measurements.

n = number of nodes.

The equation above represents a set of 2n-1 non-linear equations in terms of the state estimates. The solution may be found any general non-linear optimisation method or more commonly by the iterative Newton Raphson method in a technique similar to the Newton Raphson load flow solution.

The Newton Raphson method for solving equation 2.17 requires an initial approximation to the state vector, X^k where k is the iteration count and the Taylor series expansion to a first order approximation of h(X) which gives

$$h(X) = h(X^k) + H(X^k) \Delta X \tag{2.18}$$

where

$H(X^k)$ = Jacobian of h(X) at k.

Equation 2.18 is substituted into equation 2.17 to give

$$J(X^k) = -2H^t(X^k) R^{-1} [Z - (h(X^k) + H(X^k) \Delta X)] \tag{2.19}$$

which when expanded and rearranged gives

$$H^t(X^k) R^{-1} H(X^k) \Delta X = H^t(X^k) R^{-1} Z - H^t(X^k) R^{-1} h(X^k) \tag{2.20}$$

leading to

$$\Delta X = [H^t(X^k)\ R^{-1}\ H(X^k)]^{-1}\ H^t(X^k)\ R^{-1}\ [Z - h(X^k)] \qquad (2.21)$$

The correction to the state estimates ΔX may be added to the initial approximation X^k to produce a better approximation, ie

$$X^{k+1} = X^k + \Delta X \qquad (2.22)$$

It should be noted that the matrix product

$$H^t(X)\ R^{-1}\ H(X)$$

is often refered to as the gain or information matrix, G(X).

The above method of solution can be shown to satisfy the minimisation criteria of equation 2.16 in the following way: If convergence is achieved as $k \to \infty$ then the value of ΔX tends to zero ie.

$$X = (X^{k+1} - X^k)_{k \to \infty} = \qquad (2.23)$$

$$[H^t(X^\infty)\ R^{-1}\ H(X^\infty)]^{-1}\ H^t(X^\infty)\ R^{-1}\ [Z - h(X^\infty)] = 0$$

Now if $m \geq 2n-1$ and the rank of $H(X^\infty)$ is $2n-1$ then the matrix product $H^t(X^\infty)\ R^{-1}\ H(X^\infty)$ is non-singular and has a conventional inverse. Consequently equation 2.23 must be satisfied by

$$H^t(X^\infty)\ R^{-1}\ [Z - h(X^\infty)] = 0 \qquad (2.24)$$

which then satisfies equation 2.16.

A distinction can be drawn between least squares fitting in which a best fit $\hat{X}$ to the measurements Z is found such that the following expression is a minimum

$$[Z - h(\hat{X})]^t\ W\ [Z - h(\hat{X})]$$

where

W = weighting matrix.

and the least squares estimation in which a best estimate $\hat{X}$ of the true state X is found from the measurements Z. In this case the expected sum of weighted

squares errors is minimised ie.

$$E\{(X - X)^t W (X - X)\}$$

The least squares best fit neglects the fundamental statistics of the measurement process while the least squares estimation obtains an estimate of the true state that is statistically optimal. It can then be shown[120] however that the two methods are identical provided the weighting matrix of the best fit method is taken to be the inverse of the observation co-variance matrix, R and the noise components have zero mean and are uncorrelated.

2.2.3 Development

Prior to the 1970's statistical treatment of measured values had been used in other applications, especially the aero industry.[98,99] Much of the information gained from this work in the aero industry has been published in a text by Jazwinski[68] and was in use in developing the first practical state estimation programs for power system analysis. During the late sixties numerous authors were analysing the requirements of a power system state estimation program. Larson in conjunction with Peschon and then Hajdu published his findings in two separate reports.[79,80] The first report of a power system state estimation program was published by Siroux and Adnet.[118] Their technique was based on the Gauss Seidel solutiom method which required many more measurements than would normally be available on a power system. The estimation algorithm was over-simplified and was highly susceptable to measurement errors. The first reports on the implementations of weighted least squares state estimators appeared in the 1st half of 1970 by Schweppe, Rom and Wildes,[112,113,114] Larson, Tinney, Hajdu, Peschon and Piercy,[81,82] Smith[120] and Stagg, Dopazo, Klitin and Van Slyck.[123]

The papers by Schweppe et al.[112,113,114] illustrate some of the approximations that needed to be made to the Newton Raphson based weighted least squares state estimator to ensure that a solution could be obtained within the constraints of early computers. The assumptions the authors made were as follows. All transmission lines have a high X/R ratio. The voltage magnitude at all nodes is approximately 1 per unit. The voltage phase angle difference across a transmission line is close to zero and finally the active power flow measurements are uncorrelated with the voltage magnitude and reactive power flow measurements which enables the decoupling of the state estimation problem as in the case of the decoupled load flow method.

The method presented by Larson et al.[81,82] had significant computational simplifications compared with the method described by Schweppe et al.. Larson et al. noted the computational burden of inverting the gain matrix, $(H^t(X) R^{-1} H(X))$ in equation 2.21 for large networks. The authors proposed a method whereby a suitable set of measurements was used to obtain a convention load flow solution then each redundant measurement is processed in turn to update the state estimates using the recursive least squares equations.

$$X_k = X_k + W_k(z_k - H_k X_k) \tag{2.25}$$

$$P_k = (I - W_k H_k)P_{k-1} \tag{2.26}$$

$$W_k = P_{k-1}(H_k)^t((\sigma_k)^2 + H_k P_{k-1}(H_k)^t) \tag{2.27}$$

where

X_k = state estimate vector after processing k additional measurements.
P_k = state co-variance matrix corresponding to X_k, note only diagonal elements are stored, $P_{ij} \cong 0$, $i \neq j$.
W_k = weighting vector for measurement k.
z_k = additional measurement number k.
H_{kl} = linearised equation vector for measurement k.
I = unit matrix.
$(\sigma_k)^2$ = noise variance for measurement k.

Stagg et al.[123] compared the performance of a full Newton Raphson weighted least squares method and a method using independent equations. The method of independent equations uses a basic Newton Raphson load flow solution with dummy values substituted for the nodal injections where no measurement exists. Once a load flow solution has been obtained, the sensitivity of the nodal injections to the line flows is evaluated and an equal number of line flow measurements are then used to determine new values for the dummy injection measurements. The load flow solution is then recalculated. The remaining line flow measurements are then used in conjunction with the sensitivity matrix to correct the injection measurements, the load flow solution is then recalculated. This last stage is re-iterated until convergence is obtained.

The state estimation methods described by both Schweppe et al. and Larson et al. calculate an estimate of the true state which is statistically optimal and requires that the weighting matrix used is the inverse of the co-variance matrix of the observation errors. Smith argues in the discussion of his

paper[120] that the time co-variance matrix is never known and that the deviation of the voltage estimates to the true state voltages is of no concern. Smith suggests that a criteria for minimisation should be based on a 'cost' to the power company.

Thus the active power flow measurements near load centres should be weighted heavily because a power company charges a consumer for the quantity of active power it has consumed and hence the power company must accurately know the active power flows near load centres. Consequently active power flow measurements near a generation point are weighted less heavily than those near a consumer load point. A power company does not usually charge for reactive power consumption, except to very large consumers, and hence reactive power flow measurements should be weighted very lightly. This concept has not been widely accepted and the majority of state estimation techniques in current use are based on the weighted least squares method using measurements weights which are in proportion to the inverse of the co-variance of the observation error.[4,41,63,133] It should be noted however that the sensitivity of the state estimates to the inverse of the co-variance matrix is not great[56] and in some cases the diagonal values of the matrix are set to one[9] or in proportion to the inverse of the transducer's full scale reading.[9] (The off diagonal elements are zero in any case).

Since the publication of the first state estimation techniques much work has been done to improve the algorithms and the methods of implementation to enable larger networks to be solved in shorter times and with a higher degree of reliability. Consequently many papers have been published on the subject of state estimation and no attempt will be made here to supply a comprehensive survey of all the papers. However a selection of papers will be referenced to allow the reader to persue some of the aspects of power system state estimation. The development of state estimators can be broadly divided into four categories. Modifications made to the original Newton Raphson based weighted least squares state estimator. Alternative methods of estimating the states of the system. Addition of bad data detection and correction methods to the state estimation algorithm and finally the adaptation of state estimation programs to run in a multiprocessor environment.

The first two developments will be discussed in the following two sub-sections, methods of bad data detection and correction in section 2.2.4 and methods of distributed state estimation in chapter 8. General discussions on the subject of state estimation have been published in several texts[4,9,11,12,13,42,54,111,125,142,144] and the proceedings of the 4th Power System Computation Conference (PSCC) held at Grenoble, France in 1972 contains

numerous papers of varied aspects of state estimation.

2.2.3.1 Modifications to the original method

As in the case of the load flow solution the state estimation process can be divided into 2 smaller sub-estimation problems by making use of the weak interactions between the active power flow/voltage phase angle sub-system and the reactive power flow voltage magnitude sub-system. Schweppe et al.[113] detailed a method of decoupling the state estimation problem in one of their original papers and numerous other authors have subsequently studied the topic.[10,49,61,85,86,103] Sirisona and Brown[116] studied the convergence properties of both the full weighted least squares and the fast decoupled weighted least squares state estimation methods using an eigen-value analysis technique. They concluded that both methods were stable with respect to variations in the measurement values and the initial operating point and that the fast decoupled method would remain stable if the ratio of the transmission line reactance to resistance (X/R ratio) is greater than 4. Decoupled state estimators often have convergence problems if some of the transmission lines have a low X/R ratio and much of the work in this area has been directed towarded solving this problem.[10,53,102,103,104,115,146] It soon became apparent that the stability of the Newton Raphson based weighted least squares state estimator also depended on the ratio of the number of line flow measurements to injection measurements, the higher the ratio, the greater the stability of the state estimator. GU, Clements, Krumpholz and Davis [53] derived a method of evaluating a condition value for a network based on the node-to-branch incidence matrix and a measurement-to-branch incidence matrix. The lower condition value the more stable the state estimator. The authors also proposed a method of transforming the least squares solution method to reduce the condition number and hence increase the stability of the state estimator. The method is based on the linear least squares solution method of Peters and Wilkinson and makes use of Cholesky matrix factorisation as follows. Consider a least squares solution of the overdetermined set of linear equations

$$AX = Z \tag{2.28}$$

The method of Peters and Wilkinson decompose the m*n coefficient matrix A into the matrix product BC where B is a m*n matrix of rank n and C is a n*n non-singular matrix. The transformation requires the solution of the standard

least squares problem defined by the coefficient matrix B and a new variable defined by CX. The true state estimates are then obtained by premultiplying the intermediate solution by C^{-1}. If the coefficient matrix A is factorised into the standard triangular LDU form where L is a m*n unit lower trapezodial matrix, D is a n*n diagonal matrix and U is a n*n unit upper triangular matrix then the matrix B can be represented by LD and the matrix C by U. This results in the following definition of the state variables.

$$X = U^{-1}D^{-1}[L^tL]^{-1}L^t Z \tag{2.29}$$

Cholestry factorisation is used to decompose the matrix L^tL into $L'D'(L')^t$ where L' is a n*n unit lower triangular matrix and D' is a n*n diagonal matrix.

An alternative method of decoupling the state estimation problem has been described by Zhuang and Balasubramanian.[146] Conventional decoupling involves the negligence of the coupling terms between the active power flow/voltage phase angle and the reactive power flow/voltage magnitude sub-systems. The authors proposed a linear transformation of the power flow measurements which is equivalent to applying a small rotation to the Jacobian matrix in the plane of the active and reactive power flows. This method of decoupling does not rely on the system being lightly loaded and the transmission lines having large X/R ratios and thus should exhibit a greater stability in such circumstances.

One of the most comprehensive investigations into methods of simplifying the state estimation was performed by Allemong, Radu and Sasson[4] on behalf of the American Electric Power Service Corporation as part of the development of a state estimator for their new control centre. The authors listed eight simplifying assumptions and compared the performance of 18 different programs which were implemented with different combinations of the eight simplifications. The first method was the full weighted least squares method which contained no simplifications and was used as a benchmark against which the performance of all the variations were compared. The authors recommended the following simplifications:

1. Use a flat voltage profile (V=1.0, θ=0.0) when computing the gain matrix.
2. Decouple the gain matrix.
3. Decouple the Jacobian matrix when computing the gain matrix and neglect the series resistances when computing the terms of the active component of the Jacobian matrix.

4. Decouple the Jacobian matrix when computing the input vector of the least squares problem.
5. Transform the power flow measurements by dividing by the corresponding bus voltage magnitude.

Finally in this section it is worth mentioning that the implementation of the state estimators in general has been significantly influenced by the sparse nature of the matrices used in the estimation process. The computational storage and manipulation of sparse matrices is a topic which has been and still does receive a lot of attention. However the mathematics behind the subject are very complex and not directly relevant to state estimation theory, hence the reader is referred to other texts for further information.[15,18,21,105]

2.2.3.2 Alternative methods of state estimation

Dopazo, Klitin, Stagg and Van Slyck[36,37,39,40] were the first authors to propose an alternative method of state estimation. The authors derived a method of least squares estimation based upon their original work[123] in which they described a method of using only line flow power measurements. The voltage drop across all the transmission lines can be expressed in terms of the complex nodal voltages, which may or may not have a known value. The voltage drop may also be calculated from the line flow measurements provided an assumption is made about the nodal voltage at one end of the line. An over-determined set of linear equations can be formed which is solved using a least squares method to provide values of all the unknown complex bus voltages. An additional advantage of this method is that it removes the problem of the relative weightings of the injection and line flow weightings thus simplifying the choice of measurement weights. The method utilises an information or 'gain' matrix which remains constant even if some measurement values are temporarily lost due to telemetry failure. The reduced measurement set reduces the overall size of the problem and this feature in conjunction with the constant information matrix results in a fast stable solution. The authors also noted that if measurements at both ends of all the lines are available then the data is easy to check for gross errors.

A method of state estimation which differs only slightly from the conventional Newton Raphson based least squares state estimator has been investigated by several authors. Traditionally the states of the system and the variables used in the equations reflecting the behaviour of the network

have been expressed in polar co-ordinates. However it is equally feasible to express the variables in cartesian co-ordinates, this results in the Taylor expansion of the partial differential equations containing terms upto the second order derivatives only. These exact equations enable a solution to be obtained without the need to neglect any high order terms as in the polar representation and thus the method ought to bemore stable on an ill-conditional system. Within a short period of time several independent groups of authors presented papers on both load flow solutions[67] and state estimation solutions[3,103,121] using cartesian co-ordinates. The precise details of the implementation of the methods differ slightly and the convergence of the algorithms is still questionable. A recent paper by Keyhani and Abur[71] compared the full weighted least squares method of state estimation, the fast decoupled method used in the new control centre for the American Electric Service Corporation[4] and a cartesian co-ordinate method. They concluded that further work is required on the cartesian co-ordinate method if it is to be of practical use on large systems.

The conventional Newton Raphson based weighted least squares state estimator obtains a solution to an over-determined set of linear equations which define the measurement values in terms of the state variables. The least squares method of solving the linear equations minimises the weighted sum of the squares of the residuals as defined by equation 2.15 and repeated below.

$$\text{Min. w.r.t. X} \quad J = \sum_{i=1}^{m} \left[\frac{(z_i - h_i(x_i))^2}{\sigma_i^2} \right]$$

However Irving, Owen and Sterling[65] have suggested that the state estimates maybe calculated by minimising a weighted function in terms of the modulii of the residuals. Thus the following function is in effect minimised.

$$\text{Min. w.r.t. X} \quad J = \sum_{i=1}^{m} \left[\frac{|z_i - h_i(x_i)|}{\sigma_i} \right] \tag{2.30}$$

The method may be implemented by appending a pair of slack variables to each of the linear equations and solving the set of equations using a conventional linear programming method with an objective function based on the sum of the weighted magnitude of the slack variables. The set of linear equations are solved repeatedly with the input vector re-evaluated as in the conventional state estimation procedure until the change in the magnitude of the state

variables is below a predefined tolerance. The linear programming method will obtain a solution defined by a linearly independent sub-set of the linear equations. An interpretation of the solution point is that the least erroneous (or least noisy) sub-set of equations is selected to define the state variables and that the remaining equations may or may not contain an input value derived from a measurement value which is in error. Thus the method rejects all measurements which have gross errors and then selects the least noisy set of measurements from the remainder, the solution point is thus unaffected by gross measurement errors. It is suggested that the least noisy set of measurements is sufficiently accurate for the control of a power system.

A similar method to that of Irving et al. has been proposed by Kotiaga and Vidyasager[77] in which a weighted least absolute value method of solving the linearised network equations is used. It should be noted that the least absolute value problem and the linear programming problem are equivalent and one may be expressed in terms of the other, indeed the least absolute value problem is usually solved using a special form of linear programming.[51,107] The paper by Kotiaga et al. formulates the problem in a similar way to the conventional weighted least squares method. The authors claim that their representation is computationally more efficient than that of Irving et al.

The conventional Newton Raphson based least squares estimator minimises a quadratic function, however recently several authors have proposed state estimation methods using non-quadratic functions in order to reduce the effects of bad data. These methods are usually known as bad data suppression and are discussed in greater detail in section 2.3.2.

Prewett, Farmer, Lang and Jervis[100] have put forward an argument that if a meter is functioning correctly then the measurement value is sufficiently accurate for monitoring and control purposes. Thus the state estimation process should primarily be concerned with removing bad data rather than smoothing noisy measurements. The authors propose a state estimation method which is divided into 3 stages. The first uses simple logical checks on the data to remove bad data such as power flow indications on an open circuit and measurements which are beyond a realistic limit. The second stage is a D.C. least squares estimation and the third stage is a full A.C. least squares estimation based on the results of the D.C. estimation.

The authors have developed a technique of evaluating an improvement index for each measurement based on the square root of the difference between the the value of the objective function, obtained when all the measurements are used and the value obtained when the measurement is removed from the set. The

measurement is considered bad if the improvement index is greater than a constant (eg. 0.05 times the full scale reading of the meter). The method uses the technique of matrix modification in computing the improvement indices and maybe applied to both the D.C. and A.C. stages. The authors suggest that provided the magnitude of the difference between the improvement indices any adjacent measurements is less than 0.1 times the larger of the two indices then it is not necessary to resort to an A.C. solution because the D.C. solution is sufficiently accurate to eliminate all the bad data. (A single pass of the A.C. solution may be required to smooth the remaining noisy measurements). In the event that the D.C. estimation is unable to resolve which measurement in a group is bad then the A.C. estimation process is used to analyse the suspect measurements. The overall CPU. time required for the three stages ought to be less than the time used in a blanket processing of all the measurements using the A.C. estimator. This method of state estimation is also supported in a discussion on state estimation presented by Knight.[73]

The various implementation methods of state estimation discussed so far have been static state estimators in which a snap shot of all the measurements is taken and the state estimators are then calculated. This procedure has the following disadvantages. The measurement values do not arrive simulaneously at the control centre, more often than not the measurement transducers are scanned in sequence, thus different measurement values reflect the network at different instances in time. The full processing of all the measurements can be a time consuming process especially for a large network. The static state estimator does not make use of any prior information gained from the previos measurement values or state estimates and gives no information on the future trend of the network.

Two alternative methods of state estimation, namely tracking state estimation and dynamic state estimation have been proposed in an attempt to overcome the disadvantages of static state estimation. In general terms a tracking state estimation extends the theory of static state estimation to the time varying case without explicit definition of the dynamic model of the power system. Dynamic state estimation includes a very much simplified model of the time variation of the network. It would not be computationally feasible to accurately model the dynamic behaviour of a power system even if a suitable model could be derived and both methods make use of the fact that overall the system is changing very slowly.

In its most simplistic form a tracking state estimator is equivalent to a single iteration of the conventional Newton Raphson based weighted least

squares state estimator.[54] The initial starting point for the next iteration is the estimates calculated at the previous time step and the change in the measurement is evaluated from the new measurements and the equivalent measurement values which would give rise to the estimates calculated at the previous time step. It is assumed that the change in the measurement values is small enough to allow the state estimates to be obtained in just one iteration. More complex algorithms may be derived in which an objective function based on both the previous estimates and the present measurements is minimised.[20]

The first account of a tracking state estimator was published by Debs and Larson[33] in 1970 shortly followed by Masiello and Schweppe.[89] Debs and Larson assumed that the states of the network could be evaluated from the estimates at the previous time step plus a disturbance evaluated from the past performance of the network. The past performance of the network is actually used to evaluate the terms of an expression which represents a filter. The filter is applied to a function based on the present measurements and the state estimates at the previous time step to calculate an update which may be added to the previous estimates to produce the estimates for the current time step.

Masiello and Schweppe evaluated a gain matrix similar to the gain matrix in the standard notation of Newton Raphson weighted least squares state estimation. Ideally the gain matrix should be constant and valid for a wide range of operating conditions, hence a suitable matrix was obtained by intuition combined with trial and error. The gain matrix was applied to the state estimates from the previous time step and the present measurements to calculate the new state estimates in a single iteration.

Arafeh and Schinzinger[9] and later Falcao, Cooke and Brameller[44] have reported comparisons on the various methods of implementing a tracking state estimator. Other authors have also published their work on the subject and most recently Kotiuga reported the development of his least absolute value static state estimator[77] into a tracking estimator.[76]

A tracking state estimator has no knowledge of the dynamic behaviour of the network and consequently, like a static state estimator, it always lags behind the actual behaviour of the network. Dynamic state estimation requires a model of the behaviour of the network and is thus able to predict the states of the network. The accuracy of the model governs the ability of the algorithm to predict or forecast the state of the network. The 'dynamic' model used by Debs and Larson[33] was simplified to such an extent that it enabled the previous estimates to be used in conjunction with the latest

measurements to calculate the new state estimates. However it would not produce very reliable results if used to forecast the state of the network. The division between tracking state estimation and dynamic state estimation is thus not clearly defined, but it is obvious that the more accurate (and hence complex) the dynamic model of the system then the greater the reliability of the predicted state estimates. The reader is referred to other publications for a mathematical description of dynamic state estimation.[48,84,111,125]

2.3 Bad data detection and correction

The telemetered data received at a power system control centre will be subject to two basic forms of error as discussed in section 2.2. The Newton Raphson based least squares state estimator is ideal at processing telemetered data which is only subject to small variations, the co-variance of which has been previously determined. However, it became obvious shortly after the testing of the first state estimation programs that telemetered data subject to large errors seriously affected the reliability of the results. Several authors have suggested that the detection and correction of bad data is more important that the smoothing of random noise,[44,73,100] thus a considerable amount of work has been directed towards methods of detecting bad data and then either removing it from the measurement set used as input to the state estimator or substituting the value for a more accurate one which may either be a pseudomeasurement value or one calculated from other data. The methods of bad data detection can be broadly divided into two categories. Firstly methods in which the telemetered measurements are analysed before the state estimator processes the data and secondly those in which the bad data is detected during or after the state estimation process. A survey of publications in the two categories is presented in the following two sub-sections.

2.3.1 Pre state estimation

The data validation which may be performed before state estimation varies from simple logical checks which are usually performed as a matter of course, to complex algorithms which require a fair amount of CPU time. Simple logical checks are often performed by the data acquisition processor and involve checks such as testing if an analogue value is within an upper and lower bound, checking that provided no topological change has occurred then the rate of change should be less than a specified amount. Measurements which fail

those tests often generate an alarm message to be displayed at an operator's console and are subsequently marked as invalid. More sophisticated checks of this nature include summing all the power flow measurements at a node and checking the result is approximately equal to zero and checking the magnitude of power flow measurements at each end of a line are similar. The switch status measurements may also be compared with power flow measurements to verify the status of the plant.

Two groups of authors have proposed a method of validating the power flow measurements using an algorithm essentially based on Kirchoff's first law.

Irving and Sterling[66] proposed a data validation algorithm for active power flows and switch status measurements in a network using linear programming. The data validation was achieved by formulating the problem in terms of the estimation of the active power flows as follows. The estimated power flow in an element (generator, load or line) must be equal to the measured value (if one exists) plus an error term to allow for any bad data. The sum of the estimated power flows at a node must be equal to zero plus an error term which in this case is forced to be zero by large weighting factors. A third type of linear equation could be appended to those above if any switch status measurements indicated an open circuit. In this case the estimated power flow must be equal to zero plus an error term to allow for an incorrect switch status measurement. The linear programming method calculates values for all the power flow estimates by minimising the sum of the modulii of the error terms. This algorithm forms the basis of the new state estimation technique proposed in this thesis and a more detailed discussion on the method is presented in chapter 4.

An alternative algorithm for validating the active power flow measurements has been proposed by Sawicki, Wilkosz and Kremens.[110] The authors propose that a set of inequality equations be formed by summing the transmission line flows at every node. Any errors in the measurements and the transmission line losses are accounted for by the inequality equation which requires that the nodal sum is less than a pre-defined amount. Each line is included in two inequality equations and a complex statistical analysis of the possible solutions of the set of inequalities enabled conclusions to be drawn on the validity of the measurements. However this method does not take account of the switch status measurements and the algorithm required that at least one of the two flow measurements for each line was subject only to random noise and did contain a gross error.

A combined bad data detection and state estimation process based on the least squares method has been proposed by Prewett et al.[100] and Knight.[73]

This process commences with logical checks on the data and then proceeds to a D.C. estimation followed by an A.C. estimation if necessary. The method has already been discussed in greater detail in section 2.2.3.2, however the overall method contains features relevant to this section and also to the following section on post state estimation bad data detection.

A final point to note is that simple pre-filtering techniques are generally unable to detect bad data whose deviations are less than 30 times the meter's standard deviations.[55,101] It is the role of the state estimator or a more sophisticated filtering scheme to detect these bad data.

2.3.2 Post state estimation detection

Post state estimation bad data detection and correction techniques have generally been applied to the Newton Raphson based weighted least squares method. The title of this section may be a little mis-leading because a number of authors have published reports in which the process of detecting and removing the bad data has commenced before the state estimator has converged. However these methods require that at least one iteration of the estimation process has been completed and it is thus appropriate to include these techniques in this section as the methods are similar to those in which the state estimator has already converged.

The overall process of avoiding the corruption of the state estimates by bad data can be divided into three distinct stages. First, establish that the measurement set contains bad data, then determine which measurements are suspect and finally re-calculate the state estimates with the suspect measurements either removed or corrected.

There are three alternative methods of implementing the three stages. The first is a 3 step procedure in which each stage is treated separately. After the bad data has been detected and identified, it is then removed from the measurement set and a new set of state estimates calculated from the reduced measurement set. The second method employs a 2 step procedure in which the detection and identification of the bad data are combined in the first step and the measurement replacement and re-estimation performed in the second step. The third alternative combines all three stages in 1 step, in other words bad data is detected and its effect on the state estimates removed after each iteration of the state estimator. This last method is often referred to as bad data suppression. It maybe necessary in the first two methods to repeat the detection, removal and re-estimation stages to ensure that all of the bad data has been removed from the measurement set.

The principles behind each of the three methods are outlined below with references to the literature where relevant.

The method of detecting the presence of bad data used in the 3 step method of detection, identification and removal of the bad data employs a statistical treatment of the objective function, J(X). The value of the objective function has a chi-square, χ^2 distribution (which can be approximated to a normal distribution for large samples) and may be used to evaluate the probability of rejecting the hypothesis H_0: 'no bad data are present' and the probability of accepting the hypothesis H_1: 'H_0 is not true, bad data are present'. The reader is referred to text books on statistics[25,46,91] for further information on the mathematical theory of the chi-square test and to papers by Dopazo, Klitin and Sasson,[38] Aboytes and Cory,[1] Handschin, Schweppe, Kohlas and Fiechter[56] and Quintana, Simoes-Costa and Mier[101] for the application of the chi-square test to power system analysis.

Schweppe[114] was the first to discuss the monitoring of the objective function, J(X) to determine if bad data was present. He suggested that a sudden change in value of J(X) from one solution point to the next suggested that bad data had been introduced either by a transmission line tripping and the model of the network not being updated or a transducer failing to an invalid value. If the value of J(X) was gradually increasing from one solution point to the next, then this was an indication that pseudo-measurements were becoming out of date. Usually however bad data was assumed to be present if any one estimation run the value of J(X) exceeded a pre-determined level. The value of detection threshold depended on the required probabilities of failing to detect bad data when some was present and incorrectly flagging the existence of bad data when there was none present.

Having established that the measurement set contained some bad data, it could then be isolated by examining the magnitude of either the weighted residual vector $r_w(X)$ or the normalised residual vector $r_n(X)$.

The weighted residuals are evaluated by pre-multiplying the residual vector, r(X) by the square root of the inverse of the measurement co-variance matrix, R(X).

$$r_w(X) = \sqrt{(R^{-1}(X))}\ r(X) \tag{2.31}$$

where

$r_w(X)$ = weighted residual vector.

R(X) = measurement co-variance matrix.

r(X) = residual vector.

The normalised residuals are evaluated by pre-multiplying the residual vector by the square root of the inverse of the diagonal of the residual co-variance matrix.

$$r_n(X) = \sqrt{(D^{-1}(X))}\ r(X) \tag{2.32}$$

where

$r_w(X)$ = normalised residual vector.

$D(X)$ = diagonal of the residual co-variance matrix $\Sigma_r(X)$, evaluated as shown below.

$$\Sigma_r(X) = W(X)\ r(X) \tag{2.33}$$

$$W(X) = I - H(X)G^{-1}(X)H^t(X)R^{-1}(X) \tag{2.34}$$

$$G(X) = H^t(X)R^{-1}(X)H(X) \tag{2.35}$$

where

$W(X)$ = residual sensitivity matrix.

$G(X)$ = gain matrix.

$H(X)$ = Jacobian matrix.

I = identity matrix.

The following papers by Sirisena and Brown[117] and Handschin et al.[56] provide a more detailed description of the calculation of the weighted and normalised residuals.

Those measurements with large weighted or normalised residuals could be considered to be bad data. Given a set of measurements with no bad data the distribution of the magnitude of either the weighted or normalised residuals could be assumed to be normal.[56] Thus a threshold value for the magnitude of either the weighted or normalised residuals could be calculated and if the measurement residual exceeded this value it could be considered as bad. It would be possible to delete all the bad measurements simultaneously and then re-compute the state estimates and re-check for bad data. However in practice this procedure is not to be recommended due to the effect of one measurement value upon the residuals of adjacent measurements.[1,49,85,93] Thus in practice only the measurement with the largest weighted or normalised residual is

deleted from the measurement set. This may result in the cycle of detecting, identifying and removing the bad data followed by re-estimation being performed several times before all the bad data is removed but it is less likely to result in the removal of a valid measurement which had a large residual due to the interaction of an adjacent bad measurement.

It is generally accepted that the test of the normalised residual is more sensitive than the test of the weighted residual.[49,56,92,111,117] However as can be seen from the equations 2.32 to 2.35 the evaluation of the normalised residuals is a complex and hence time consuming process. A number of authors[85,117] have therefore preferred the use of the weighted residual test on the basis of faster soluter solution times. It should be noted that in the evaluation of the normalised residuals the only elements of the inverse of the gain matrix, G(X) which are required are those occupying the same positions as the non-zero positions of the gain matrix itself. Thus as discussed by Broussolle[17] sparse inverse techniques can be exploited to reduce the time taken to calculate the normalised residuals. Handschin et al.[56] have observed that the normalised residual test is more effective at detecting (and identifying) a single bad measurement than the J(X) test but in the presence of multiple bad data (interacting or non-interacting) neither test is superior. They thus recommended that both tests be used to detect the presence of bad data and if either fails then the measurement with the largest residual must be deleted and the state estimates re-calculated.

Garcia, Monticelli and Abreu[49] discuss the decoupling of the detection and identification of bad data and highlight the effect on the component values of the objective function, J(X) arising from bad data in the other measurement set. Clements, Krumpholz and Davis[27,78] describe a method of network observability which may be extended to identify those measurements which are critical. Critical measurements are those which if removed from the measurement set would result in the system no longer being observable and thus by definition have a residual of zero magnitude. The determination of the critical measurements generates a residual sensitivity matrix which may be used in analysis the residuals to determine the interaction between a measurement and an adjacent residual.

The 2 step method of processing bad data relies on the use of a constant gain matrix in the estimation process. The procedure of deleting a measurement from the set requires the gain matrix to be reformed and re-factorised every time a measurement is deleted. However if a constant gain matrix is used the bad measurement can be replaced with a corrected value derived from the original bad measurement and its residual, thus avoiding the time consuming

re-factorisation.

The methods of identifying the bad data are similar to those already discussed for the 3 step method although Monticelli and Garcia[93] have proposed an alternative single step method based on the normalised residuals. The method is similar to the conventional normalised residual test but instead of calculating a threshold value which if exceeded by any measurement residual identifies that measurement as being bad, the authors propose the calculation of a threshold value which if exceeded by the largest normalised residual identifies that measurement as being bad. The following formula is used to give an estimate of the size of the gross errors

$$b_i = \frac{\sigma_i \, r_{n_i}}{\sqrt{d_{ii}}} \tag{2.36}$$

where

b_i = estimate of the size of the gross error on measurement i.

σ_i = standard deviation of measurement i.

d_{ii} = diagonal element of the residual co-variance matrix $\Sigma_r(X)$.

r_n = normalised residual of measurement i.

If the value of the error, b_i is greater than 4 standard deviations for measurement i then the measurement can be considered to be a bad measurement. In an earlier paper Garcia, Monticelli and Abreu[49] proposed the following equation to calculate a replacement value for the bad measurement

$$z_i^{new} = z_i^{bad} - \frac{(\sigma_i)^2 (z_i^{bad} - z_i^{est})}{d_{ii}} \tag{2.37}$$

where

z_i^{est} = measurement value as calculated from the present estimates.

The authors reported that experimental results showed that the value of the replacement measurement was very close to the true measurement value.

Sirisena and Brown[117] suggest that the process of evaluating the normalised residuals is too time consuming and hence the diagonal of the residual co-vriance matrix is not available to calculate a replacement measurement. The authors propose replacing the measurement preferably with the value calculated from the current state estimates or alternatively by the measurement from the previous scan or from the value calculated using the

state estimates obtained from the previous scan.

Lo, Ong, McColl, Moffatt and Sulley[86] argue that the accuracy of the method of estimating the replacement measurement given in equation 2.37 depends significantly on the accuracy of the standard deviation, σ_i and that in practical terms no advantage is gained in this time consuming calculation. The authors propose that a reliable estimate can be obtained from the original measurement value, the residual, the elements of the Jacobian matrix and the final change in value of the state estimate as defined by the equation

$$z_i^{est} = z_i^{bad} - r_i + \sum_{j=1}^{n} \left[\left| \frac{\delta h_i}{\delta x_j} \right| \Delta x_j \right] \quad (2.38)$$

where

z_i^{est} = new estimated measurement value.

z_i^{bad} = original measurement value.

r_i = measurement residual.

$\delta h_i / \delta x_j$ = elements of the Jacobian matrix evaluated at the solution point.

Δx_j = final change in the state variable j.

Nian-de, Shi-ying and Er-keng[96] have proposed an alternative algorithm for implementing the 2 step method of bad data identification and correction. The method assumes that the measurement set will have p bad data points, however an estimate of the measurement error for s measurements will be calculated. The value of s is defined to lie in the range $p \leq s \ll m-n$, where m = number of measurements and n = number of state variables. The s measurements to be processed are selected using the principle of search for doubtful data. The estimates of the measurement errors are calculated using only a subset of the conventional residual co-variance matrix and the least squares solution method. The low dimensionality of the problem ensures a fast solution time. The estimates of the s measurement errors are then used to correct the original state estimates, the estimates of the remaining m-s measurement errors are set to zero.

The 1 step method of bad data identification and correction often referred to as bad data suppression is based on the Newton Raphson method of state estimation. However the standard least squares solution method is not used and the objective function is no longer quadratic. The term 'non-quadratic criteria' is also often used to refer to these methods.

The least squares method with its quadratic objective function is ideal at

smoothing the results obtained from a measurement set with random normally distributed noise levels. However since the 'effective' weight assigned to a measurement is proportional to the square of the magnitude of the residual, a measurement which is grossly in error will have a large residual and hence a large 'effective' weight. This unfortunately biases the solution point towards the measurement with gross error. Thus an objective function is required which retains the quadratic function for those measurements with small residuals but limits the magnitude of the 'effective' weight assigned to measurements with large residuals. The four non-quadratic criteria which are usually considered, namely quadratic-straight, quadratic-square root, quadratic-multisegment and quadratic-flat (or quadratic-constant) are illustrated by the graph shown in figure 2.1, the conventional quadratic form is also shown for comparison purposes. The point at which the curve deviates from the quadratic function is known as the breaking point, λ. It could be argued that the linear programming state estimator of Irving et al.[65], the least absolute value state estimator (a form of linear programming) of Kotiuga et al.[77] and the state estimation technique based on linear programming proposed in the thesis should be considered as bad data suppression methods. However these methods do not retain the initial quadratic section of the objective function which is usually accepted as being present in a non-quadratic state estimator.

During the early seventies a number of authors, including Merrill and Schweppe[90] and Handschin et al.[56] considered the use of non-quadratic objective functions. However the additional computational burden of implementing the scheme prevented the methods from being of practical use. The application of a non-quadratic criteria modifies the original definition of the objective function as defined by equations 2.15 and 2.16 to

$$\text{Min. w.r.t. X} \quad J = \sum_{i=1}^{m} \left[\Phi_i \left[\frac{z_i - h_i(x_i)}{\sigma_i} \right]^2 \right] \tag{2.39}$$

or in matrix form

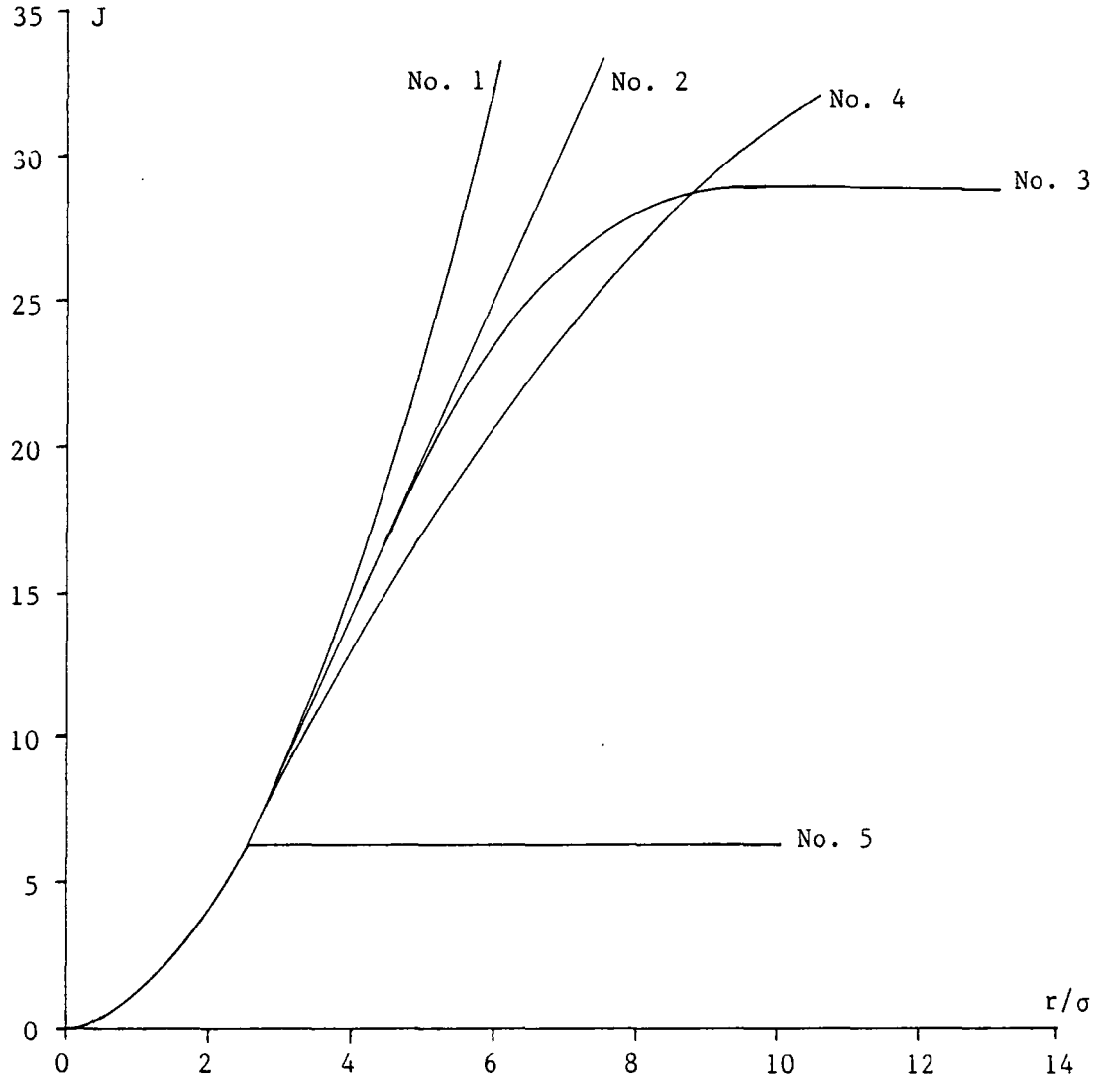
$$\text{Min. w.r.t. X} \quad J = P^t R^{-1} P \tag{2.40}$$

where

Φ_i = penalty associated with the i^{th} residual.

P = vector derived from the residual vector.

The above objection function maybe used to obtain the following definition for



Estimation criteria

No. 1 = Conventional quadratic

No. 2 = Quadratic-straight

No. 3 = Quadratic-multisegment

No. 4 = Quadratic-square root

No. 5 = Quadratic-constant

Fig. 2.1: Various estimation criteria

the value of the update to the state vector, the reader is referred to the literature[56] for details on the method.

$$H^t(X')C^tR^{-1}CH(X')\,\Delta X = H^t(X')R^{-1}CP \qquad (2.41)$$

where

$H(X')$ = Jacobian of h(X) at X'.

C = diagonal matrix of measurement penalties.

The values of the P vector and the diagonal of the C matrix are re-defined after every iteration and depend on the non-quadratic criteria being implemented. The values of the C matrix on the left hand side of equation 2.41 are usually set to one to allow a constant gain matrix to be used. It is reported[85] that this approximation does not affect the final solution point. In the case of the quadratic-flat criteria, the values of the C matrix and P vector are set to one and to the value of the residual respectively, if the measurement residual divided by the measurement standard deviation is below the breaking point and to zero and to the product of the measurement standard deviation and the breaking point otherwise. The values of the C matrix and the P vector for the other non-quadratic criteria are calculated from expressions involving the standard deviation, residual value and the breaking point. The paper by Handschin et al.[56] contains a comprehensive definition of the values for the C matrix and P vector although other authors have suggested minor variations.[85,90,145]

Merrill et al.[90] concluded from their work on non-quadratic state estimators that the quadratic-square root criteria was the most favourable method to use, the authors claim that when tested on a 5 bus system the method was reliable and the convergence was as fast as the conventional weighted least squares method. However other authors[85,145] consider the quadratic-flat criteria to be the most effective method at eliminating bad data although it may be prone to convergence difficulties, it is also the easiest to implement. Conversely the quadratic-straight criteria is the least effective method but is least susceptable to convergence problems. The performance of the other criteria lies somewhere between these two extremes.

The selection of a suitable value for the breaking point has received much attention. The value of the breaking point has a significant effect on the performance of the technique in terms of the mis-identification of bad data. Handschin et al.[56] discuss the possibility of using different breaking points for each measurement, in other words comparing the normalised residuals to the

breaking point. Lo et al.[85] have proposed a method whereby the breaking point is reduced from an upper to a lower limit by a fixed amount after each iteration. The values of the upper and lower limits and the step length were determined by experimental trials. It has been suggested by Zhuang and Balasubransanian[145] that the method of Lo et al. can be improved upon if the breaking point is modified according to the magnitude of the largest update to the state vector and the largest normalised residual below the current value of the breaking point.

To conclude, the bad data suppression techniques generally show good performance. However, they do have the following disadvantages. A fairly high level of measurement redundancy is required for them to be effective and if the bad data is interacting the performance is degraded. The same approximations may be applied to the non-quadratic state estimators as applied to the conventional quadratic state estimators to improve the solution times. However, as would be expected, the solution times will be greater due to the additional computational requirements.

Chapter 3
Simulation of the test networks

An overview of the power system used in this country has been presented in the introductory chapter, chapter 1. This chapter presents a more detailed description of some of the plant found in a power system, together with the mathematical models used to represent the plant and the simulation of the behaviour of power system networks.

3.1 Substation layout

The majority of computer control programs for power systems are not concerned with the details of the layout of a substation and require some pre-processing of the data to form a list of electrical nodes.[30,109,134] A node is defined as a point with a unique voltage magnitude and phase angle, thus all the busbars within a substation which are inter-connected by bus couplers form a node since the bus couplers effectively have no impedance. The generators, loads, transformers and transmission lines are assigned to the node in which their busbar has been placed. There are two reasons for this pre-processing. Firstly a number of the control programs are not able to operate at the busbar level because conventional models of the network break down if elements with zero impedance are included. This therefore precludes the bus couplers from the model of the network. Secondly the scale of the problem is reduced by compressing a number of busbars into one point which in turn will result in faster solution times.

However, the bad data detection and state estimation algorithm presented in thesis is able to operate at the busbar level and in order to enable the full potential of the method to be evaluated the test networks have been modelled down to the busbar level. A conventional topology program[134] has been written to process the network data into a nodal list for those programs in the suite which operate at the nodal level.

The following sub-sections present a more detailed description of the types of switchgear, the measurement transducers and the layout of busbars found in typical substations.

3.1.1 Switchgear

The CEGB uses a variety of different types of switchgear all of which have an intricate system of both electrical and mechanical safety interlocks. The

various types of switchgear can be classified into three groups according to their function. The function of each of the three groups is briefly explained below.

3.1.1.1 Circuit breakers

The only type of switchgear which may be used to connect or disconnect plant from the network is a circuit breaker. A circuit breaker is usually capable of operating under fault conditions and is hence designed to extinguish the arc formed when a large current is interrupted. There are several methods of extinguishing the arc ranging from blowing out the arc with a blast of air as in air-blast circuit breakers, to chemically quenching the arc with an inert gas called sulphur hexafluoride (SF_6).

Circuit breakers are operated by remote control and may be tripped either by an operator or by fault detection equipment. A few circuit breakers found in mesh substations and known as mesh corner disconnectors are intended for use under normal operating conditions only as they are not capable of interrupting fault currents. The layout and operation of a mesh substation is described in section 3.1.3.2.

3.1.1.2 Isolators

Each circuit breaker has an isolator on either side of it which are again remote controlled but are used solely for isolating plant and are not normally capable of interrupting a load current. The major components of one phase of an isolator, often used by the CEGB, are illustrated in figure 3.1, a motor rotates the central insulator so the connecting bar is at 90 degrees to the catches which in turn connect to the rest of the plant.

3.1.1.3 Earthing switches

All plant which is taken out of service for maintenance must be earthed and most switchgear has a manually operated bar which can be swung into contact with the plant. A complex system of mechanical interlocks connected to the isolators together with keys from the control boxes of the circuit breakers ensure that the plant is 'dead' before the earth switch may be operated. The status of the earth switch is not usually telemetered to the control centre and is not usually included in the model of the power system.

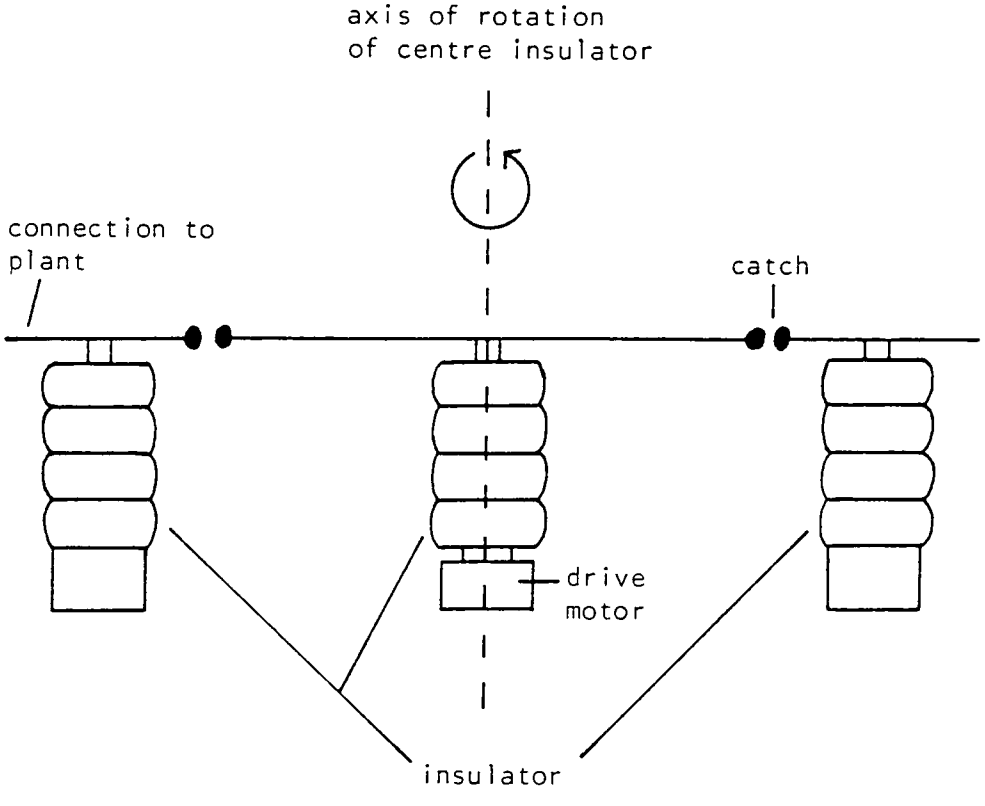


Fig. 3.1: Illustration of an Isolator

3.1.1.4 The numbering and nomenclature of the switchgear

Table 3.1 details the complex numbering system used for the three Supergrid voltage levels, the definition of each of the digits in the number is outlined below.

The three digit number of the 400kV switchgear is prefixed by the letter X. The first digit is the unique sequence number of the switch and the method of determining this number is explained at the end of this section. The second digit defines the type of plant to which the switch is connected, eg. X*0* for a transmission line, X*1* for the high voltage side of a transformer etc.. The third digit defines the function of the switch, eg. X**0 for a circuit breaker other than those connected to transmission lines, X**1 for an earthing switch etc..

The 275kV switchgear is defined by a letter and two digits. The letter defines the type of plant to which the switch is connected, eg. L** for a transmission line, H** for the high voltage side of a transformer etc.. The first digit is the unique sequence number and the second digit defines the function of the switch as in the case of the 400kV switchgear.

The numbering of the 132kV switchgear is similar to that of the 400kV switchgear except that the prefix X is omitted.

The unique sequence number of the switchgear depends on the design of the substation at which the switchgear is located. If the substation has a point which is not designed to be extended in the future then this point is used as the starting position of the sequence of numbers. However if no such point exists then an arbitrary point is chosen, often a bus-coupler circuit breaker. The switchgear in one direction from this point are given odd numbers only and the switchgear in the opposite direction are given even numbers only.

3.1.2 Measurement transducers

The CEGB network has two basic measurement transducers, these are a potential transformer (PT) and a current transformer (CT). Each phase of all the transmission lines has both a potential and current transformer at both ends. The output of these transducers is used by fault detection equipment and the substation operator. The substation operator uses the information for plant synchronization of generators and in the case of a network split the 2 sub-systems since re-connection of plant when the voltage magnitude or the voltage phase angle across the circuit breaker is too large would cause large

CLASS	TITLE	SYMBOLS		
		400 kV	275 kV	132 k
Lines	Switch Disconnector	X*00	L*0	*00
	Line Earthing Switch	X*01	L*1	*01
	By-Pass Disconnector	X*02	L*2	*02
	Line Disconnector	X*03	L*3	*03
	Main Busbar Selector Disconnector	X*04	L*4	*04
	Circuit Breaker	X*05	L*5	*05
	Reserve Busbar Selector Disconnector	X*06	L*6	*06
	Circuit Breaker Disconnector (Busbar Side)	X*07	L*7	*07
Transformer High Voltage Side	Transformer Circuit Breaker	X*10	H*0	*10
	Transformer Earthing Switch	X*11	H*1	*11
	Transformer By-Pass Disconnector	X*12	H*2	*12
	Transformer Disconnector	X*13	H*3	*13
	Main Busbar Selector Disconnector	X*14	H*4	*14
	Switch Disconnector	X*15	H*5	*15
	Reserve Busbar Selector Disconnector	X*16	H*6	*16
Main Bus Section	Main Bus Section Circuit Breaker	X*20	S*0	*20
	Main Bus Section Earthing Switch	X*21	S*1	*21
	Main Bus Section Disconnector (No. 1 side)	X*24	S*4	*24
	Switch Disconnector	X*25	S*5	*25
	Mesh Opening Corner Disconnector	X*26	S*6	*26
	Main Bus Section Disconnector (No. 2 side)	X*28	S*8	*28
Reserve Bus Section	Reserve Bus Section Circuit Breaker	X*60	P*0	*60
	Reserve Bus Section Earthing Switch	X*61	P*1	*61
	Reserve Bus Section Disconnector (No. 1 side)	X*66	P*6	*66
	Reserve Bus Section Disconnector (No. 2 side)	X*69	P*9	*69
Bus Coupler	Bus Coupler Circuit Breaker	X*30	W*0	*30
	Bus Coupler Earthing Switch	X*31	W*1	*31
	Bus Coupler Main Busbar Disconnector	X*34	W*4	*34
	Bus Coupler Reserve Busbar Disconnector	X*36	W*6	*36
Static Series Compensator or Series Reactor	Reactor Circuit Breaker	X*40	R*0	*40
	Reactor Earthing Switch	X*41	R*1	*41
	Main Busbar Selector Disconnector (1st Choice)	X*44	R*4	*44
	2nd Reactor Circuit Breaker where 2 per Reactor	X*45	R*5	*45
	Reserve Busbar Selector Disconnector (1st Choice)	X*46	R*6	*46
	Circuit Breaker Disconnector (Busbar side)	X*47	R*7	*47
	Main Busbar Selector Disconnector (2nd Choice)	X*48	R*8	*48
	Reactor Tie Busbar Disconnector or Reserve Busbar Selector Disconnector (2nd Choice)	X*49	R*9	*49

* Denotes sequence of switch groups

Table 3.1: Switchgear Numbering and Nomenclature

CLASS	TITLE	SYMBOLS		
		400 kV	275 kV	132 k
Static Shunt Compensator	Reactor/Capacitor Circuit Breaker	X*50	K*0	*50
	Reactor/Capacitor Earthing Switch	X*51	K*1	*51
	Reactor/Capacitor Disconnector	X*53	K*3	*53
	Reactor/Capacitor Switch Disconnector	X*55	K*5	*55
Transformer Low Voltage Side	Transformer Circuit Breaker		T*0	*80
	Transformer Earthing Switch		T*1	*81
	Transformer Disconnector		T*3	*83
	Main Busbar Selector Disconnector		T*4	*84
	Switch Disconnector		T*5	*85
	Reserve Busbar Selector Disconnector		T*6	*86
Generators	Generator Circuit Breaker (where 2 per generator, main busbar)	X*90	M*0	*90
	Generator Transformer Earthing Switch	X*91	M*1	*91
	By-Pass Disconnector	X*92	M*2	*92
	Generator Transformer Disconnector	X*93	M*3	*93
	Main Busbar Selector Disconnector	X*94	M*4	*94
	Generator Circuit Breaker where 2 per generator (reserve Busbar) or Switch Disconnector	X*95	M*5	*95
	Reserve Busbar Selector Disconnector	X*96	M*6	*96
	Circuit Breaker Disconnector (Busbar Side)	X*97	M*7	*97

Table 3.1: Contd.

transients in the network which are likely to trip fault detection equipment. The voltage magnitude may be displayed on a dial or more recently on digital equipment, the voltage phase angle is displayed on a synchroscope which is a dial indicating the phase angle lag or lead of the plant to be re-connected with respect to the rest of the network, the pointer will rotate if the operating frequencies of the plant on opposite sides of the circuit breaker are different. The potential transformers on the transmission lines are not sufficiently accurate for display at the control centres, thus if the voltage magnitude of a substation is required at the control centre then a more accurate potential transformer is placed on one of the phases of one of the transmission lines. The power flow in a transmission line is given by the product of the output of the potential and current transformers, this value is telemetered to the control centre if required.

3.1.3 Busbar layout in CEGB substations

The substations commonly found in this country are based on one of the designs described below. It should be noted that when a substation has two or more different voltage levels then the substation is sub-divided according to the nominal voltage levels, transformers may interconnect the different levels.

3.1.3.1 Double busbar substation

A double busbar substation in its most basic form has two busbars, the first named the main bus and the second the transfer bus. The transmission lines, transformers etc. may be connected via a circuit breaker and a pair of isolators to either of the busbars, thus allowing one of the busbars to be taken out of service for maintenance. The main and transfer busbars are sometimes linked together by a circuit breaker, usually referred to as a bus coupler. At larger substations the busbars may be further sub-divided into two sections linked together by a bus coupler. The layout and nomenclature of a typical 400kV substation is illustrated in figure 3.2.

3.1.3.2 Mesh substation

A mesh substation usually has four busbars connected together by bus couplers in a ring. However in some cases the circuit breaker in the bus coupler may be replaced by an isolator or omitted altogether. Each corner of

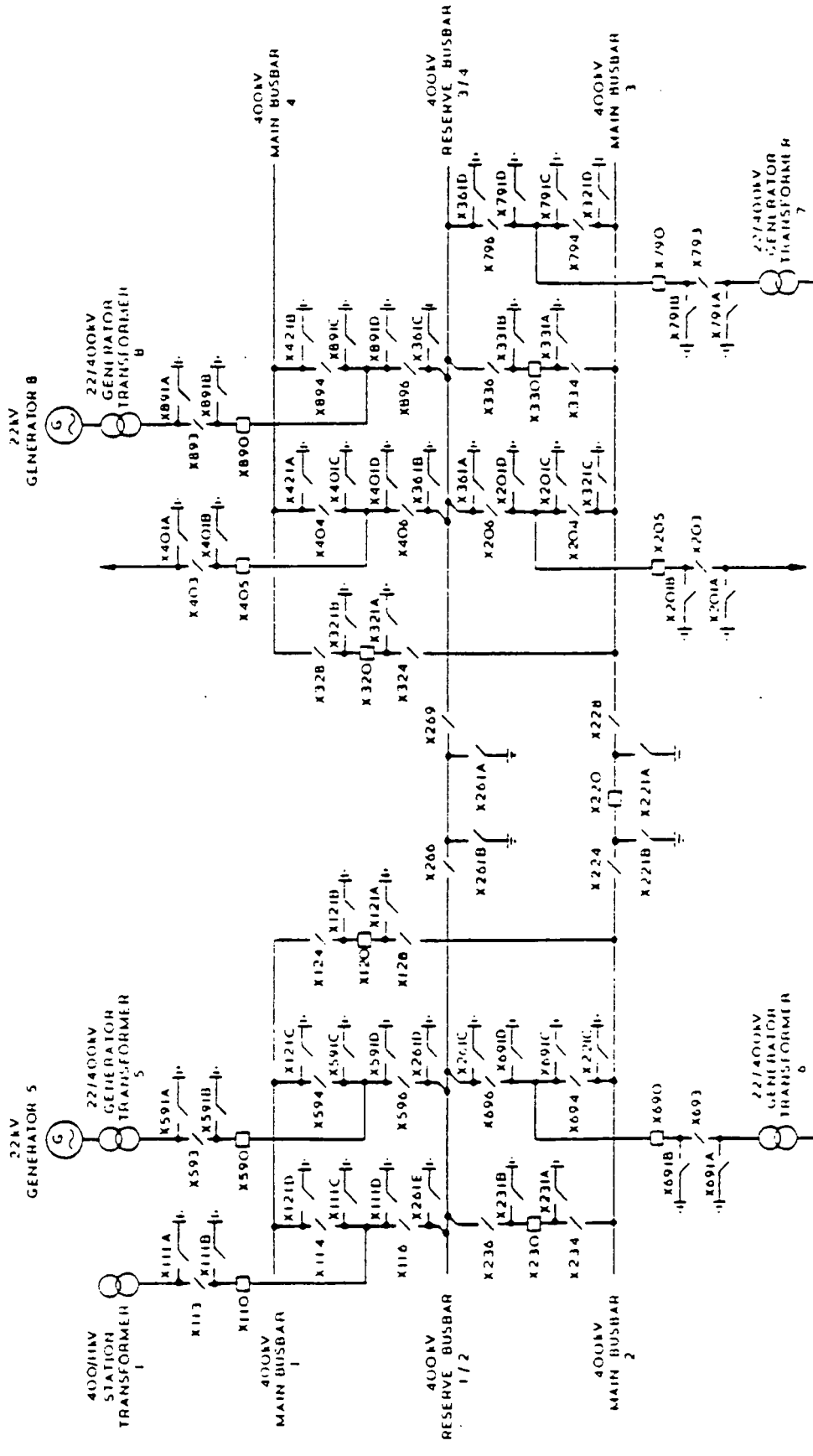


Fig. 3.2: A Back-to-Back 400 kV Double Busbar Substation

the mesh usually has connected to it a transmission line and a load transformer (ie a transformer supplying a low voltage distribution network). Generally the plant is connected to the mesh through an isolator only, hence in the event of a fault both the transmission line and the transformer will be disconnectred simultaneously by the tripping of the circuit breakers in both the adjacent bus couplers. In the larger mesh substations it may be desired to disconnect either the transmission line or the transformer from the mesh without interrupting the supply to the other. This is achieved by placing mesh corner disconnectors in between the two pieces of plant. A mesh corner disconnector is a simple and hence cheaper circuit breaker which is not capable of interrupting a large current and hence may be operated only when the rest of the ring is intact. The following sequence of events is used to perform the operation. Open the mesh corner disconnector associated with the plant element to be disconnected, open the circuit breaker in the adjacent bus coupler, open the isolators of the plant element, close the circuit breaker and finally close the mesh corner disconnector.

Figures 3.3 and 3.4 illustrate two typical mesh substations. The first is a 400kV mesh with no mesh corner disconnectors, the second at 275kV includes mesh corner disconnectors.

3.1.4 Busbar layout in American substations

Unlike the CEGB the electrical power industry in the United States of America consists of a number of independent companies each with their own generating power stations and distribution networks. This results in a series of more nuclear networks with a few interconnections (tie lines) with neighbouring companies. The companies trade power with each other according to a strict schedule which has severe penalties for failing to adhere to it. It is thus in the companies interest to know the exact state of the network and hence a far greater effort than in this country has been put into the computerised operation of the system.

Little published information is available as to the precise details of the layout of the American substations, however two additional types of substation are widely used.

3.1.4.1 Breaker and a Half Substation

A breaker and half substation has two busbars as in the British double busbar substation. However, instead of a transmission line having connections

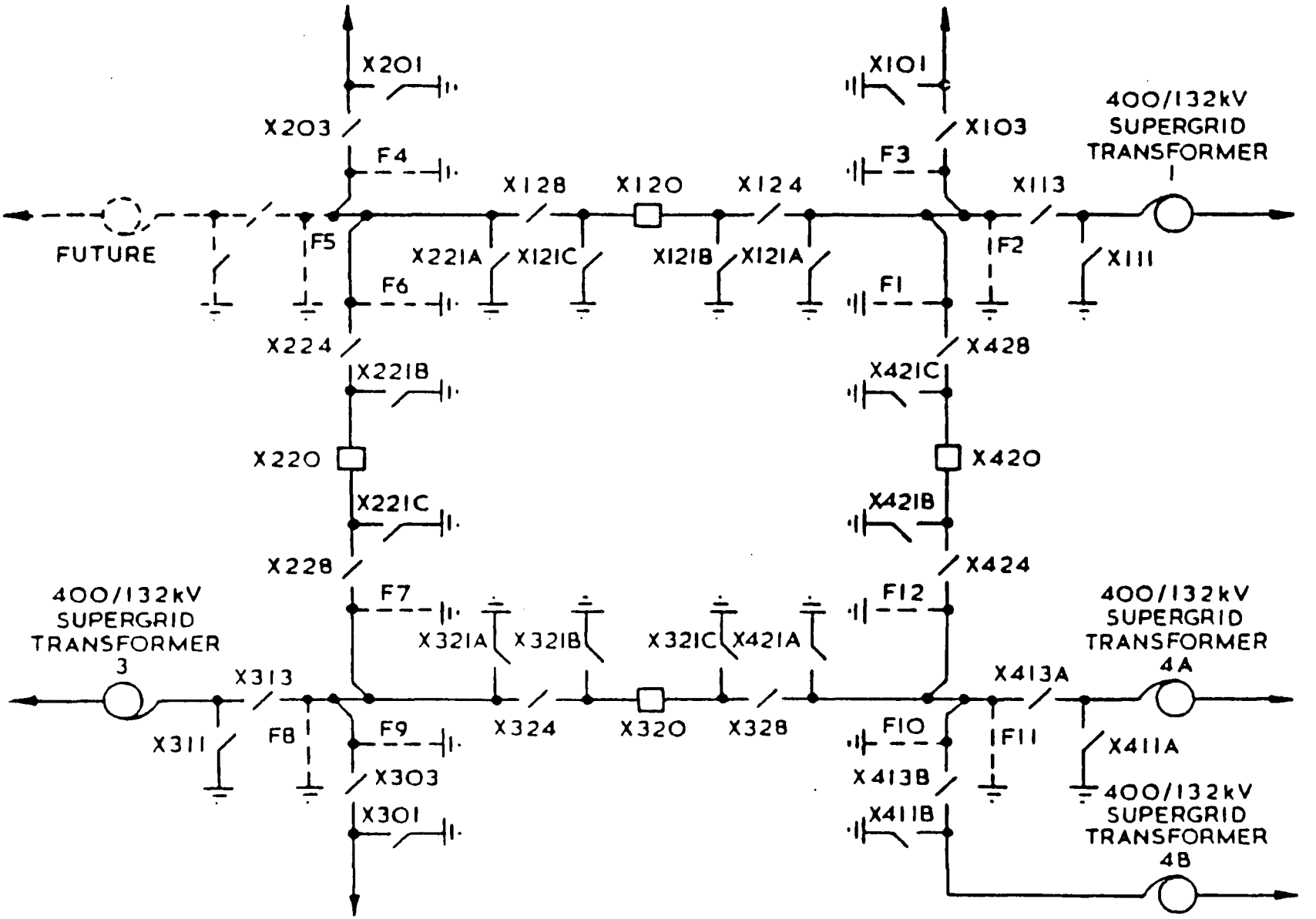


Fig. 3.3: A 400 kV Four Circuit Breaker Mesh Substation

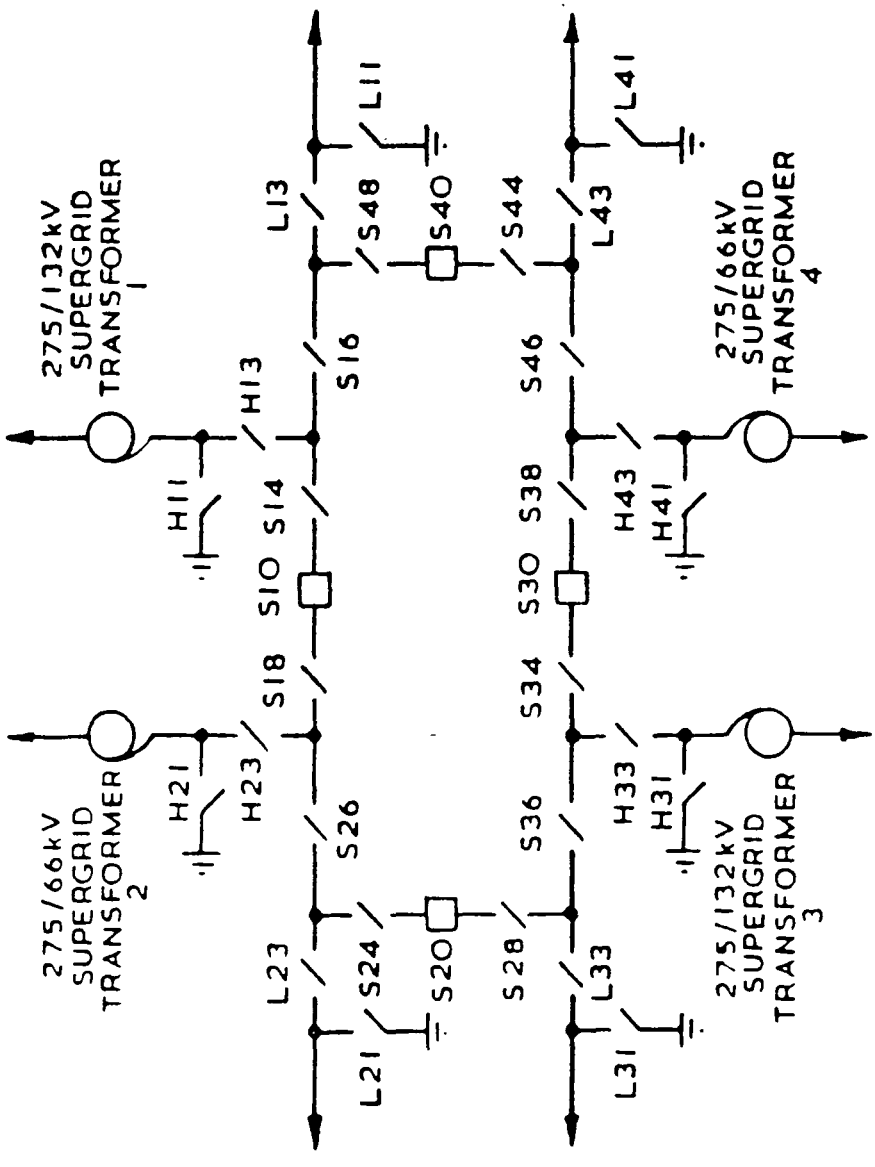


Fig. 3.4: A 275 kV Four Circuit Breaker Mesh Substation With Mesh Opening Corner Disconnectors.

to either busbar via a circuit breaker and two isolators a pair of lines have a direct connection to one busbar via a circuit breaker and an indirect connection through a circuit breaker between the pair of lines and the circuit breaker of the other line. Thus two transmission lines share three circuit breakers, hence the term breaker and a half. The positioning of the circuit breakers is illustrated in figure 3.5.

3.1.4.2 Ringbus substation

A ringbus substation is similar in concept to the British mesh substation, although the number of busbars is greater than four. As the name suggests, the busbars are linked together by bus couplers to form a ring, the number of busbars depends on the amount of plant at the substation.

3.2 Representation of the plant in the test networks

The test networks are a single line representation of the power system. It is assumed that the three phases are balances in all cases except for fault analysis when additional information about the interaction of the three phases is required. Computationally it is far easier to represent the plant in the power system by lists of device numbers, eg. transmission line numbers, substation numbers etc. However in practice the power system operators refer to the plant by names and numbers. Thus a look-up table would be required to translate the real name and number into a unique list of numbers to represent the plant in the model.

3.2.1 Busbars

The busbars have been represented by a device called a bus-section. The bus-sections may be linked together by a device representing the bus couplers, (described in section 3.2.3). The plant elements such as generators etc. may then be connected to these bus-sections. The bus-sections must be assigned to a substation but there is no restriction on the number of bus-sections in any one substation. However any two bus-sections connected together by a bus coupler must be in the same substation and any two bus-section connected together by a transmission line or a transformer must be in different substations. In order to represent the feature found in a double busbar substation where a plant element maybe connected to either or both the busbars by a pair of isolators, an imaginary bus-section has had to be introduced.

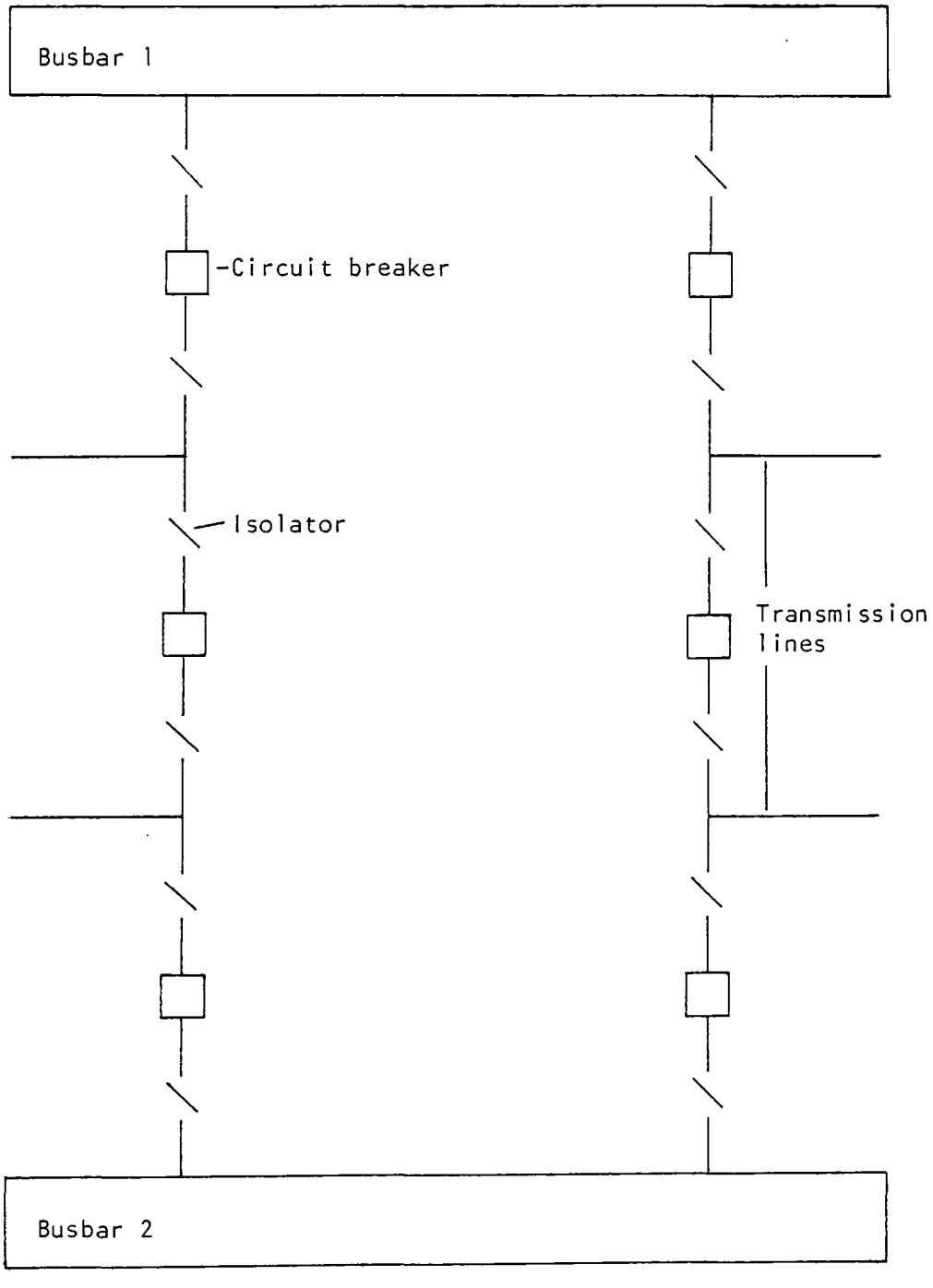


Fig. 3.5: Illustration of a breaker and a half substation

The plant element is connected to the imaginary bus-section which is then linked to the two bus-sections representing the busbars. No distinction is presently made between the bus-sections representing the busbars, usually referred to as real bus-sections and the imaginary bus-sections.

3.2.2 Switchgear

All the types of switchgear found in power systems have been represented by a device called a switch, the earthing switches have not been included in the test networks. The status of the switch is stored in an integer word and may be either zero or one which represents circuit open or closed respectively. Modern telemetry systems use a two bit code for the status of switchgear which gives a possibility of four states, two error states and one each for the switch open and closed. The error states may be due to the switchgear malfunctioning or an error in the data transmission from the switch status transducers to the control centre. This state of uncertainty is an enhancement which will be added in the future, a code of either minus one or two will be used to represent this state.

It has been assumed in the test networks that all transmission lines, transformers, generators and loads have a switch connecting them to the bus-section. Only one switch is catered for so if for instance both the status of the circuit breaker and the isolator were telemetered to the control centre then some pre-processing would be necessary to combine the two status measurements into one. All the switches have been assumed to be capable of operating under all conditions.

3.2.3 Bus couplers

Bus couplers link the busbars together and usually have a circuit breaker in them which is capable of operating under fault conditions although the bus couplers linking transfer busbars may only contain an isolator and hence are not capable of operating when the busbars are under load. The bus couplers have been represented by a device called a link and may connect both real and imaginary bus-sections. The links may contain any number of switches or none at all, and the status of all the switches must be closed for the link to be active. The link switches have been assumed to be capable of operating under all conditions.

3.2.4 Transmission lines and transformers

Transmission lines and transformers have both been represented by a single device called a line. This is a temporary measure and the transformers will be represented by their own device in the future when auto tap-changing will be introduced. The tap position of a transformer is currently fixed at its nominal value. The lines may terminate on any type of bus-section subject to the restriction that they are in different substations. Although transformers will be located at one substation they will be connecting busbars at different voltage levels and these busbars will be in different sub-sets of the substation thus the restriction is not unjustified.

3.2.5 Generators and loads

The generators and loads have been represented by devices of the same name and may be connected to any bus-section.

The generators always supply active power to the network but may supply or draw reactive power from the network according to the model described in section 3.3.3. If the generator connects to a transformer then an imaginary substation is needed in between the generator and the line representing the transformer.

Load transformers are not usually represented, a load draws active power from the network and may supply or draw reactive power from the network. The loads have also been used to represent static reactive compensators which may supply or draw reactive power only. In load flow studies these compensators are often represented by a line with no resistance which has both ends terminating at the same electrical node. This method of representation was not suitable for the simulation of the network and hence the current representation was adopted. This representation has the disadvantages that the amount of compensation is fixed and does alter with the changing nodal voltages and secondly the control algorithms which validate the measurements in the network and estimate the states of any unmeasured points cannot use the constraint that all loads must draw both active and reactive power from the network. The static reactive compensators are likely to be represented by their own device in the future.

3.2.6 Measurement transducers

The active power flow, the reactive power flow or both power flows may be measured at any generator, load, line or link. The devices may also have

duplicate measurements or no measurement at all. The liklihood is that if a piece of plant has a power flow measurement then both the active and reactive flows will be available and in the all test networks this is so. It is unlikely, at least in the near future that the links which represent bus couplers would have any power flow measurements available, however a recent paper by Rossier and Germond[108] studied the optimisation of the use of switchgear to minimise the chance of overloads on the busbars and bus couplers. An estimation of the power flows through the bus couplers would be a big advantage in this process hence the power flow measurements in links have been made available to allow for the full investigation of an algorithm attempting the estimation.

The voltage magnitude may be measured at any bus-section, whether it is real or imaginary, and not at the ends of transmission lines as is the case in the CEGB network. Provision has also been made for the measurement of the voltage phase angle at any bus-section although in practice it will be a long time before phase angle measurements become available. None of the test networks make use of this provision.

The control centre of a network is likely to have two frequency measurements available, one made on the supply to the centre and the other telemetered in from a measurement made on a separate part of the network in case the local supply fails. Generating stations will have their own local measurements for use in synchronising generators. The frequency may be measured at any bus-section in the test networks.

3.3 Mathematical model of the power system

3.3.1 Bus couplers

The bus couplers (links) are short conductors and it has been assumed that they have negligible impedance. Any pair of bus-sections connected together by an active link therefore have the same voltage magnitude and phase angle and may be considered as an electrical node.

3.3.2 Transmission lines and transformers

The transmission lines and transformers (lines) have been modelled by the eqivalent circuit as illustrated in figure 3.6. The three admittances have been coupled together to form the shape of the Greek letter π and hence this model of a line is often referred to as a π section model. The three

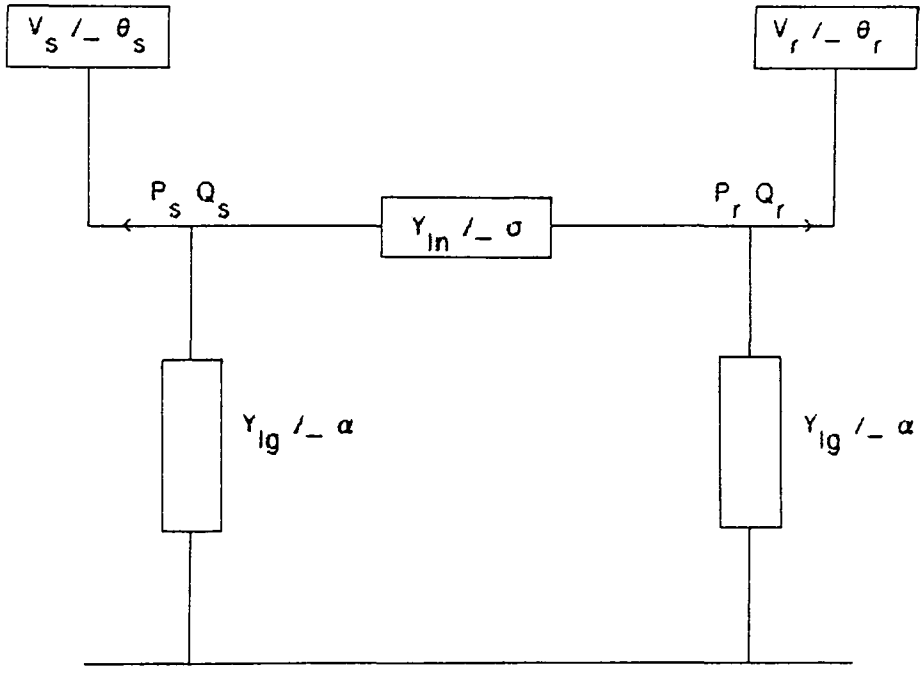


Fig. 3.6: π representation of a transmission line

admittances do not represent the same physical parameters for transmission lines as for the transformers.

In the case of the transmission lines the series admittance is the reciprocal of the line impedance. It should be noted that in the literature[47,122,124] the line impedance is usually expressed in terms of the series resistance and (inductive) reactance. The two shunt admittances represent the effects of the capacitive line changing between the line and the ground. As is often the case the value of the admittances at both ends of the line have been assumed to be equal, and the value of the conductance has been set to zero. The effect of line charging on the currents flowing in certain lines are often small and in a number of cases the effect is neglected all together, ie the values of the shunt admittances are set to zero.

It is usual to express the value of the line changing admittance as one half of the total susceptance of the line, ie the susceptance represents the value of one of the shunt admittances in figure 3.6 and may be thought of as the effect the line charging has on the current flowing at that terminal bus-section. However is some cases the total line charging susceptance for the line may be listed. The equations presented in the thesis assume that the line charging susceptance is one half of the total.

The series admittance of figure 3.6 represents the leakage reactance of a transformer. The leakage reactance of a transformer is usually greater than the series reactance of a transmission line and this can be observed when studying the parameters of the line data which represent transformers. The series resistance of a transformer would represent the energy lost in the windings of the transformer, however this is negligible and hence the series resistance is set to zero for a transformer. The shunt admittances represent the magnetising reactances (inductive) of the transformer. The magnetising reactance of a transformer is usually very large and hence the susceptance is very small, thus the line charging susceptance of the line data representing a transformer is usually set to zero.

The series impedance of the line can be expressed in terms of the above parameters as follows

$$Z = R + j\,X \tag{3.1}$$

where

Z = series line impedance.

R = series line resistance.

X = series line reactance.

and hence the series line admittance as

$$A_{ln} = G_{ln} + jB_{ln} \tag{3.2}$$

$$= \frac{1}{R + j\,X} = \frac{R}{R^2 + X^2} + j\,\frac{-X}{R^2 + X^2}$$

where

A_{ln} = series line admittance.
G_{ln} = series line conductance.
B_{ln} = series line susceptance.

The above equation may be expressed in polar terms as

$$(Y_{ln})^2 = \frac{R^2}{(R^2 + X^2)^2} + \frac{X^2}{(R^2 + X^2)^2} \tag{3.3}$$

$$\sigma = \arctan\,(-X/R)$$

where

Y_{ln} = series line admittance magnitude.
σ = series line admittance angle.

Similarly the shunt admittance between the line and the ground can be expressed as

$$A_{lg} = G_{lg} + jB_{lg} \tag{3.4}$$

where

A_{lg} = shunt admittance between the line and the ground.
G_{lg} = shunt conductance between the line and the ground.
B_{lg} = line charging susceptance between the line and the ground.

or in polar terms as

$$(Y_{lg})^2 = (G_{lg})^2 + (B_{lg})^2 \tag{3.5}$$

$$\alpha = \arctan\,(B_{lg}/G_{lg})$$

where

Y_{lg} = shunt admittance magnitude between the line and the ground.

α = shunt admittance angle between the line and the ground.

In a power system most of the elements at the load points on a power system have an inductive reactance which means that the current lags behind the voltage. The result of this effect is that the load draws reactive power from the network. The term "the net power" or alternatively the "the nodal injection" refers to the total power absorbed or supplied by the loads and generators connected to the node. Inorder to ensure that the imaginary component of the power flows, known as the reactive power flows, have positive values the equation defining the power flows in the system is usually written with a minus sign as shown below

$$S = P - j\,Q \tag{3.6}$$

where

S = complex or apparent power flow, measured in Volt-Amperes (VA).

P = real or active component of the power flow, measured in Watts (W).

Q = imaginary or reactive component of the power flow, measured in Volt-Amperes-reactive (VAr).

This convention unfortunately is not always adhered to and can lead to confusion over the signs of the terms used in equations defining power flows. The thesis follows the above convention. In the literature[65,103] the equations defining the power flows in a line usually use the following convention for the direction of flow. If a line connects bus-section i and j then the line flow (P_{ij},Q_{ij}) is positive if the direction of flow is from bus-section i to bus-section j. However in the test networks the power flows of all the devices except the links have been defined to be positive if the power is flowing into the bus-section to which the device is connected, the link flows are defined to be positive if the power is flowing from the sending to the receiving bus-sections. This convention was adopted so that on an on-line display diagram of a substation the direction of flow could easily be seen without the need to know any details of the inter-connections of the lines.

The line flows can be expressed in terms of the line parameters and the terminal voltages as follows

$$P_s - j\,Q_s = -(E_s)^*(E_s - E_r)A_{ln} - (E_s)^*E_sA_{lg} \tag{3.7}$$

where
the subscripts s and r refer to the receiving and sending ends of the line respectively.
E = complex terminal voltage of the line.

Writing the voltage in terms of its real and imaginary components

$$E = e + j\,f \tag{3.8}$$

then equation 3.7 may be expanded and separated into the real and imaginary components as below

$$P_s = \frac{-(e_s\,\Delta e + f_s\,\Delta f)R + (f_s\,\Delta e - e_s\,\Delta f)X}{R^2 + X^2} \tag{3.9}$$

$$Q_s = \frac{-(e_s\,\Delta f + f_s\,\Delta e)R + (e_s\,\Delta e + f_s\,\Delta f)X}{R^2 + X^2} + ((e_s)^2 + f_s)^2)B_{lg} \tag{3.10}$$

where
$\Delta e = e_s - e_r$.
$\Delta f = f_s - f_r$.

Written in polar terms equations 3.9 and 3.10 become

$$P_s = V_sV_rY_{ln}\cos(\theta_s - \theta_r - \sigma) - (V_s)^2(Y_{ln}\cos(\sigma) + Y_{lg}\cos(\alpha)) \tag{3.11}$$

$$Q_s = V_sV_rY_{ln}\sin(\theta_s - \theta_r - \sigma) + (V_s)^2(Y_{ln}\sin(\sigma) + Y_{lg}\sin(\alpha)) \tag{3.12}$$

where
V = magnitude of the terminal voltage.
θ = phase angle of the terminal voltage.

The power flows for the receiving end of the line are calculated by inter-changing the values of the sending and receiving voltages.

3.3.3 Generators

The majority of the control programs use a simple model to represent a generator in which a generator may supply active power to the network and either supply or draw reactive power from the network, and it is assumed that both the active and reactive power flows are likely to be measured. However, any control program analysing the transient behaviour of the network and also the program used to simulate the behaviour of the networks needs a more sophisticated model which reflects the dynamic behaviour of the generators.

The dynamic behaviour of the generators may be modelled by representing the generator by an equivalent circuit which contains a voltage source in series with a reactance, as illustrated in figure 3.7. The term usually used to describe this equivalent circuit is a 'voltage behind a transient reactance'. The value of the reactance is constant as is the magnitude of the voltage source, however the phase angle of the voltage source with respect to the phase angle of the terminal voltage is allowed to change. The value of the phase angle of the voltage source with respect to the voltage phase angle at the generator terminals determines the active power that the generator supplies to the network, thus as the total load on the network changes the phase angle of the voltage source changes in order to meet the change in the load. However the rate of change of the phase angle of the voltage source is governed by other factors such as the inertia of the generator rotor, the mechanical power input to the generator etc.. Thus the instantaneous value of the phase angle of the voltage source may be such that total power generated by all the generators does not equal the total load on the network. This leads to an oscillatory situation as the parameters of the generators change in response to the imbalance because the controllers will initially overshoot resulting in an over compensation. The dynamic behaviour of the generators can be modelled mathematically by the following equations, for further details on the analysis of rotating machines see chapter 10 of Stagg and El-Abiad.[124] Mathematically the voltage behind the transient reactance can be written as follows

$$E' = E + r_a I_t + j\, x'_d I_t \qquad (3.13)$$

where

E' = complex voltage behind the transient reactance.

E = complex terminal (bus) voltage of the generator.

I_t = complex terminal current of the generator.

r_a = armature resistance of the generator.

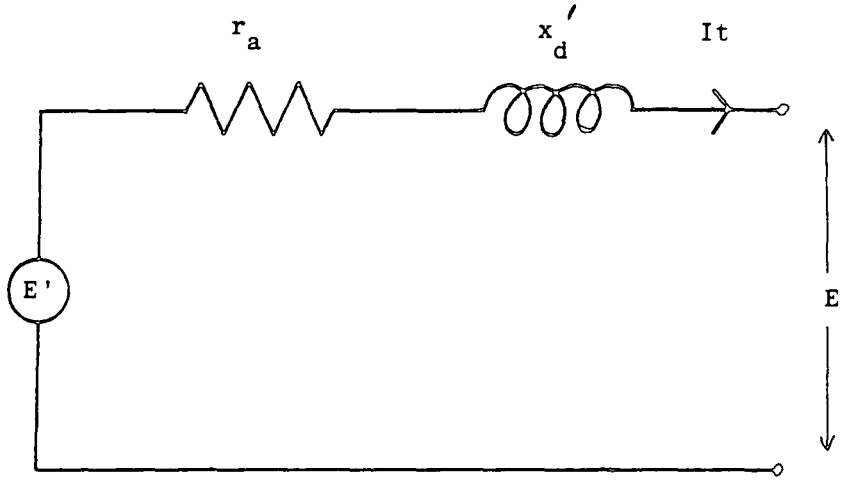


Fig. 3.7: Equivalent Circuit of a Generator

x'_d = transient reactance of the generator.

The relationship between the phase angle of the voltage source, i.e. the electrical angle and the other parameters of the generator which govern the rate of change of the angle can be expressed by two first order differential equations as follows

$$\frac{d^2\delta}{dt^2} = \omega^{\cdot} = \frac{\pi F(P_m - P_e)}{H} \tag{3.14}$$

$$\delta^{\cdot} = \frac{d\theta_e}{dt} - \omega^0 = \omega - 2\pi F \tag{3.15}$$

where

δ = electrical angular position in radians of the generator rotor with respect to a synchronously rotating reference axis.

ω = rate of change of the electrical angle in radians/second.

ω^0 = rated synchronous speed in radians/second.

F = frequency of the system.

H = inertia constant of the generator.

P_m = mechanical power input to the generator.

P_e = electrical power output of the generator.

θ_e = electrical angle of the generator (mechanical angle * number of pairs of poles).

The above two equations are sufficient to represent the behaviour of a generator for a short period of time as in the case of a transient stability analysis when the mechanical power input of the machine can be assumed to be constant. However over a longer time period the mechanical power input will change under the action of the steam valve control governors. The governors respond to deviations in the frequency of the system from a pre-set value and will have a respond lag which is proportional to the steam constant of the boilers.

The rate of change of the mechanical power input can be defined by a third differential equation as follows

$$\frac{d P_m}{dt} = (P_m)^{\cdot} = \frac{P_{set} + G_g(F_{set} - \omega/2\pi) - P_m}{T_c} \tag{3.16}$$

where

P_{set} = present power set point.

F_{set} = present frequency set point.

G_g = generator governor gain.

T_c = steam time constant.

The above differential equations are linked to the algebraic equations used to represent the behaviour of the network by a further four algebraic equations. Network analysis programs such as load flow studies usually relate the net power injection at the electrical nodes to the voltage levels throughout the network, hence the generator algebraic equations equate the active power injection to the voltage behind the transient reactance and the electrical angular position of the generator rotor.
The generator algebraic equations can be written as follows

$$P_e = \text{Real part } (I_t(E')^*) \tag{3.17}$$

$$e' = |E'| \cos(\delta) \tag{3.18}$$

$$f' = |E'| \sin(\delta) \tag{3.19}$$

$$I_t = \frac{E' - E}{r_a - j\, x'_d} \tag{3.20}$$

where

E' = $e' + j\, f'$

= complex voltage behind the generator transient reactance.

E = complex terminal (bus) voltage of the generator.

The program used to simulate the behaviour of a power system updates the value of the magnitude of the voltage behind the transient reactance ($|E'|$) at every time step in order to reflect the automatic voltage regulation of the generators. The fractional voltage error (ΔE) between the generator terminal (bus) voltage magnitude and the voltage set point is calculated as shown in equation 3.21 and if it is greater than a pre-set tolerance then the value of the voltage magnitude behind the transient reactance is updated as shown in equation 3.22.

$$\Delta E = \frac{|E| - E_{set}}{E_{set}} \tag{3.21}$$

$$|E'|^{k+1} = |E'|^{k} (1 - \Delta E G_v) \tag{3.22}$$

where

ΔE = fractional voltage error.
k = time step number.
E' = complex voltage behind the generator transient reactance.
E = complex terminal (bus) voltage of the generator.
E_{set} = magnitude of the voltage set point.
G_v = gain constant of the voltage control loop.

3.3.4 Loads

The majority of the control programs model the loads in a similar way as the generators, that is the loads draw active power from the network and may either supply or draw reactive power. Measurements of the power flows may or may not be available.

The simulator program models the loads as equivalent shunt admittances between the bus-section and the ground in order to reflect the change in the load power flows with the fluctuations in the voltage levels. However to prevent the value of the load power flows from continually drifting the values of the equivalent shunt admittances are calculated at each time step from the load demand stored in the data base and the current voltage levels. The following equations are used to calculate the rectangular components of the equivalent shunt admittance.

$$G_{ld} = P_{ld}/V^2 \tag{3.23}$$

$$B_{ld} = Q_{ld}/V^2 \tag{3.24}$$

where

G_{ld}/B_{ld} = equivalent shunt conductance/susceptance of the load.
P_{ld}/Q_{ld} = modulus of the active/reactive load power demand.
V = voltage magnitude at the load bus-section.

3.4 The test networks

The test network used to develop and test the control programs have been based on the following standard networks: the 5 node network described in Stagg and El-Abiad[124], the IEEE 30 node network[47], the IEEE 57 node network[47] and the IEEE 118 network[122]. The 5 node network has been used mainly for debugging and initial testing of the programs as it enables dumps of the program variables to be made at regular stages without creating vast lists of numbers in which an error may easily be overlooked. The 30 node system is the largest network upon which the simulator will run in real time, that is it takes one second of CPU time to simulate one second of the behaviour of the network. The 57 and 118 node networks have been used primarily for timing the control programs on larger networks.

The majority of published work on the computerised control of power networks has been concerned with a nodal representation of the network, that is all the busbars at a substation which are linked together by active bus couplers are considered as one electrical node. In normal operation the busbars within a substation will either be completely isolated or coupled to all the other busbars thus forming one node, however under extreme circumstances it is possible for some types of substation to be split into more than one node. Under these types of conditions both control and advisory programs must be able to calculate accurate results quickly, without any errors arising from the split. It is for this reason together with the interest in minimising the chances of circuit overloads by optimising the use of switchgear mentioned in section 3.5.6 that the 5 and 30 node test networks have been expanded to include individual busbars and bus couplers (bus-sections and links). Each of the nodes in the standard 5 and 30 networks has been represented by a substation, the layout of the bus-sections within the substations has been totally arbitrary as there is no published data available on the substation design, although a wide selection of typical designs have been included. In order to maintain compatibility with the standard networks and results a small topology program based on an algorithm by Sullivan, Reichart and Saly[134] monitors the status of all switches and forms a list of active nodes and the inter-connections of the lines. Thus any program concerned only with a nodal representation of the system may run on the nodal lists without the need for access to the bus-section data and the results may be compared with those in the literature. These programs will continue to run should a substation split into two nodes because the topology program will form a new nodal list which has increased in length by one. Similarly by disconnecting the lines to a substation and isolating it from the

system the number of active electrical nodes and hence the length of the nodal list may also be decreased. To help avoid confusion over the names of the networks, the names of the test networks presented in the thesis have been changed to 5 substation test network etc., as opposed to the 5 node network etc. which are the names generally applied to standard networks.

The standard IEEE networks include synchronous compensators which are similar to generators in that they are a rotating machine, however they may only supply or draw reactive power from the network. Nowadays synchronous compensators are not widely used, the reactive power flows throughout the network being controlled by the use of switchable shunt capacitors or inductors (static compensators).

The replacement of the synchronous compensators by the equivalent load representation of the static compensators would reduce the number of generators on the test networks significantly making the economic dispatch and rescheduling problems trivial. The synchronous compensators have therefore been replaced by small generators with the result that the power flows and the voltage levels throughout the test networks have been modified slightly from the standard load flow results. The starting conditions for the 5 substation test network are identical to the load flow results in Stagg and El-Abiad[124] and the starting conditions for the 30 substation test network are similar to the published results.[65,123]

The load flow results for the standard IEEE 57 node network are not readily available, only the line parameters and the initial starting voltages appear to have been published[47]. Therefore the generator set points for all the generators (including those which were originally synchronous compensators) and the voltage levels for the 57 substation test network were set to give a starting condition for the simulator which satisfied all the network constraints, ie the voltage magnitude levels and power flow limits. Details on the load flow results and the simulator starting conditions for the 118 substation test network can be found in section 3.4.4.

These changes mean that any control program running in a simulated on-line environment may not compare results with those from the standard networks. This is of no real loss because once the system has been disturbed from the initial steady state then the dynamic behaviour of the generators changes the voltage levels and power flows throughout the network thus making any comparisons with the standard networks invalid. However the control programs may be modified slightly to run in an off-line mode in which case the generator set points and the voltage levels may be set to those of the standard networks.

3.4.1 The 5 substation test network

The 5 node network in Stagg and El-Abiad[124] was chosen for one of the test networks because a small network was needed to allow for the easy degbugging of the control programs during their development. The 5 substation network has had two substations expanded into multiple bus-sections, one with two bus-sections the other with four.

A key to the symbols used in the network diagrams is presented in figure 3.8 and figure 3.9 details the layout of the 5 substation test network together with all the measurement points. The generator and line parameters are listed in appendix 1.

3.4.2 The 30 substation test network

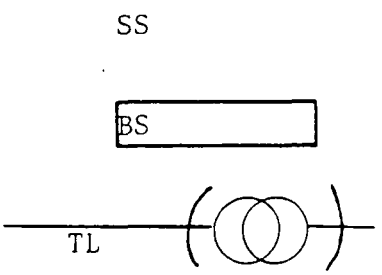
The IEEE 30 node network[47] was chosen as a test network as it is the largest network which the simulator program can simulate in real time. The number of bus-sections in a substation varies from one in the simple substations to a maximum of ten in the most complex. Figure 3.10 is a schematic representation of the geographical layout of the 30 substation test network and figures 3.11 to 3.40 detail the layout of the bus-sections and the measurement points. The generator and line parameters are listed in appendix 1.

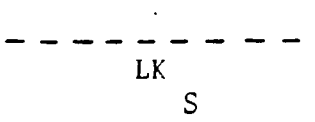
3.4.3 The 57 substation test network

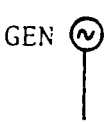
The IEEE 57 node network[47] has not had any of the substations expanded, thus each node is equivalent to a substation and contains only one bus-section. The maximum number of electrical nodes in the 57 substation test network is therefore 57, although by disconnecting the lines to a substation the number of active nodes may be decreased. The simulator program is able to simulate the 57 substation network at approximately half the speed of real time. A schematic illustration of the geographical layout of the substations is presented in figure 3.41, full details of all the network parameters are listed in appendix 1.

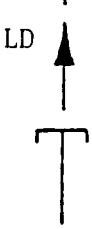
3.4.4 The 118 substation test network

The 118 substation test network was developed from a load flow solution









SS	SUBSTATION
BS	BUS-SECTION
	TRANSMISSION LINE (WITH TRANSFORMER)
	LINK
	SWITCH, BREAKER OR ISOLATOR
	GENERATOR
	LOAD
	STATIC REACTIVE LOAD COMPENSATOR
V	VOLTAGE MAGNITUDE MEASUREMENT
F	FREQUENCY MEASUREMENT
LNP	TRANSMISSION LINE ACTIVE POWER FLOW MEASUREMENT
LNQ	TRANSMISSION LINE REACTIVE POWER FLOW MEASUREMENT
LKP	LINK ACTIVE POWER FLOW MEASUREMENT
LKQ	LINK REACTIVE POWER FLOW MEASUREMENT
GP	GENERATOR ACTIVE POWER MEASUREMENT
GQ	GENERATOR REACTIVE POWER MEASUREMENT
LDP	LOAD ACTIVE POWER MEASUREMENT
LDQ	LOAD REACTIVE POWER MEASUREMENT

Fig. 3.8: Key to symbols and abbreviations used in the network diagrams

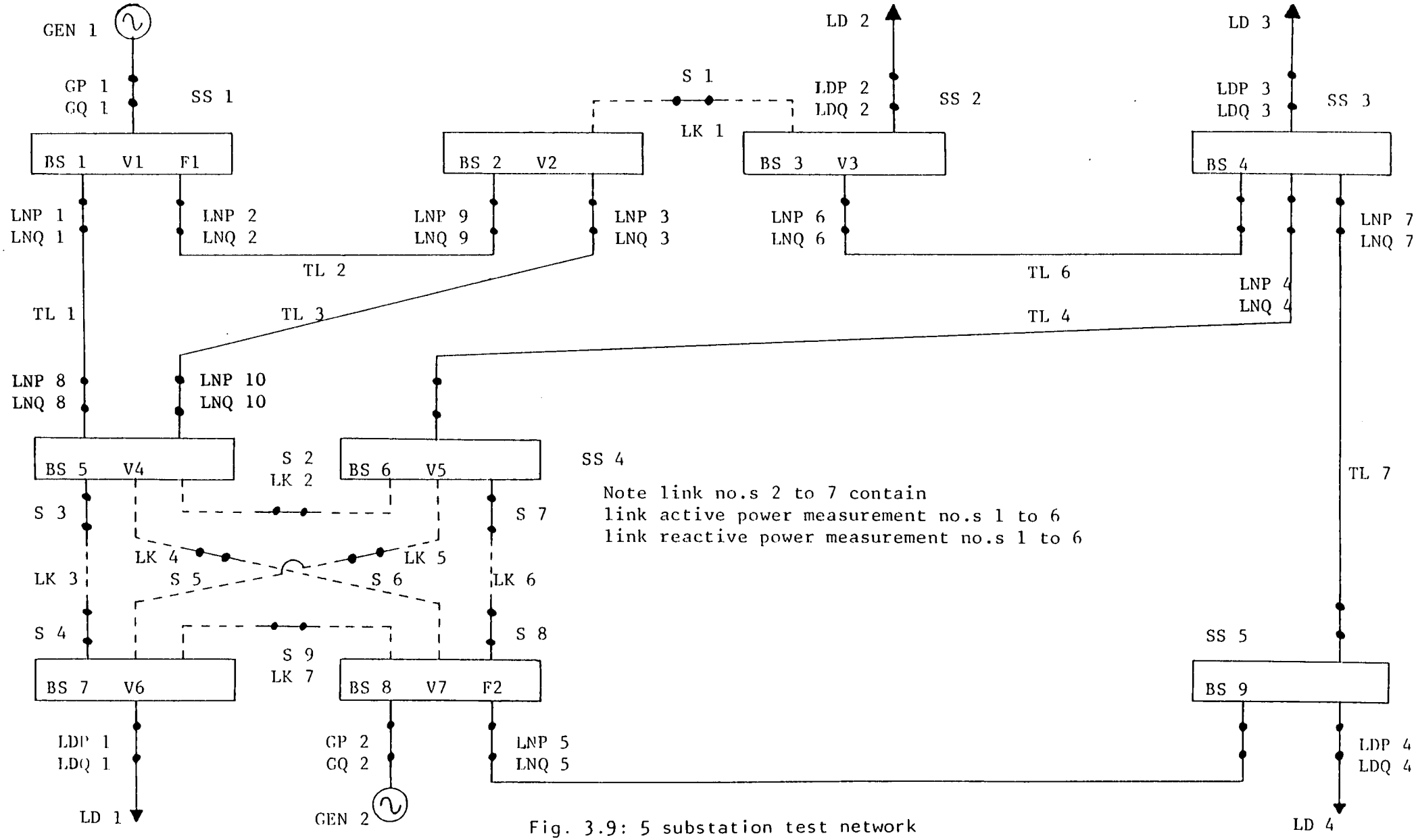


Fig. 3.9: 5 substation test network

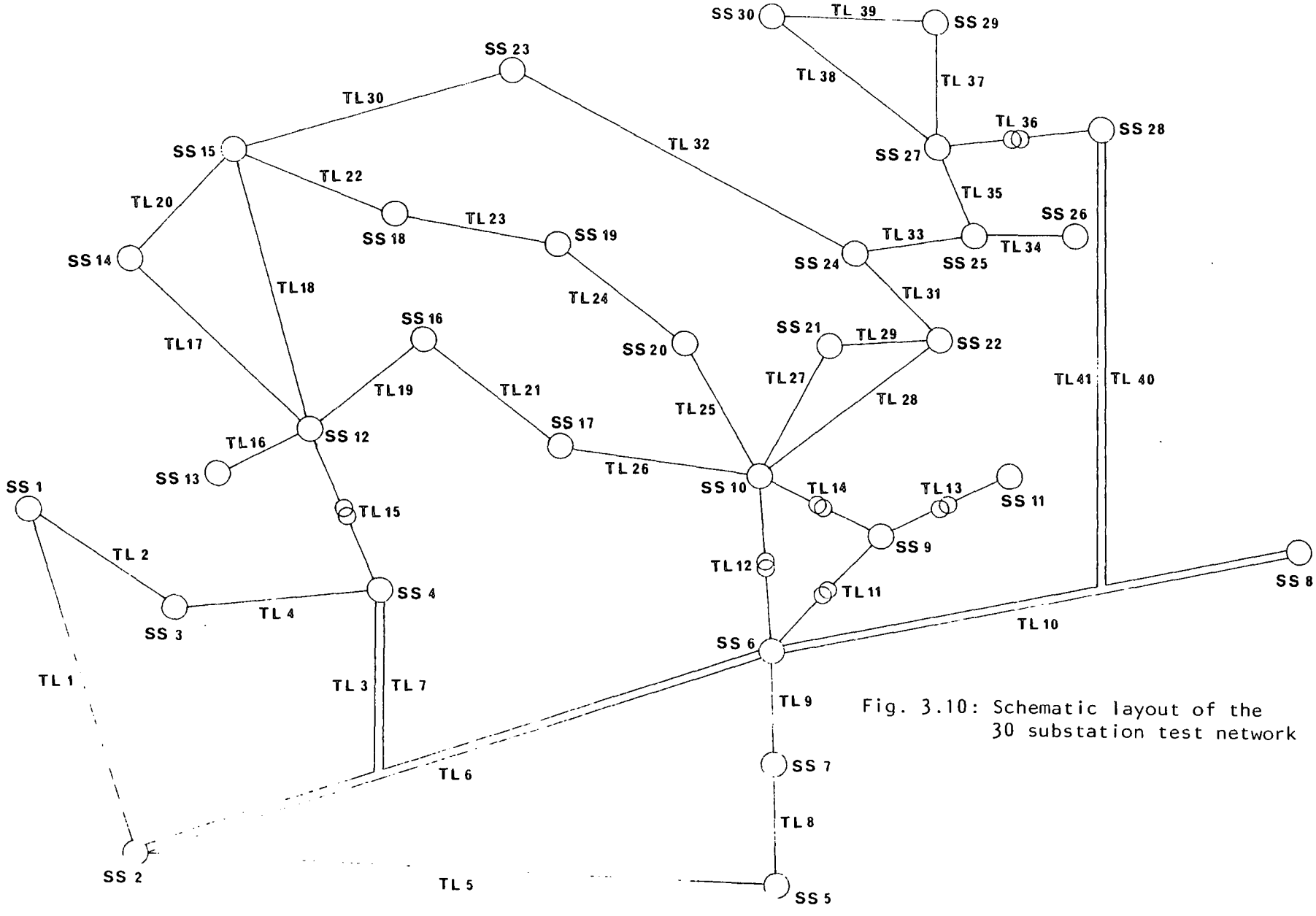


Fig. 3.10: Schematic layout of the 30 substation test network

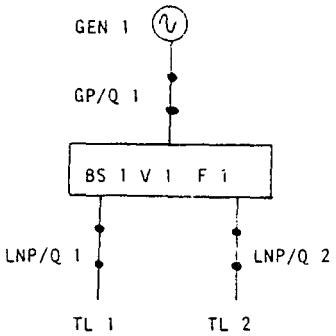


Fig. 3.11: SS 1

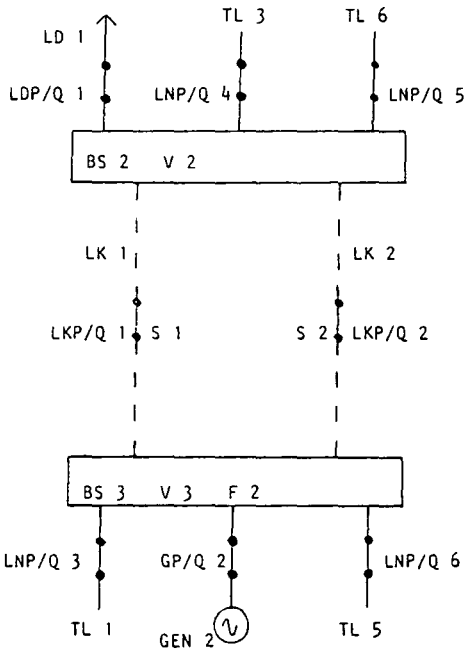


Fig. 3.12: SS 2

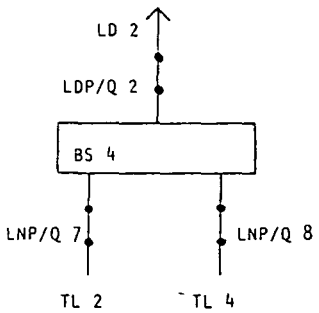


Fig. 3.13: SS 3

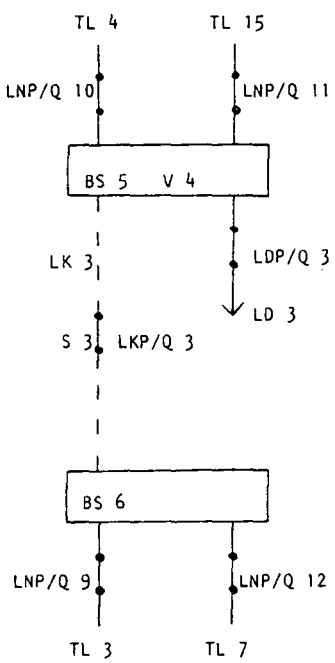


Fig. 3.14: SS 4

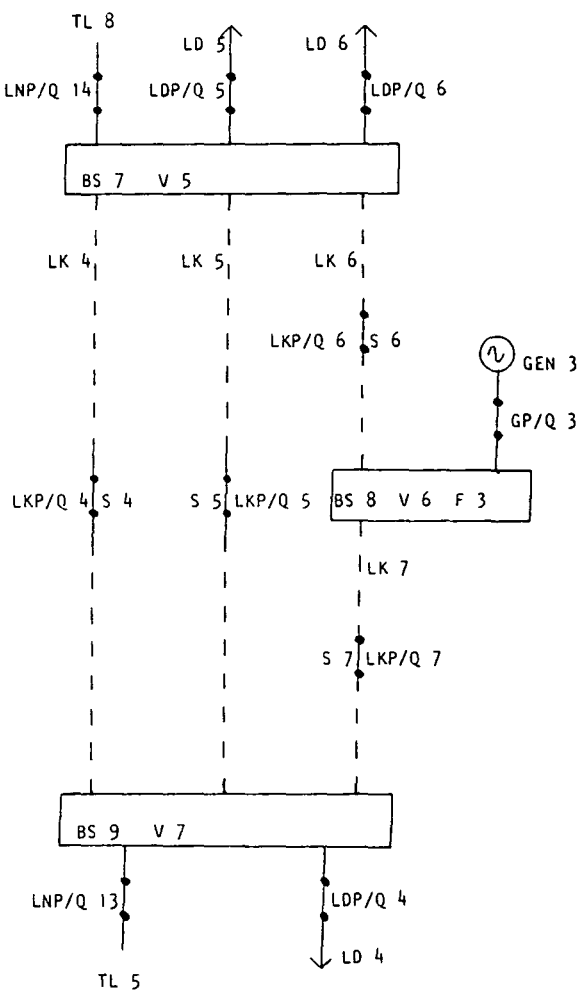


Fig. 3.15: SS 5

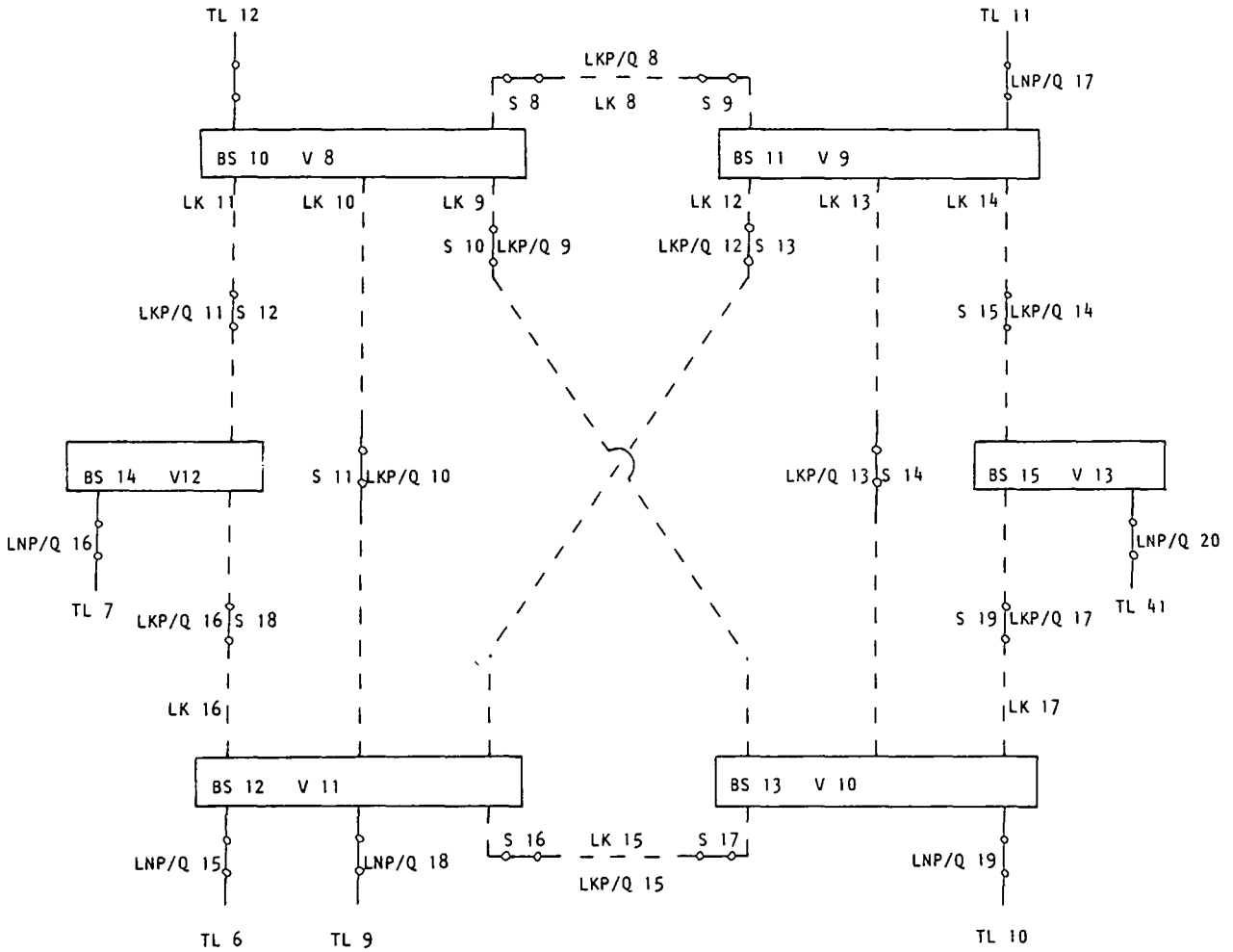


Fig. 3.16: SS 6

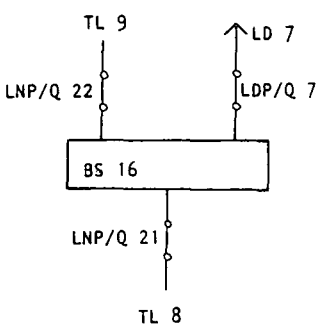


Fig. 3.17: SS 7

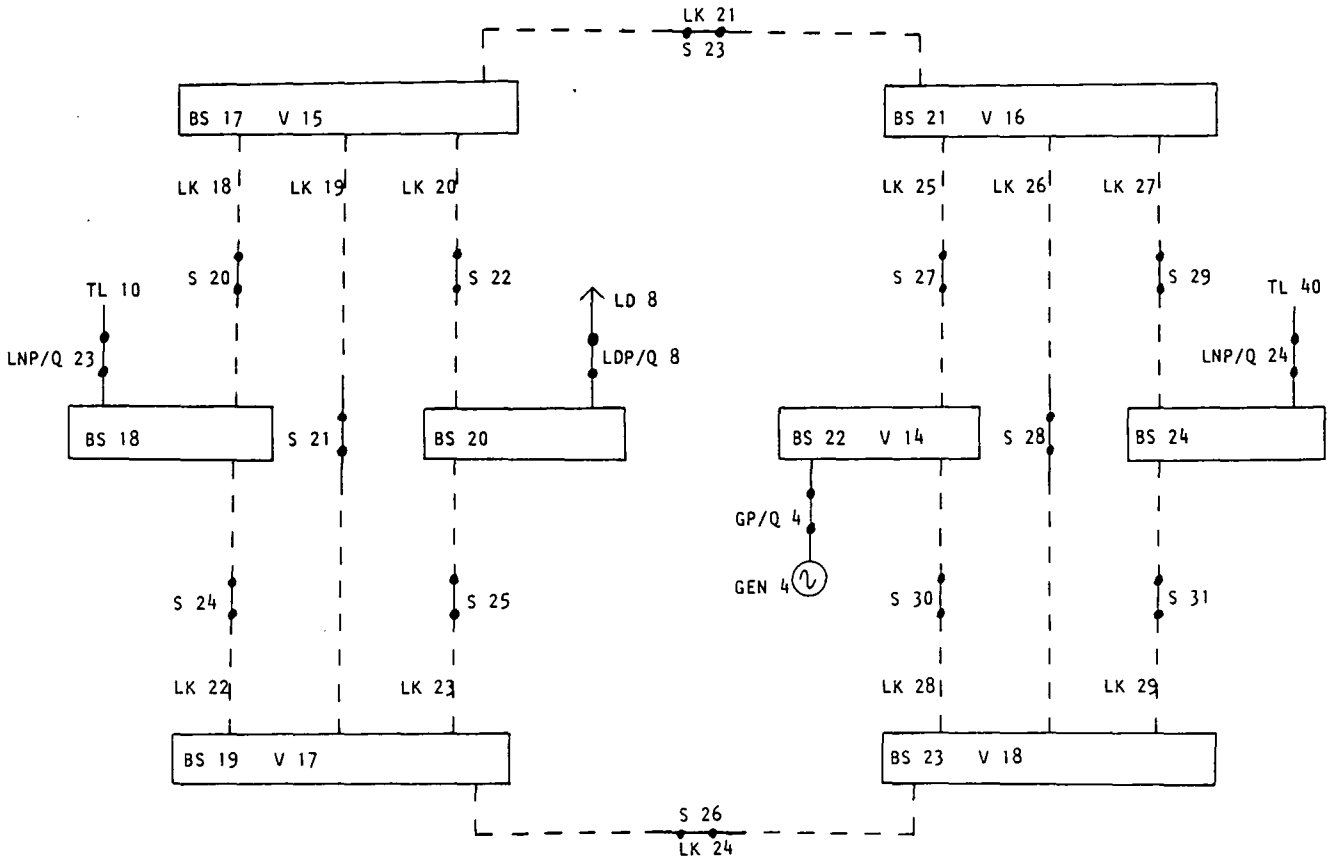


Fig. 3.18: SS 8

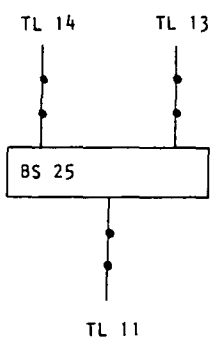


Fig. 3.19: SS 9

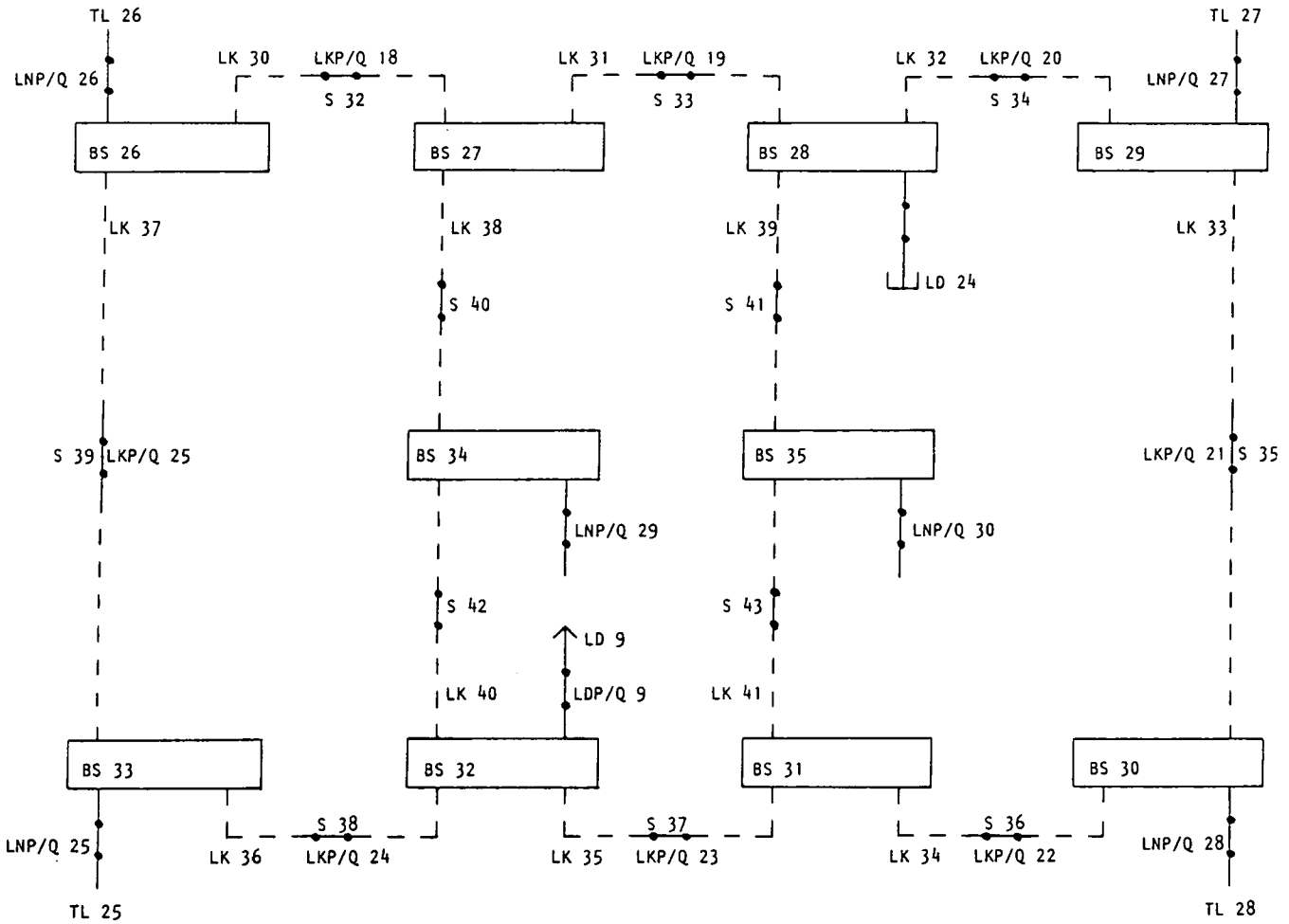


Fig. 3.20: SS10

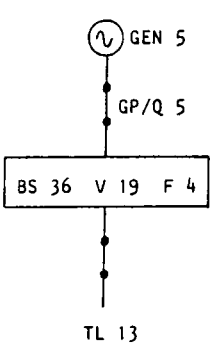


Fig. 3.21: SS 11

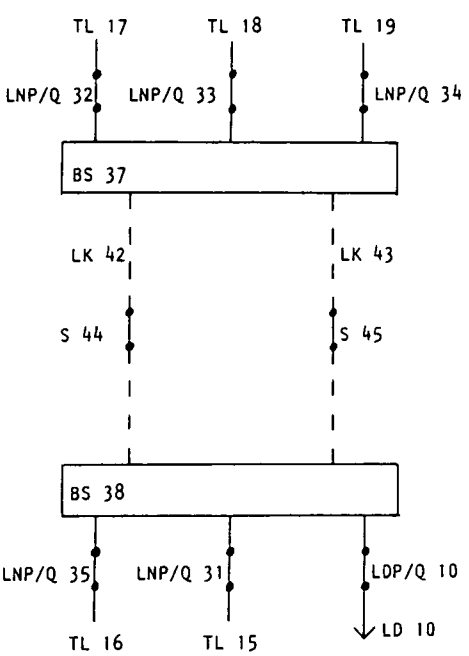


Fig. 3.22: SS 12

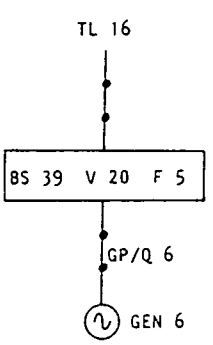


Fig. 3.23: SS 13

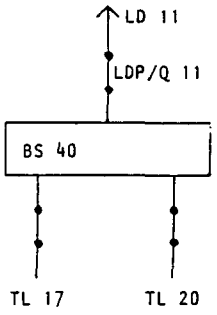


Fig. 3.24: SS 14

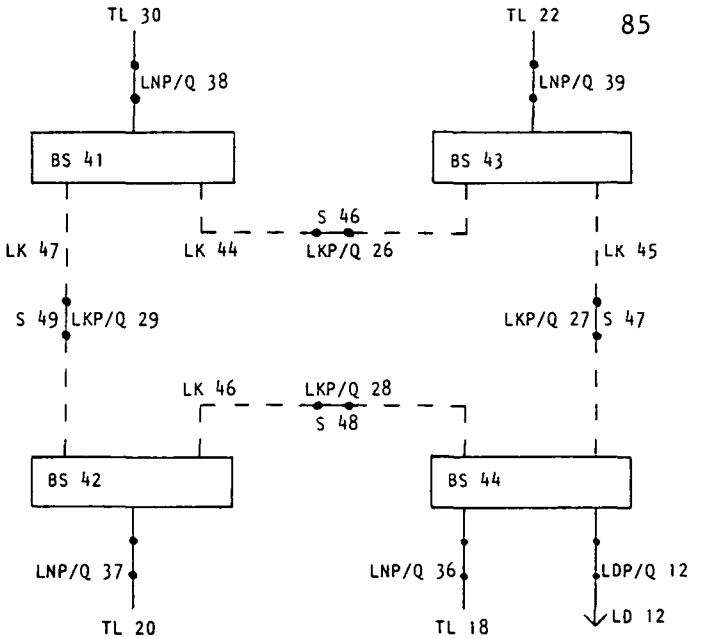


Fig. 3.25: SS 15

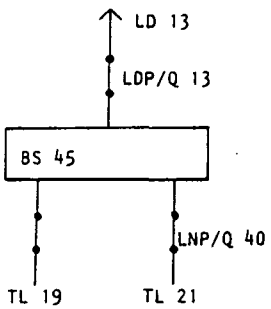


Fig. 3.26: SS 16

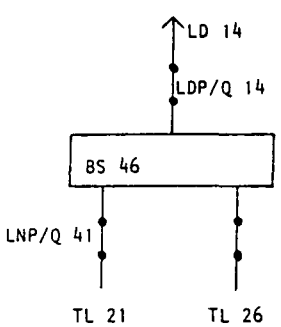


Fig. 3.27: SS 17

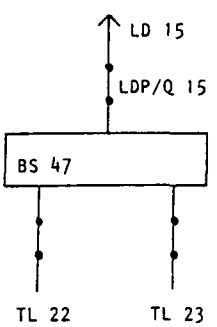


Fig. 3.28: SS 18

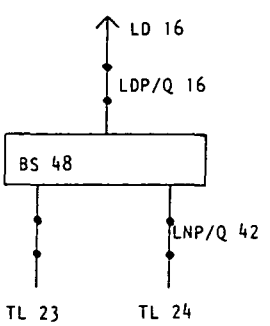


Fig. 3.29: SS 19

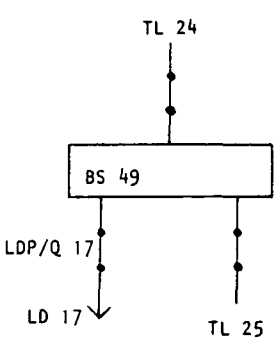


Fig. 3.30: SS 20

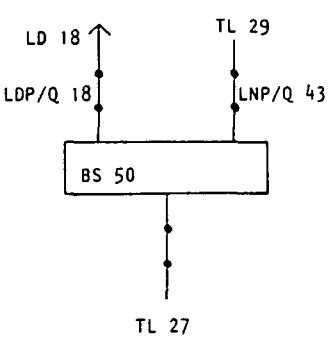


Fig. 3.31: SS 21

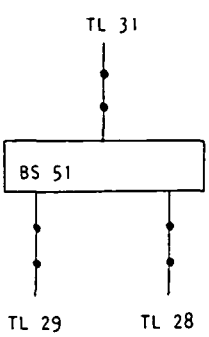


Fig. 3.32: SS 22

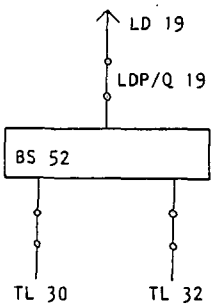


Fig. 3.33: SS 23

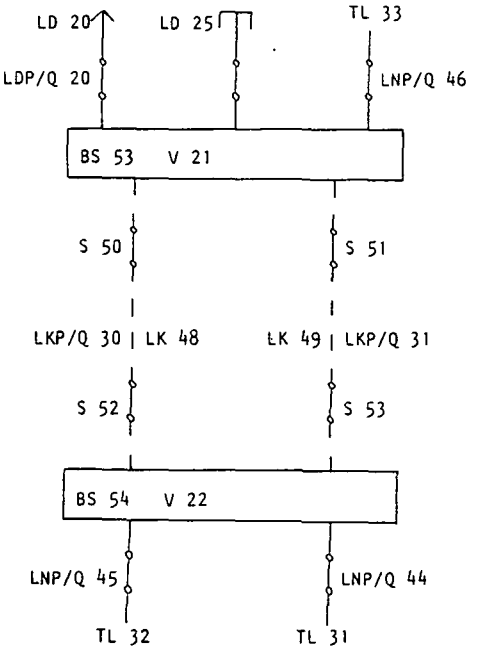


Fig. 3.34: SS 24

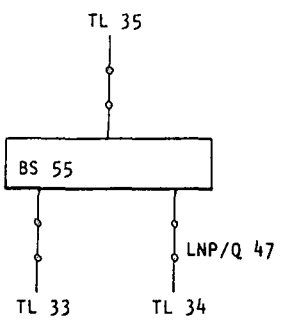


Fig. 3.35: SS 25

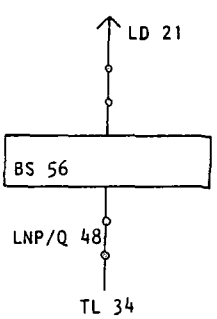


Fig. 3.36: SS 26

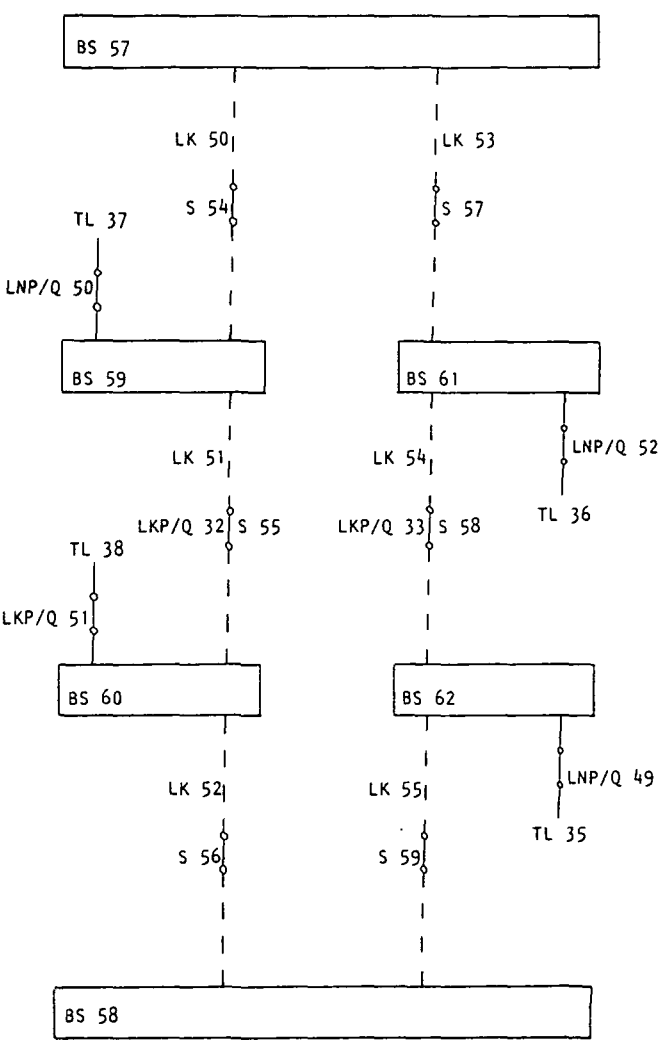


Fig. 3.37: SS 27

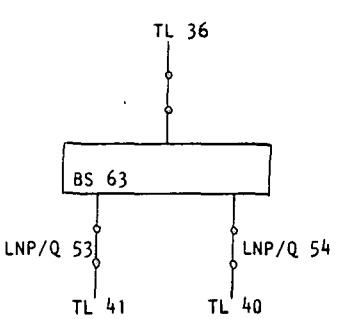


Fig. 3.38: SS 28

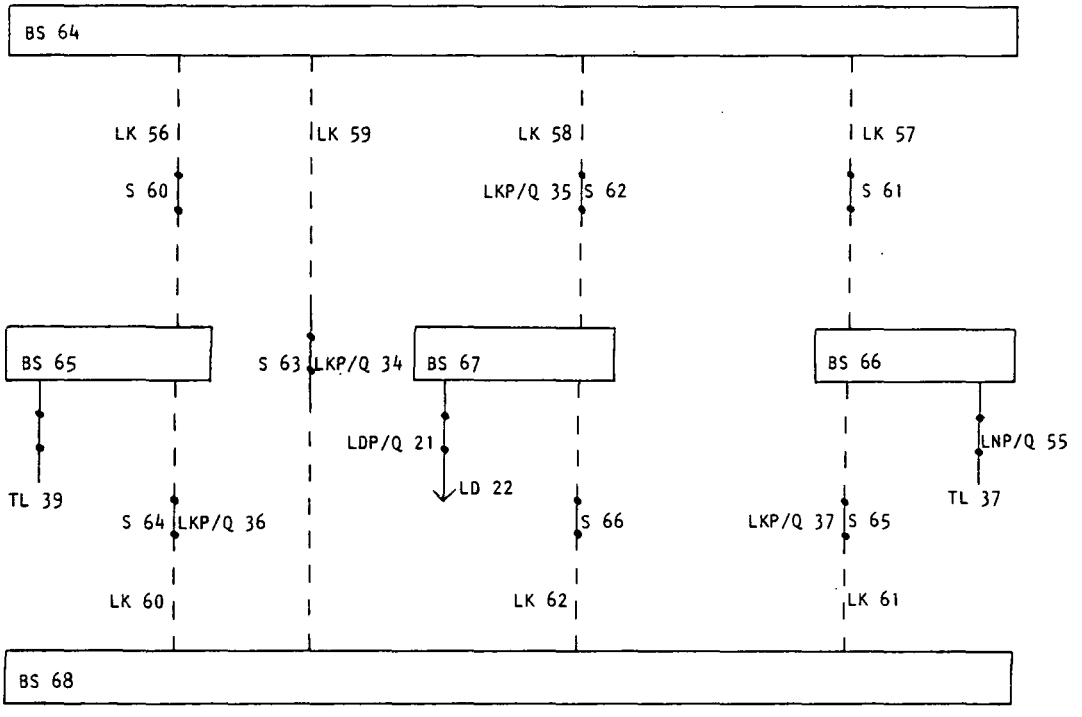


Fig. 3.39: SS 29

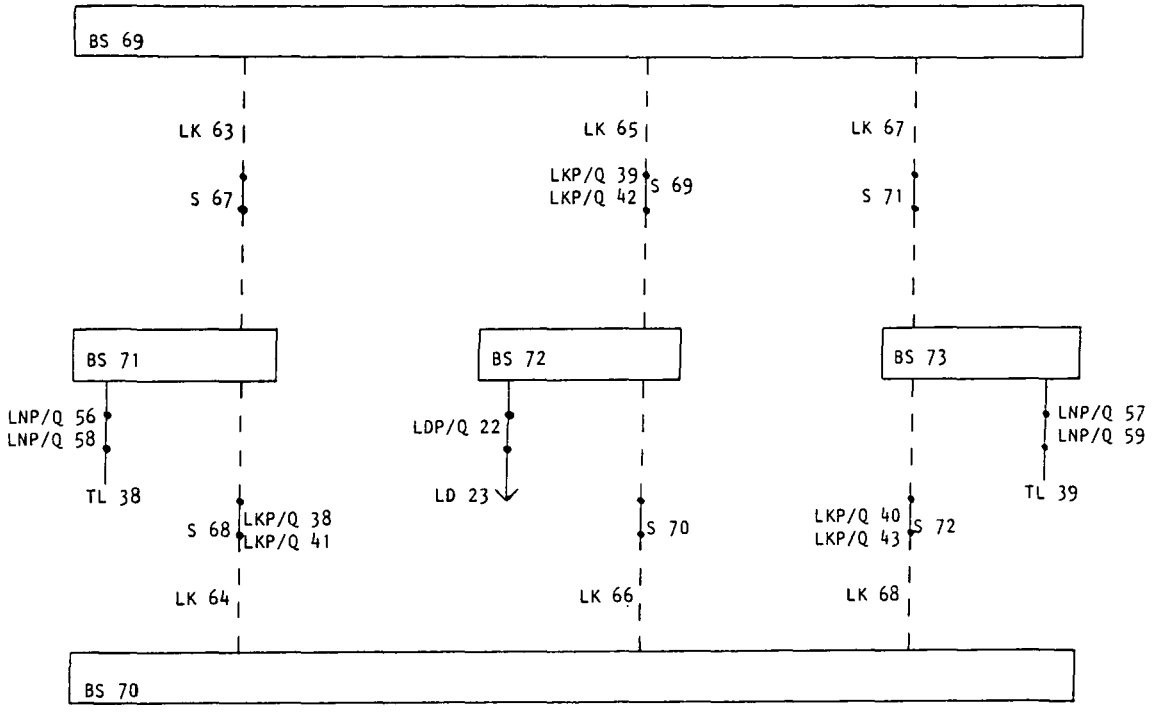


Fig. 3.40: SS 30

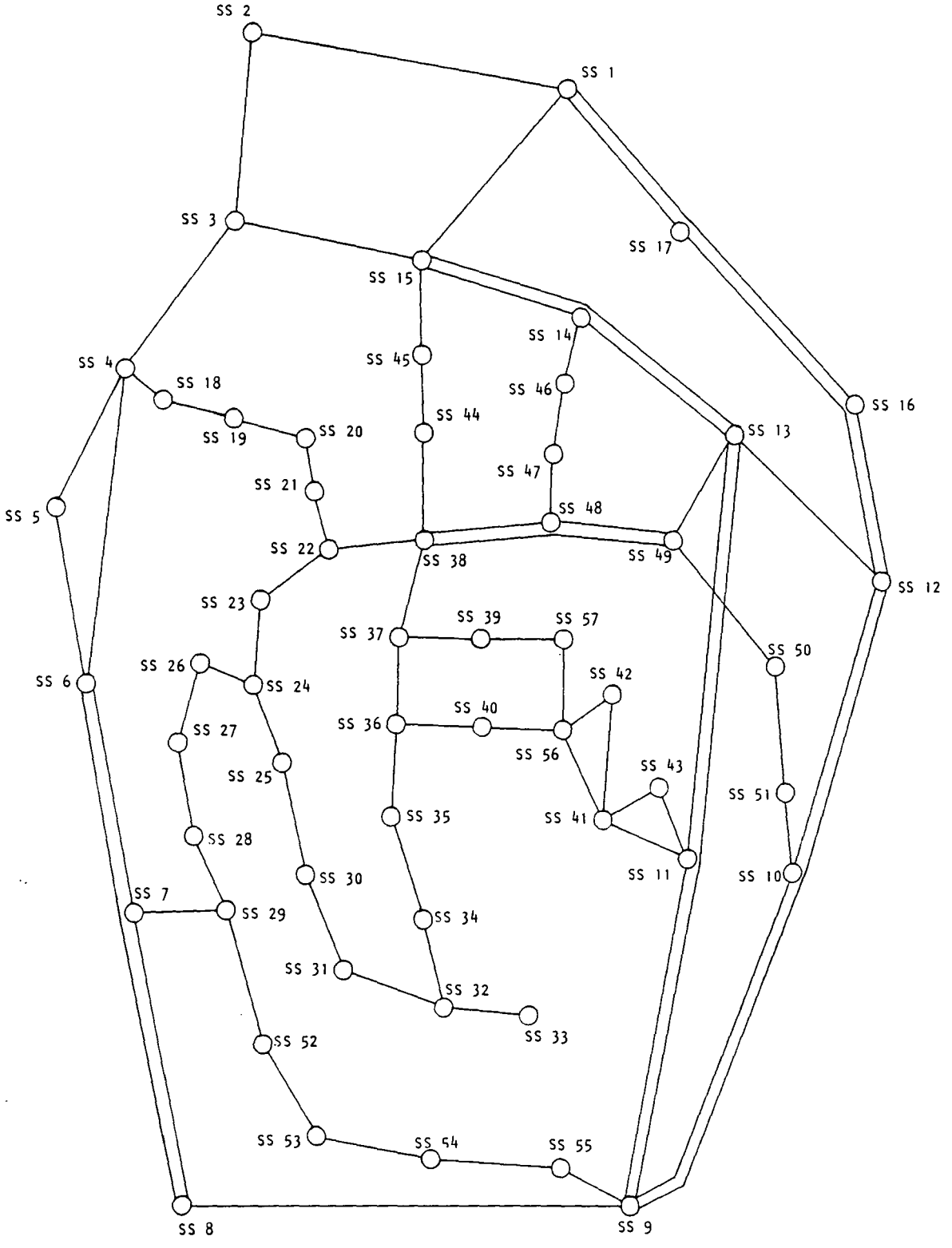


Fig. 3.41: Schematic layout of the 57 substation test network

together with a diagram showing the location of the substations and generators, both of which were obtained from the editor of the IEEE journal Power Apparatus and Systems. This standard network (together with the other IEEE standard networks mentioned in the thesis) have been made available by the American Electric Power Service Corporation,[122] published details of the 118 node network are not widely available but further details can be obtained from the Corporation.

The diagram had been photo reduced from a larger diagram and was hence difficult to read, the data for the 118 substation test network was therefore taken from the load flow results. This led to the omission of a generator on substation number 82 because the generation in the load flow solution for that substation was zero. The generator may be added at a later date but the omission causes no problems as the standard results show no generation at that point in any case. The diagram of the network has a number of generators on the edges of the network represented by a dotted symbol, the load flow results show the active power generation as being negative at some of these points. It has been assumed that these generators represent the lines to neighbouring power systems. In the 118 substation test network these generators have been retained as the mathematical model of a generator is also a good model for the representation of a power system provided that the parameters such as the rotational inertia and the steam constant etc. are modified accordingly. Unfortunately a number of the control programs expect the generators only supply active power to the network while the tie lines (and the mathematical model of a generator) allow for active power to flow into or out of the network. Thus the set points of these generators are presently set so that they supply active power to the network and hence the power flows and the voltage levels throughout the rest of the 118 substation test network have been changed from the standard values. A future enhancement to the data base would be to represent the tie line by their own device and model them in a similar way to the generators.

Figure 3.42 represents a schematic illustration of the geographical layout of the substations and full details of the network parameters are listed in appendix 1.

3.5 The simulation of the test networks

3.5.1 The simulator program

On-line control programs are generally concerned with controlling events

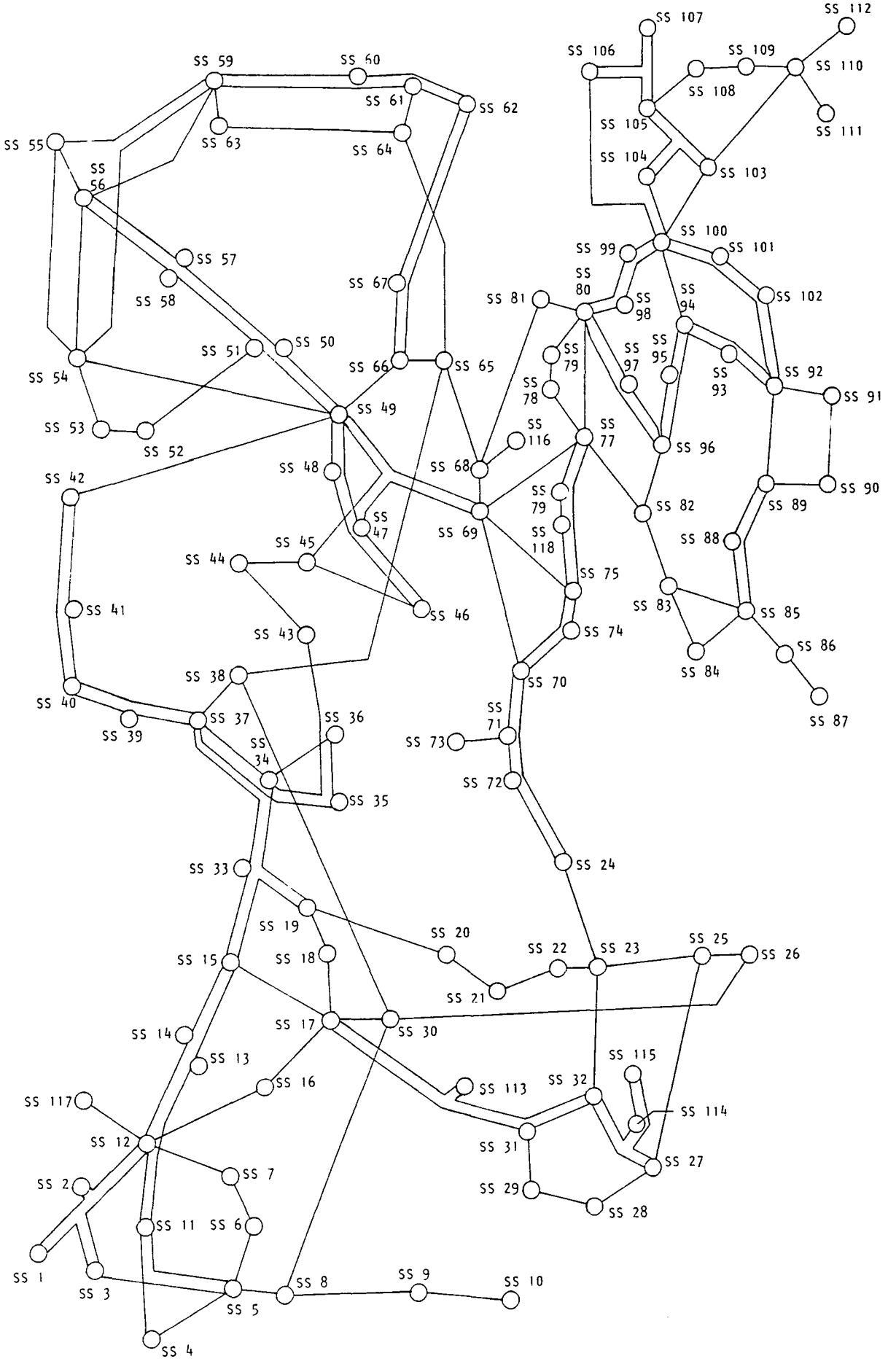


Fig. 3.42: Schematic layout of the 118 substation test network

which happen on a time scale greater than a few seconds. Events on a time scale less than this such as frequency and voltage regulation are controlled by closed loop control devices and are beyond the present scope of modern computing and telemetry equipment to be included in a computerised control centre. The simulator program therefore needs to simulate the behaviour of the network on a time scale of approximately one second. The transient behaviour of the network on a smaller time scale would prove to be too great a burden on the computer thus making the real time simulation infeasible. The current time step for the simulator has been set to one second which means the simulator updates the values of the states of the system, i.e. the frequency, the voltage levels and the power flows, in discrete intervals of one second. It should be noted that in some cases for large systems, it may take the computer longer than one second to calculate the results but the updates are still made for one second time steps of power system time, ie. in these circumstances the simulator is not running in real time. The time step of one second gives good results for the dynamic behaviour of the network whilst allowing the mathematical model of the network to remain stable and the program to converge to the new solution in only a few iterations. The priorities of the programs running in the computer have been arranged such that the simulator has absolute priority for the length of time required to calculate the results for the next update of the states of the system. The control programs then share on an equal priority one second of CPU time. Thus the computer, in effect, devotes approximately half the CPU time to simulating the network and half the time to the control programs when operating on the 30 substation test network.

The simulator works at a nodal level and has a small topology program devoted to it forming a nodal list of the devices which are currently active in the network. The simulator requires the state values from the previous time step or values read from a file on the first time step as the starting point for the solution of the differential and algebraic equations for the current time step. Implicit trapezoidal integration[35] is used to evaluate the solution to the generator differential equations which are then solved simultaneously with the generator and network algebraic equations. The solution of the simultaneous equations is a Newton Raphson based iterative process which uses the Harwell library subroutines LA05A and LA05B to return the solution of the linearly independent set of equations formed in each iteration. An outline of the Newton Raphson method has been presented in chapter 2, details of the formation of the Jacobian matrix can be found in appendix 2 together with a summary of the implicit trapezoidal integration

method.

The main section of the program solves the differential and algebraic equations in terms of the following variables.

1. The electrical angular position of the generator rotor.
2. The rate of change of the electrical angle of the generator.
3. The mechanical power input to the generator.
4. The electrical power output of the generator.
5. The rectangular components of the complex generator terminal current.
6. The rectangular components of the complex voltage behind the transient reactance of the generator.
7. The rectangular components of the complex nodal voltages throughout the network.

The nodal voltages from the main section of the program are mapped on to the bus-sections as follows. All the active bus-sections, ie those with an active line or link connected to them, will have been assigned a node number by the topology program. The voltage levels of the bus-section are therefore equal to the nodal voltage levels.

Two small sections of code then evaluate the power flows in the loads, lines and links from the voltage levels and the network parameters. The load power flows are calculated from the voltage magnitude of the load bus-section and the equivalent shunt admittance calculated at the previous time step, ie equations 3.23 and 3.24 are re-arranged so that the power flow terms are on the left hand side of the equations. Note this gives the current load power flows, the load demand is stored in the data base and it is this demand value and the current load bus-section voltage magnitude which are used to calculate the equivalent shunt admittance for the next time step. The line power flows are calculated from the line bus-section voltages and the line parameters using equations 3.9 and 3.10.

The calculation of the link power flows is in itself a small estimation problem. It has been assumed that all the links in a substation have negligible impedance and would hence carry equal power flows. However the sum of the link power flows at a bus-section must equal the net power injection at that bus-section from the generators, loads and lines. The problem may be defined mathematically by a set of linear equations which equate the sum of the link power flows at a bus-section to the net injection. The set of linear equations formed for the calculation of the link power flows at a substation where there are more links than bus-sections will be an under-determined

problem.

A solution can still be obtained if the assumption is made that the link power flows tend to a minimum. This assumption is fairly realistic in that although the impedance of the branches have been assumed to be equal to zero, in reality each link will have a small impedance which will restrict the flow of power. The impedances of all the branches however are still assumed to be the same.

A least squares method may be used to solve the link power flow estimation problems, provided the method does not explicitly calculate the inverse of the coefficient matrix and obtains the solution by successively moving from one solution to the next. In this case the initial solution of the link power flows is set to zero and the least squares method then moves from this solution point to the solution point which satisfies the power flow sum check equations at each bus-section. This ensures that the link power flows have the minimum possible values which satisfy both the sum check equations and the assumption that the impedance of the links are equal but very small. Further details on the least squares method used can be found in chapter 5.

The sequence of events taken in each time step may be summarised as below:

1. Initialisation.
2. Form a set of linearly independent equations defining the change in the state variables of the generators and the network in terms of the partial derivatives (stored in a Jacobian matrix) of both the equations defining the solution of the generator differential equations for this time step and the equations defining the behaviour of the network.
3. Check if the values of the Jacobian matrix are less than a pre-set tolerance, if so convergence has been obtained go to step 7.
4. Solve the linear set of equations using the Harwell subroutines.
5. Add the values of the change of the state variables to the state variables, if all the changes are below a pre-set tolerance convergence has been obtained, go to step 7.
6. Go to step 2.
7. Evaluate the new parameters to represent the loads and update the magnitude of the voltages behind the transient reactances for all the generators for the next time step.
8. Calculate the present load and line power flows.
9. Calculate the link power flows.
10. Output the results.
11. Wait one second then goto step 2.

The time consuming parts of each iteration are the formation of the Jacobian matrix and the solution of the linear set of equations, ie steps 2 and 4, hence a check for convergence is made after each of these steps to terminate the iterative process as quickly as possible.

3.5.2 Solution times and results

The total CPU time used by the simulator for a sequence of nine time steps for the 5, the 30, the 57 and the 118 substation test networks are listed in tables 3.2 to 3.5 respectively. Where the simulator has been allowed to run then no change has been made to the topology of the network. However the dynamic behaviour of the generators may cause the power flows throughout the network to change from one step to another. The greater the change in the generator variables such as the electrical angular position of the rotor, or the electrical power output then the larger the number of iterations and hence the solution time required to obtain convergence. The iteration count is incremented at the start of each iteration, hence if convergence is obtained after the formation of the Jacobian matrix the simulator is deemed to have taken a full iteration. Whenever the simulator is started a small initialisation section is used to read in the starting values for the states of the network, the CPU time for this section is listed at the head of tables 3.2 to 3.5. The variations in the CPU time required for a given number of iterations arises from the point at which convergence was obtained, ie whether it was after the formation of the Jacobian matrix or after the solution of the linear set of equations.

The solution times for the 5 and the 30 substation test networks include the CPU time taken to calculate both the active and reactive link power flows. The CPU time required for this section is fairly constant whatever the rate of change of the rest of the network state variables. The CPU time for the calculation of both the active and reactive link power flows for the 30 substation test network varies between 0.380 and 0.440 seconds and takes between 22 and 26 iterations of the least squares subroutine (between 11 and 13 iterations for each of the calculations).

During normal operation of the power system the simulator usually takes two iterations to converge the state variables. The opening and closing of the line circuits does not cause the generators to oscillate significantly and hence has little effect on the solution times. A step change in the load demand or the generation caused by the status of switch changing does cause

Table 3.2: Simulator CPU. times for the 5 substation network

Initialisation = 0.090 seconds

Event	Solution time/seconds	No. of iterations
Start	0.061	1
Run	0.061	1
Run	0.062	1
Open line 4	0.181	2
Run	0.294	2
Run	0.286	2
Open load 1	0.291	3
Run	0.291	3
Run	0.406	4

Table 3.3: Simulator CPU. times for the 30 substation network

Initialisation = 0.339 seconds

Event	Solution time/seconds	No. of iterations
Start	1.117	1
Run	1.118	1
Run	1.117	1
Open line 7	1.157	2
Run	1.164	2
Run	1.155	2
Open load 8	1.848	3
Run	1.826	3
Run	2.459	3

Table 3.4: Simulator CPU. times for the 57 substation network

Initialisation = 0.426 seconds

Event	Solution time/seconds	No. of iterations
Start	1.748	1
Run	0.267	1
Run	0.268	1
Open line 1	3.005	2
Run	1.652	2
Run	1.701	2
Open load 35	1.658	2
Run	3.062	2
Run	3.037	3

Table 3.5: Simulator CPU. times for the 118 substation network

Initialisation = 1.321 seconds

Event	Solution time/seconds	No. of iterations
Start	8.594	1
Run	0.763	1
Run	0.754	1
Open line 50	8.651	2
Run	8.755	2
Run	8.694	2
Open load 7	16.585	2
Run	16.708	3
Run	16.746	3

the generators to oscillate and the number of iterations required to obtain a solution increases to four or five. A limit of six iterations has been placed on the program because if the state variables have not converged by this time then their values are likely to be unrealistic and the simulator has gone into an unrecoverable state as is the case when a real power system undergoes too great a step change. The number of iterations of the simulator gradually decreases to two after a step change as the oscillations of the generators die away.

The simulator has recently had an additional piece of code added to it which writes into a reserved area of memory known as a task common block the voltage levels and power flows throughout the network after each time step. See section 3.5.3 for further details on task common blocks. The contents of the task common block may be displayed on a screen or printed out for use in debugging and testing the control programs. The use of the task common blocks reduces the time needed to output the results since the output only requires memory accesses. The additional code has increased the solution time for the 30 substation test network under normal operating conditions to slightly above one second, thus a flag may be added at a later stage to suppress the output if it is not required. A separate program with a lower priority handles the time consuming output of the contents of the task common block to the screen or the printer.

The steady state starting values of the network voltage levels and power flows for the 5 substation test network are listed in table 3.6. Table 3.7 lists the same set of variables immediately after load number one has been disconnected and table 3.8 lists the values after the next time step. These tables illustrate the oscillatory nature of the generators. The steady state starting values for the three larger networks are listed in appendix 3.

The oscillations in the active power output of the generators have been illustrated in tables 3.6, 3.7 and 3.8. However they can be seen more clearly in figure 3.43 in which the active power output of both generators on the 5 substation test network have been plotted against time. The results are plotted over a 15 second time period with load 1 being disconnected after 2 seconds. The graph illustrates the general trend of the generator outputs. However with a time step of one second the high frequency transient oscillations are lost. Reducing the time step of the simulator to 0.125 seconds gives a clearer indication of the transient oscillations, the generator active power outputs for the same time period as in figure 3.43 have been plotted in figures 3.44. Ideally an even smaller time step would be required to accurately plot the power outputs but to enable the graphs to be

Table 3.6: Initial voltage levels and power flows for the 5 substation test network

Values are in P.U.
V = voltage magnitude, θ = voltage phase angle,
P = active power flow, Q = reactive power flow,
R = receiving end of a line, S = sending end of a line.

Voltage levels.

Bus No.	Node No.	V	θ
1	1	1.0600	0.0000
2	2	1.0242	-0.0872
3	2	1.0242	-0.0872
4	3	1.0236	-0.0930
5	4	1.0475	-0.0490
6	4	1.0475	-0.0490
7	4	1.0475	-0.0490
8	4	1.0475	-0.0490
9	5	1.0179	-0.1073

Generator power flows.

Gen No.	P	Q
1	1.2956	-0.0748
2	0.4000	0.3000

Load power flows.

Load No.	P	Q
1	-0.2000	-0.1000
2	-0.4500	-0.1500
3	-0.4000	-0.0500
4	-0.6000	-0.1000

Link power flows.

Link No.	P	Q
1	0.6388	0.0980
2	0.2267	-0.0169
3	0.2069	0.0007
4	0.1939	-0.0809
5	-0.0198	0.0176
6	-0.0328	-0.0640
7	-0.0129	-0.0816

Line power flows.

Line No.	P (S)	Q (S)	P (R)	Q (R)
1	-0.8884	0.8743	0.0862	-0.0619
2	-0.4071	0.3952	-0.0114	0.0300
3	0.2434	-0.2469	0.0678	-0.0354
4	0.2749	-0.2794	0.0593	-0.0296
5	-0.5483	0.5371	-0.0735	0.0718
6	-0.1890	0.1886	0.0518	-0.0319
7	-0.0634	0.0631	0.0228	0.0284

Table 3.7: Voltage levels and power flows for the 5 substation immediately after load number one opened

Values are in P.U.
V = voltage magnitude, θ = voltage phase angle,
P = active power flow, Q = reactive power flow,
R = receiving end of a line, S = sending end of a line.

Voltage levels.

Bus No.	Node No.	V	θ
1	1	1.0902	0.0000
2	2	1.0582	-0.0829
3	2	1.0582	-0.0829
4	3	1.0579	-0.0883
5	4	1.0839	-0.0430
6	4	1.0839	-0.0430
7	4	1.0839	-0.0430
8	4	1.0839	-0.0430
9	5	1.0530	-0.1018

Generator power flows.

Gen No.	P	Q
1	1.2087	-0.1740
2	0.3853	0.2948

Load power flows.

Load No.	P	Q
1	0.0000	0.0000
2	-0.4500	-0.1500
3	-0.4000	-0.0500
4	-0.6000	-0.1000

Link power flows.

Link No.	P	Q
1	0.6523	0.0931
2	0.2058	-0.0351
3	0.1287	-0.0442
4	0.1802	-0.0976
5	-0.0771	-0.0092
6	-0.0256	-0.0626
7	0.0516	-0.0534

Line power flows.

Line No.	P (S)	Q (S)	P (R)	Q (R)
1	-0.8021	0.7910	0.1704	-0.1329
2	-0.4066	0.3954	0.0036	0.0206
3	0.2721	-0.2763	0.0775	-0.0440
4	0.3035	-0.3085	0.0674	-0.0366
5	-0.5915	0.5793	-0.0812	0.0787
6	-0.1871	0.1868	0.0620	-0.0406
7	-0.0630	0.0627	0.0266	0.0282

Table 3.8: Voltage levels and power flows for the 5 substation on the second time step after load one opened

Values are in P.U.
V = voltage magnitude, θ = voltage phase angle,
P = active power flow, Q = reactive power flow,
R = receiving end of a line, S = sending end of a line.

Voltage levels.

Bus No.	Node No.	V	θ
1	1	1.0685	0.0000
2	2	1.0405	-0.0784
3	2	1.0405	-0.0784
4	3	1.0403	-0.0835
5	4	1.0639	-0.0410
6	4	1.0639	-0.0410
7	4	1.0639	-0.0410
8	4	1.0639	-0.0410
9	5	1.0358	-0.0961

Generator power flows.

Gen No.	P	Q
1	1.0959	-0.1915
2	0.3442	0.2740

Load power flows.

Load No.	P	Q
1	0.0000	0.0000
2	-0.4500	-0.1500
3	-0.4000	-0.0500
4	-0.6000	-0.1000

Link power flows.

Link No.	P	Q
1	0.6120	0.0878
2	0.1869	-0.0368
3	0.1172	-0.0442
4	0.1647	-0.0957
5	-0.0697	-0.0074
6	-0.0222	-0.0589
7	0.0475	-0.0515

Line power flows.

Line No.	P (S)	Q (S)	P (R)	Q (R)
1	-0.7280	0.7183	0.1791	-0.1399
2	-0.3680	0.3585	0.0124	0.0147
3	0.2460	-0.2495	0.0706	-0.0368
4	0.2744	-0.2787	0.0610	-0.0295
5	-0.5343	0.5239	-0.0678	0.0699
6	-0.1695	0.1692	0.0597	-0.0389
7	-0.0569	0.0567	0.0263	0.0269

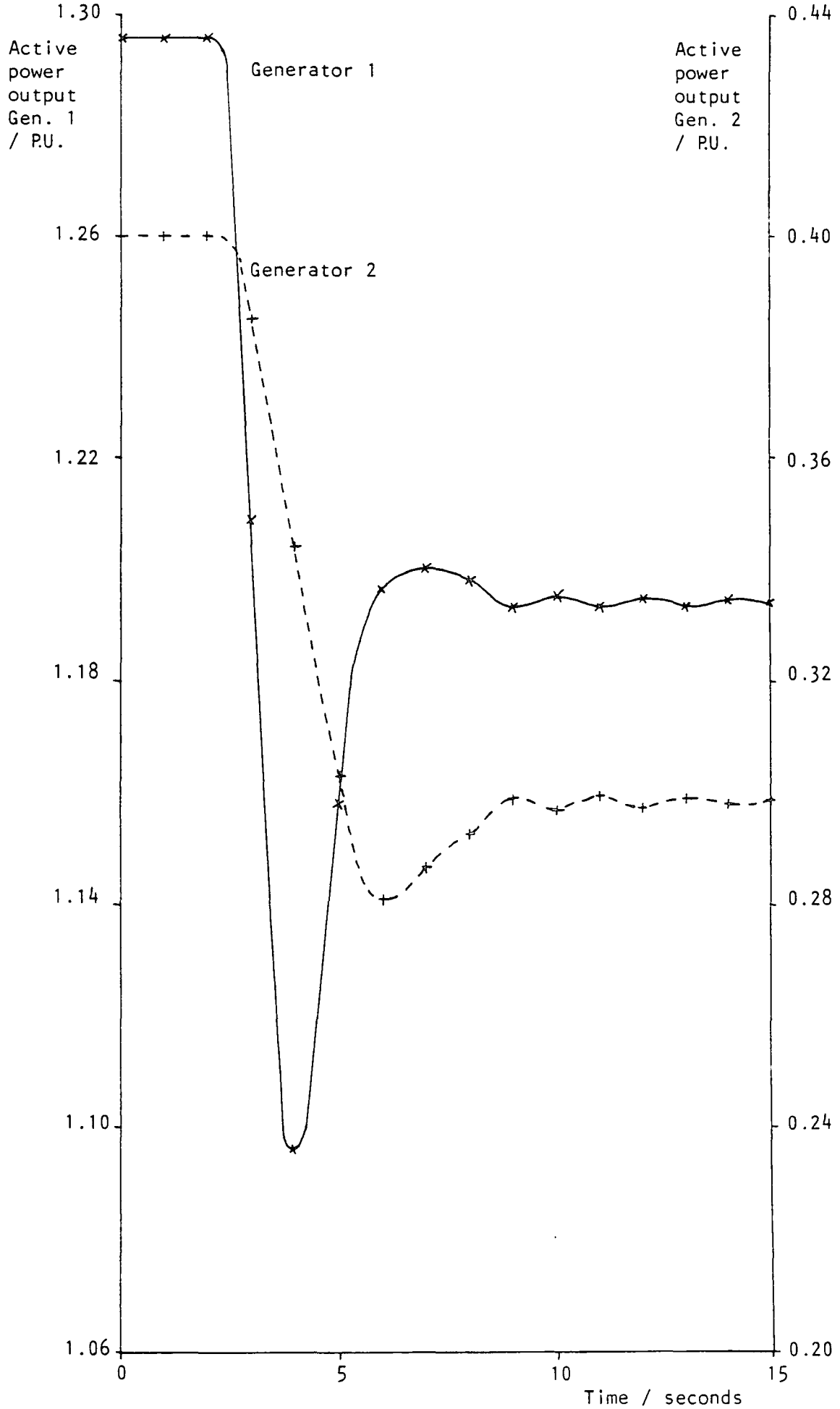


Fig. 3.43: Transient oscillations of the generators using a 1.0 second time step

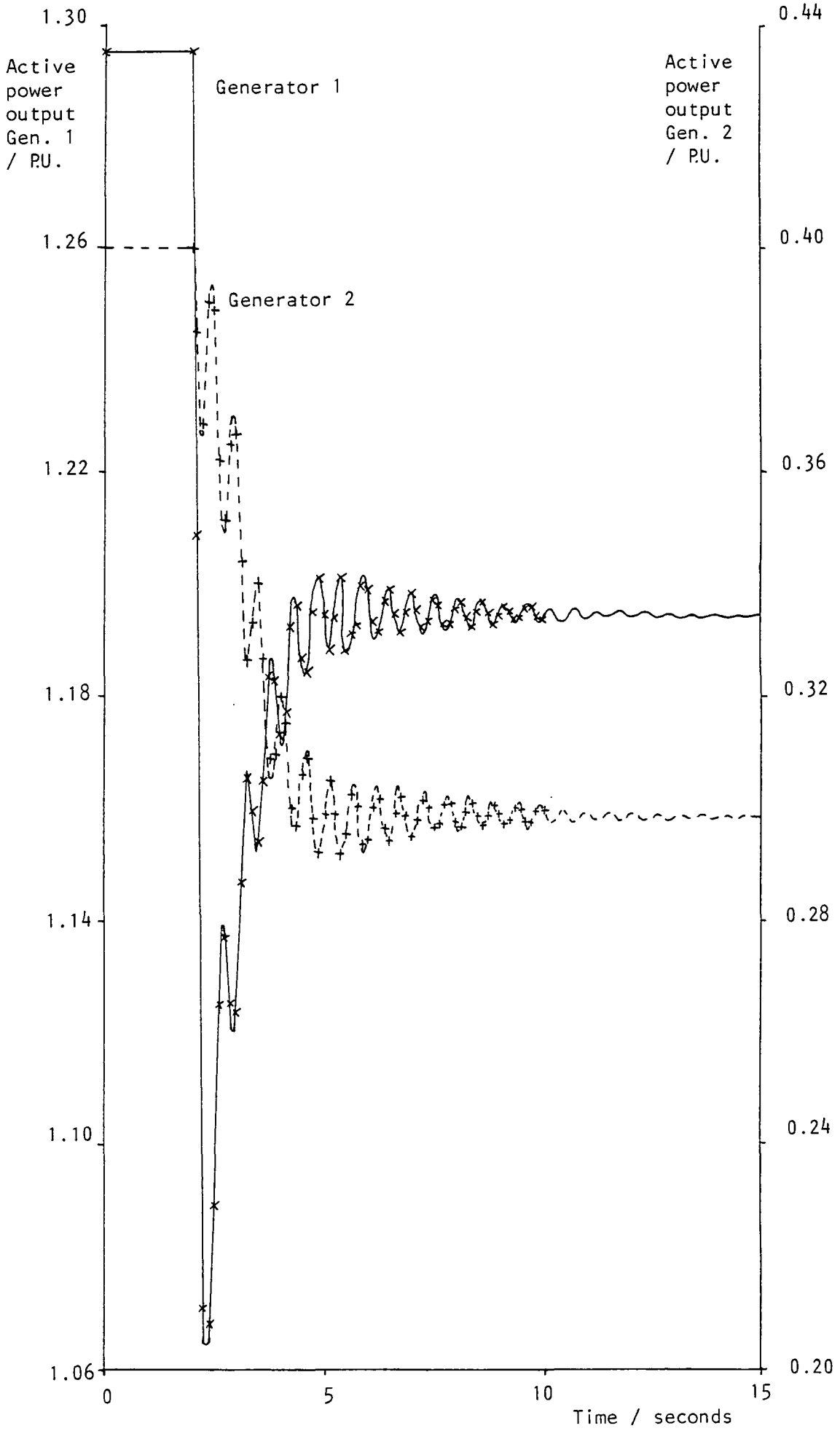


Fig. 3.44: Transient oscillations of the generators using a 0.125 second time step

more easily compared the same scales have been used in both cases. To help clarify the diagram the points on the curve where the rate of change of the power output is small have been omitted. The time step on the simulator is set at 1 second because the control programs are generally only interested in the behaviour of the network for time scales in the order of a few seconds and also because the computer does not have sufficient power to simulate the network with a time step of 0.125 seconds in real time.

3.5.3 Inter-program communication and measurement output

The on-line power system control suite consists of a number of programs designed to perform different functions. Each program has been structured so that it is able to run on its own and does not interact directly with any other program. However data communication is essential between the programs not only to provide the relevant data on the present state of the system but also to coordinate the sequence of the running of the programs in certain circumstances such as the change of state of a switch. The data communication method adopted uses reserved section of memory which are referred to as task common blocks. Each program has access to the relevant sections of the memory and is able to update the data at any time. Should a program fail to update the data for any reason then any programs dependent on the output from this program will continue to run on the old data although the results will become more and more unreliable as time goes on. This method of communication is very fast requiring only a small amount of CPU time to address the relevant areas of memory and is also very easy to build into the programs.

The sections of memory are reserved before any of the programs are loaded and each section reserved is given a unique name. The programs treat the reserved sections of memory as common blocks, referring to them by the unique names and using the variables contained in them in the same manner as the variables in a standard Fortran common block. Any subroutine of the programs thus may have access to the data by including the common statement at the top of the subroutine. Ordinarily any number of programs could simultaneously read or write the data in the task common blocks. However should a program be timed out while it is updating the data a second program may then read all the data and end up with a skew set of data, ie one half of the data is from the previous update while the other half is the most recent update. The following protocol has been adopted to help prevent this from occurring. Each task common block has a flag which is set to one by any program which requires access to the data and is set to zero once the program has finished with the

data. No other program may gain access to the data while the flag is set to one. However as a precautionary measure if the flag is not reset to zero after five seconds, then something is deemed to be wrong. An error message is printed and the program gains access to the data. The programs keep a copy of all the date they use in all the common blocks in their own local storage areas and transfer the data from the task common block to the local storage or vice versa once they have gained access to the common block. This keeps the time taken to update the data to a minimum and allows other programs quick access to the data.

The values of all the measurements in the test networks are stored in a task common block and are updated at every time step of the simulator. To simulate the effects of transducer mis-calibration (a systematic error) and errors introduced by the telemetry equipment (random noise) the simulator may distort the measurement values by a user defined percentage which is entered during the initialisation step of the simulator.
The value of the measurement is calculated from the following equation

Measurement = True value + Systematic error + Random noise (3.25)

The value of the systematic error is set for each measurement during the initialisation stage according to the equation.

Systematic error = Syserr * Rnorm * Full scale reading (3.26)

where
Syserr = user defined percentage error, a typical value = 0.2%
(entered as 0.2 not 0.002).
Rnorm = a random number created by a random number generation.

The numbers have a normal distribution with a mean of zero and a standard deviation of one.

The full scale meter readings have been set as follows:

1.5 P.U. for voltage magnitude measurements
54 Hz for frequency measurements
1.2 P.U./0.2 P.U. for generator active/reactive power flow measurements
0.2 P.U./0.1 P.U. for load activereactive power flow measurements
0.1 P.U./0.05 P.U. for link active/reactive power flow measurements

The random noise component of the measurement is calculated from the equation

$$\text{Random noise} = \text{Randerr} * \text{Rnorm} * \text{True value} \qquad (3.27)$$

where

Randerr = user defined percentage level noise, a typical value = 3.0% (entered at 3.0 not 0.3).

Rnorm = as above.

It should be noted that the values for the systematic errors are set during the initialisation stage and are not subsequently altered. However the random noise components is re-calculated at every time step and hence changes according to the new values of the random number. The random number generator calculates the same series of numbers every time the simulator is run so some degree of repeatability is possible if this is desired. It has been found necessary to divide the values of Syserr and Randerr by ten when calculating the frequency measurements noise components in order to reflect the accuracy of the frequency measurements available to power system operators.

A small separate program has been written to allow the user to interact with the simulator and select up to 10 analogue measurements which will have a gross measurement error applied to them. A gross measurement error may be used to simulate either a total transducer failure or a mal-function of the telemtry equipment. The user selects which of the analogue measurements points are to be corrupted and enters a multiplication factor for each of the chosen measurement points. The value of each of the measurements, after the addition of the noise components is multiplied by the corresponding multiplication factor.

Thus to simulate a total transducer failure a multiplication factor of zero would be entered and the value of the measurement written into the task common blocks would be zero. A multiplication factor of -1.0 would be entered to simulate an incorrect sign and a multiplication factor of 1.3 would simulate a transducer reading 30% too high. The multiplication factors of any of the selected measurement points may be updated or removed at any time while the simulator is running. The values of the multiplication factors and the corresponding measurement point are stored in a task common block.

A similar program allows the user to corrupt the value of a switch status measurement. As with the analogue measurements up to 10 switch status measurements may be corrupted. The values of the chosen switch status

measurements are always the opposite of the true switch status. No facility exists at the present time for setting the switch status measurement to a pre-defined state regardless of the true switch status.

Chapter 4
Development of a robust state estimation algorithm

4.1 Substation data validation

The principle steps in the estimation of the states of an electrical power system have been outlined in chapter 2. As has been discussed in chapter 2 it is critical that the gross measurement errors and corrupt switch status measurements are detected before the state estimation program processes the data, otherwise the effects of the errors are spread over a number of the estimates. The robust state estimation algorithm presented in this thesis is able to operate in the presence of both gross measurement errors and corrupt switch status measurements without the need for any pre-processing of the data.

The algorithm was developed from the 'substation data validation' algorithm by Irving and Sterling[66] and this chapter traces the development of the theory from the original algorithm. The theory and computational results of the original algorithm are presented in the following section.

4.1.1 Theory

The substation data validation of Irving and Sterling[66] uses the following simple mathematical model to define the network of busbar inter-connections within a either a single substation or a group of substations. A node is a point where power flows merge, e.g. at a busbar or a junction and a branch is any connection between the nodes. It should be noted that in the paper the term link is used to describe the connections between the nodes but to avoid any confusion between the term used to represent a bus coupler in the test networks the term branch will be used instead. The branch may contain any number of switches, transformers or lines and is assigned a sending and receiving node number. Branches representing generators are assigned a sending node number of zero and those representing loads a receiving node number of zero. In order to validate the power flow measurements and switch status information it is necessary to estimate the power flows in all the branches. This can be achieved by applying a number of constraints to the power flows. These constraints are summarised as follows.

At every node the algebraic sum of the power flows is zero. Kirchoff's first law.

$$\sum_{a \in I_b} P_{e_a} + E_b = 0 \quad \text{for all } b \tag{4.1}$$

where

b = node number.

P_{e_a} = estimates power flow in branch a.

I_b = set of all branches connected to node b.

E_b = error term associated with the nodal sum check equation b.

A number of the branches will have a flow or injection measurement associated with them which constrains the flow in the branch to be equal to the measurement subject to an error term to allow for any inconsistencies in the measurements arising from the telemetry equipment.

$$P_{e_c} + E_d = P_{m_d} \quad \text{for all } d \tag{4.2}$$

where

d = flow measurement number.

P_{e_c} = estimate of flow in branch c.

P_{m_d} = value of flow measurement number d, associated with branch c.

E_d = error term associated with measurement number d.

The third type of constraint can be applied to any branch which has an open switch status measurement, the branch will have zero flow subject to an error term to allow for an incorrect status measurement.

$$P_{e_f} + E_g = 0 \quad \text{for all } g \tag{4.3}$$

where

g = open switch measurement number.

P_{e_f} = estimated flow in branch f which contains the open switch.

E_g = error term associated with open switch measurement number g.

The error term associated with each of the above constraints is composed of two components and is defined by equation 4.4 as follows.

$$E = e^+ - e^- \tag{4.4}$$

where

E = error term.

e^+/e^- = positive/negative components of the error term.

The above set of constraints now form a set of linear equations which can be solved using any of the standard techniques.[32,87,106,126] The positive and negative components of the error term are needed because linear programming techniques generally require the variables to have a value greater than or equal to zero. The linear programming technique has to minimise an objective function based on the weighted components of the error terms as follows

$$\text{Min } \Phi = \sum_{\text{all } b}[w_b^+ e_b^+ + w_b^- e_b^-] + \sum_{\text{all } d}[w_d^+ e_d^+ + w_d^- e_d^-] + \sum_{\text{all } g}[w_g^+ e_g^+ + w_g^- e_g^-] \quad (4.5)$$

where

Φ = objective function.

w^+/w^- = positive/negative weighting factors associated with the error term.

The weighting factors of the error terms enable different constrains to influence the solution of the estimates of the branch power flows to a greater or lesser extent. The larger the value of the weighting factors the larger the contribution that constraint will make to the objective function, Φ if the constraint is violated, hence the greater the chance that the flow estimates will be such that the error term associated with the constraint is zero. The weighting factors of the equations representing the nodal sum check constraint, 4.1 would therefore be set large ($\sim$ 1000) because this contraint must be true for all the nodes and hence the error term must be zero.

If the telemetry equipment uses a two bit code for the switch status measurements, then the two bit code representing the state of unknown status would be assumed to represent a status of open and an equation formed to represent the constraint of zero power flow in that branch. The weighting factors associated with this equation ($\sim$ 0.5) would be set lower than an equation arising from a genuine open switch status measurement ($\sim$ 1.0) so that should the unknown status really be closed the constraint can be violated without incurring a large increase to the objective function.

The two components of the error terms together with the weighting factors can be used to constrain the flow estimates to lie within a pre-defined range. A generator for example may be constrained to always supply active power subject to its maximum rating, thus the branch representing the generator would be constrained to have a flow estimate greater than zero but less than the maximum rating. These upper and lower limits are represented

mathematically by two linear equations in which the weighting factors are different.

The lower limit on a branch flow defined by the inequality

$$P_{e_i} \geqq P_{min_i} \tag{4.6}$$

where

P_{e_i} = estimate of the flow in branch i.

P_{min_i} = minimum flow in branch i, equal to zero for a branch representing a generator.

would be represented by the linear equation

$$P_{e_i} + e^+ - e^- = P_{min_i} \tag{4.7}$$

where the value of the positive weighting factor, w^+ is large ($\sim$ 1000) and the negative weighting factor, w^- is small ($\sim$ 0.001).

If the value of the flow estimate P_{e_i} is greater than P_{min_i} then the negative component of the error term is greater than zero, but the contribution to the objective function, Φ is small. However, if the flow estimate P_{e_i} is less than P_{min_i} then the positive component of the error term is greater than zero and the contribution to the objective function, Φ is large thus making it undesirable for the estimate to take a value less than P_{min_i}.

Similarly the upper limit on a branch flow defined by the inequality

$$P_{e_i} \leqq P_{max_i} \tag{4.8}$$

where

P_{e_i} = estimate of the flow in branch i.

P_{max_i} = maximum flow in branch i, equal to the maximum rating for a branch representing a generator.

would be represented by the linear equation

$$P_{e_i} + e^+ - e^- = P_{max_i} \tag{4.9}$$

where the value of the positive weighting factor, w^+ is small ($\sim$ 0.001) and

the negative weighting factor, w^- is large ($\sim$ 1000).

It should be noted that in the paper by Irving and Sterling the error terms were placed on the left hand side of the equation, reflecting the true representation of the problem in which the measurement is likely to be subject to an error which must therefore be added to the measurement value to obtain the value of the flow estimate for that branch. In the thesis the error terms have been placed on the right hand side of the equation. The reason for this is to facilitate the understanding of the implementation of the linear programming technique used to solve the set of linear equations formed from the constraints applied to the branch flow estimates. The revised Simplex method has been used to solve the set of equations, further details on the revised Simplex method can be found in Appendix 4 and on the implementation of the technique in chapter 5.

4.1.2 Results

The original data validation program was modified slightly to read the measurements and switch status information from a task common block. The measurement values could then be updated with values read from a data file by a second small program. The original program was tested on several small examples and the reader is referred to chapter 2 and reference 66 for further information on these results. It was however necessary to investigate the potential use of the algorithm to validate the measurements for an entire network and the performance of the algorithm on a network with more injection branches where a measurement error is more likely to go undetected.

The 4 substation network described by Sullivan et al.[134] and used as a test network by Iriving and Sterling was increased in size to contain a total of twenty injection branches, the modified network is illustrated in figure 4.1. The number of upper and lower branch flow limits was also increased from 1 to 40, an upper and lower limit being applied to each of the injection branches. Note the addition of a branch flow limit has similar computational requirements as the addition of another branch flow measurement, thus the solution times for this network can be related to those of a larger network with a further 40 measurements but no branch flow limits. Initial testing of the data validation program on the modified network highlighted the inefficiency of the full matrix method used in its implementation. The computational requirements to obtain a solution from a flat start in which all the branch flow estimates were initialised to zero rose from 9 seconds quoted

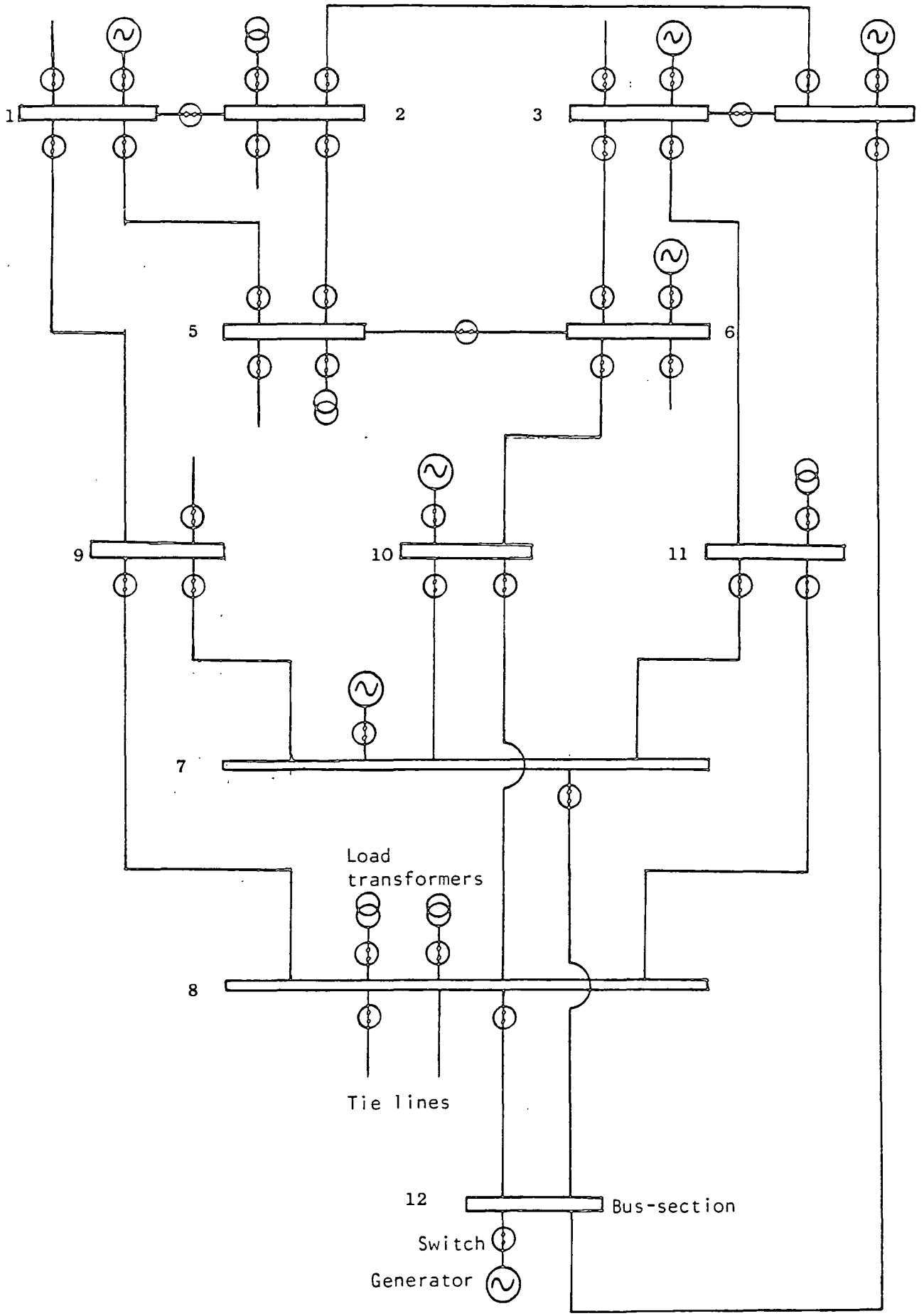


Fig. 4.1: Four substation network

in reference 66 for the original 4 substation network to 150 seconds for the modified network.

The algorithm continued to show the good ability to detect and correct gross measurement errors and corrupt switch status measurements, however the solution time was unacceptably slow.

The original implementation of the method used a full matrix linear programming routine which explains the large increase in the solution time with only a moderate increase in the problem size. The program was therefore re-written to make use of a sparse linear programming routine. The program was also adapted to run in conjunction with the simulator program and the 4 test networks described in chapter 3 which enabled the more realistic simulation of an on-line environment.

The details and results of this program have been presented in the next section.

4.1.3 The on-line sparse data validation program

The extreme sparse nature of the coefficint matrix used to define the branch flow estimates in terms of the flow measurement values and the nodal sum check equations is illustrated in figure 4.2. The grid represents the elements a_{ij} of the matrix A in the following matrix equation

$$AP_e = B \tag{4.10}$$

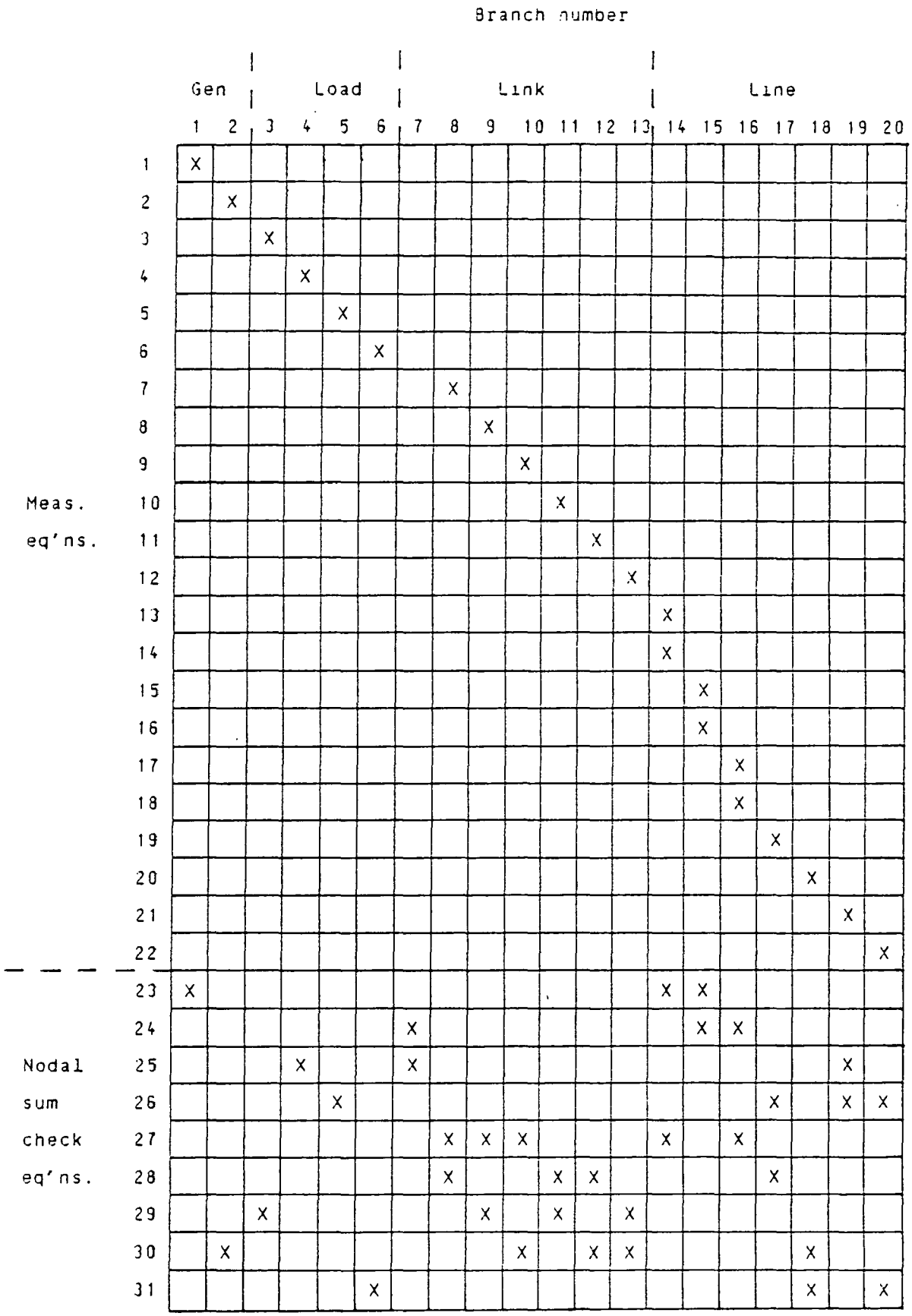
where

B = vector of branch flow measurements and values of the nodal sum check equations.

P_e = vector of branch flow estimates.

A = coefficient matrix

The vector of branch flow estimates, P_e has been partioned as follows. Injection branches representing the generators followed by injection branches representing the loads, branches representing the links (bus couplers) and finally branches representing the lines. The matrix illustrated in figure 4.2 is for the 5 substation test network, hence there are a total of 20 branches, 2 to represent the generators, 4 the loads and 7 each for the links and lines. The vector representing the right hand side of the equation, B has 22 measurement values followed by 9 values of zero arising from the nodal sum check equations at the 9 bus-sections.



X = Non zero elements of the matrix

Fig. 4.2: Illustration of the sparse nature of the data validation coefficient matrix.

During the development of the on-line sparse program the data for the modified 4 substation network was processed to enable a comparison of the solution times to be made. The solution time for a flat start was reduced from 150 seconds to 6 seconds, and the sparse routine also allowed the use of a previous solution as a starting point for the current solution which results in a further reduction of the solution time. No further testing was performed on the modified 4 substation network and development continued with the adaptation to the four test networks detailed in chapter 3.

The program was structured to allow user definition of the substations at which the validation of the active power flow measurements was to take place. The maximum number of substations to which the program could be applied was only limited by the dimensions of the arrays in the program. This arrangement allowed the program to be assigned to a wide variety of different substations selected from the test networks. It should be noted that the substations selected need not be inter-connected, for example several small groups of substations could be validated simultaneously, furthermore the program was suitably dimensioned to allow it to be applied to the entire five substation test network.

During the initial development and testing of the program it became apparent that for the results to be reliable in the majority of cases the single branch representation of a line was not adequate since this representation makes no allowance for the active power flow losses in a line, but uses a single branch flow estimate to represent the active power flows at both ends of the line. Thus if the technique is presented with a completely valid set of measurements, then some are bound to have an associated error. This is illustrated in table 4.1 where the estimates of all the branch flows are compared with those produced by the simulator for the 5 substation test network in its initial steady state. Line numbers 1, 2 and 3 all have measurements at both ends of the line and the branch flow estimate has been set to one or other of these values. This results in an error in the branch flow estimates for all but one of the injection branches (generators and loads).

The logical progression from using one branch to represent the line flow estimate is to use two, one to represent the flow at the sending end of the line, the other the flow at the receiving end of the line. The expansion of the model to include the transmission line losses is discussed in the following section.

Table 4.1: Estimates of the active power flows from the first data validation program on the 5 substation test network.

Values are in P.U.
(S) => sending end of a line, (R) => receiving end of line.
Error = Estimate - True value.

Device No.	True value	Estimate	Error
Gen 1	1.2956	1.2696	-0.0260
Gen 2	0.4000	0.3958	-0.0042
Load 1	-0.2000	-0.2000	0.0000
Load 2	-0.4500	-0.4531	-0.0031
Load 3	-0.4000	-0.4005	-0.0005
Load 4	-0.6000	-0.6117	-0.0117
Link 1	0.6388	0.6421	0.0033
Link 2	0.2267	0.2267	0.0000
Link 3	0.2069	0.2069	0.0000
Link 4	0.1939	0.1939	0.0000
Link 5	-0.0198	-0.0198	0.0000
Link 6	-0.0328	-0.0285	0.0043
Link 7	-0.0129	-0.0129	0.0000
Line 1(S)	-0.8884	-0.8743	0.0141
Line 1(R)	0.8743	0.8743	0.0000
Line 2(S)	-0.4071	-0.3952	0.0119
Line 2(R)	0.3952	0.3952	0.0000
Line 3(S)	0.2434	0.2468	0.0034
Line 3(R)	-0.2469	-0.2468	0.0001
Line 4(S)	0.2749	0.2749	0.0000
Line 4(R)	-0.2794	-0.2749	0.0045
Line 5(S)	-0.5483	-0.5483	0.0000
Line 5(R)	0.5371	0.5483	0.0112
Line 6(S)	-0.1890	-0.1890	0.0000
Line 6(R)	0.1886	0.1890	0.0004
Line 7(S)	-0.0634	-0.0634	0.0000
Line 7(R)	0.0631	0.0634	0.0003

4.1.4 Transmission line losses

The representation of a transmission line by two branches requires the addition of a sub-set of linear equations to the existing set which define the flow in one of the branches representing the line in terms of the flow in the other branch. A physical interpretation of this idea is that the two branches representing the line would join to form a node in the centre of the line. A third shunt branch between the centre node and the ground would then carry a flow equivalent to the line loss. Mathematically the addition of the centre node and third branch is not required as an equation can be formed which equates the difference in the power flows at the ends of a line to a value representing the line loss, subject of course to an error term.

Remembering that the direction of flow in a line is positive if the flow is into the bus-section to which that end of the line is connected then the equation is of the form

$$P_{e_s} + P_{e_r} + E_l = L \qquad (4.11)$$

where

P_{e_s} = branch flow estimate for the sending end of the line.
P_{e_r} = branch flow estimate for the receiving end of the line.
E_l = error term.
L = active power line loss (a loss is regarded as a negative quantity).

The selection of a suitable value for the line loss has proved to be a difficult task. Ideally a constraint is needed which limits the value to lie with in a defined range, for example the line loss could be constrained to be less than or equal to zero but greater than the maximum likely loss when the line is carrying its maximum current. Constraints of this type can be defined mathematically by the inequality equations 4.6 and 4.8 and applied to the branch flow estimates in the form of the linear equations 4.7 and 4.9 as described in section 4.1.1. However in terms of the solution of the set of linear equations these inequality constraints are treated as alternative definitions of the branch flow estimates. Thus in the absence of additional information an inequality equation may be used to define the flow estimate of one of the branches representing line i in terms of the flow estimate of the other branch, using the value entered for the line loss, l_i. Thus the two branch flow estimates would be the same if the constraint equation representing a line loss of zero was selected or alternatively the difference between the branch flow estimates would be equal to the value entered for the

maximum loss if this contraint equation was selected. It should be noted however that it may be possible to define the estimates of both branch flows without using either of the constraint equations, for instance there may be sufficient flow measurements which can be combined with the bus-section sum check equations to define the flow in every branch. The form of linear programming problem adopted is such that a single value of the line loss, l_i must be specified for each line, it is not possible therefore to specify a range for the line loss. It should be noted that a range could be specified by using two equations of the form of equation 4.11 but this increases the size of the problem with a corresponding increase in the solution time.

The vector representing the branch flow estimates has been partitioned in a similar fashion to the one in the original program, that is branches reprenting the generators followed by those representing the loads, then the links and finally a pair of branches for each line. Thus for the 5 substation test network there is now a total of 27 branches. The set of linear equations used to define the estimates of the branch flows were the same as before except an additional equation of the form of equation 4.11 was appended to the set for each of the transmission lines.

The initial values of the line flow losses, L were set to zero and a suitable value of the weighting factors found by experimental trials. If the value was set too high then the branch flow estimates were forced to be the same and the program produced the same results as before. Alternatively if the value was too low the difference in the branch flow estimates could be set to any value to satisfy the local flow measurements whether they be valid or not.

The program gave much better results when the value of the line flow loss, l_i in equation 4.11 was set to the value calculated from the estimates of the power flows obtained from the previous solution and the weighting factors set to a value similar to those for the measurement equations ($\tilde{}$ 1.0). In the case where no suitable previous estimates were available for example on the first estimation run or when the network has undergone a sudden change in the operating conditions caused, for example by a switch status changing, then the estimates produced could be significantly improved if the program was allowed to calculate the estimates twice. Again the value for the line loss would be calculated from the most recent estimates of the line flows, which in the case of the first estimation were initialised to zero.

The results of the branch flow estimates calculated from a flat start for the 5 substation test network in its steady state are presented in table 4.2. The magnitude of the errors in the estimates are generally less than those

Table 4.2: Estimates of the active power flows from the second data validation program on the 5 substation test network.

Values are in P.U.
(S) => sending end of a line, (R) => receiving end of line.
Error = Estimate - True value.

Device No.	True value	Estimate	Error
Gen 1	1.2956	1.2956	0.0000
Gen 2	0.4000	0.4000	0.0000
Load 1	-0.2000	-0.2000	0.0000
Load 2	-0.4500	-0.4500	0.0000
Load 3	-0.4000	-0.4000	0.0000
Load 4	-0.6000	-0.6000	0.0000
Link 1	0.6388	0.6386	-0.0002
Link 2	0.2267	0.2266	-0.0001
Link 3	0.2069	0.2069	0.0000
Link 4	0.1939	0.1939	0.0000
Link 5	-0.0198	-0.0198	0.0000
Link 6	-0.0328	-0.0327	0.0001
Link 7	-0.0129	-0.0129	0.0000
Line 1(S)	-0.8884	-0.8884	0.0000
Line 1(R)	0.8743	0.8743	0.0000
Line 2(S)	-0.4071	-0.4071	0.0000
Line 2(R)	0.3952	0.3952	0.0000
Line 3(S)	0.2434	0.2434	0.0000
Line 3(R)	-0.2469	-0.2469	0.0000
Line 4(S)	0.2749	0.2749	0.0000
Line 4(R)	-0.2794	-0.2791	0.0003
Line 5(S)	-0.5483	-0.5483	0.0000
Line 5(R)	0.5371	0.5483	0.0112
Line 6(S)	-0.1890	-0.1886	0.0004
Line 6(R)	0.1886	0.1884	-0.0002
Line 7(S)	-0.0634	-0.0634	0.0000
Line 7(R)	0.0631	0.0517	0.0114

from the first version of the program which are listed in table 4.1. However, the program still fails to produce reliable estimates for the receiving ends of lines 5 and 7 which are connected to bus-section 9 (substation 5) where the measurement redundancy is low. This problem is highlighted when the measurement at the sending end of line 7 is set to zero. The estimates formed from these measurements are listed in table 4.3 and it can be seen that the program is unable to detect the measurement error and sets the flow in both ends of line 7 to zero. The nodal sum check at bus-section 9 is then satisfied by the flow estimate for the receiving end of line 5 being set equal to the load at bus-section 9.

The majority of state estimation techniques presently in use are based on the least squares method,[41] these state estimators are not very reliable in the presence of switch status errors and bad measurement data[69] and some for of filtering is necessary to remove the erroneous information. Thus the date validation technique if it is to be of use as a measurement filter must be able to detect and pinpoint the bad measurements. Ideally it should also be able to replace a bad measurement by an estimate of the correct value, allowing the state estimator to resolve any of the small remaining discrepancies.

The same principle of validating active power flows can also be applied to the validation of reactive power flows and as it is likely that both the active and reactive power flows would be measured simultaneously then it is reasonable to expect a data validation program to be capable of validating both sets of measurements. However the range of line flow losses for reactive powerflow is considerably larger than that for the active power flows, the values may also be positive or negative. A heuristic method of determining the value for the line flow losses is thus even more likely to produce poor estimates leading to the failure of the program to detect bad measurements. The need to calculate accurate values for the line flow losses has resulted in the development of an algorithm for the estimation of all the states of a power system including the voltage magnitude and phase angle at all the nodes together with the active and reactive power flows throughout the network.

4.2 Robust state estimation

The active and reactive power flow losses in a line can be easily calculated from the line parameters together with the voltage magnitude and phase angle at the sending and receiving ends of the line. The equations for the calculation of the losses can be derived from the standard equation 3.11

Table 4.3: Estimates of the active power flows from the second data validation program on the 5 substation test network with the measurement on line 7 set to zero.

Values are in P.U.
(S) => sending end of a line, (R) => receiving end of line.
Error = Estimate - True value.

Device No.	True value	Estimate	Error
Gen 1	1.2956	1.2956	0.0000
Gen 2	0.4000	0.4000	0.0000
Load 1	-0.2000	-0.2000	0.0000
Load 2	-0.4500	-0.4500	0.0000
Load 3	-0.4000	-0.4000	0.0000
Load 4	-0.6000	-0.6000	0.0000
Link 1	0.6388	0.6386	-0.0002
Link 2	0.2267	0.2266	-0.0001
Link 3	0.2069	0.2069	0.0000
Link 4	0.1939	0.1939	0.0000
Link 5	-0.0198	-0.0198	0.0000
Link 6	-0.0328	-0.0327	0.0001
Link 7	-0.0129	-0.0129	0.0000
Line 1(S)	-0.8884	-0.8884	0.0000
Line 1(R)	0.8743	0.8743	0.0000
Line 2(S)	-0.4071	-0.4071	0.0000
Line 2(R)	0.3952	0.3952	0.0000
Line 3(S)	0.2434	0.2434	0.0000
Line 3(R)	-0.2469	-0.2469	0.0000
Line 4(S)	0.2749	0.2749	0.0000
Line 4(R)	-0.2794	-0.2791	0.0003
Line 5(S)	-0.5483	-0.5483	0.0000
Line 5(R)	0.5371	0.6000	0.0629
Line 6(S)	-0.1890	-0.1886	0.0004
Line 6(R)	0.1886	0.1251	-0.0635
Line 7(S)	-0.0634	0.0000	0.0634
Line 7(R)	0.0631	0.0000	-0.0631

and 3.12 in section 3.3.2 which define the active and reactive power flows at one end of a line in terms of the voltage levels at both ends of the line. The values of the voltage magnitude are unlikely to be measured at both ends of all the lines in a network and in the foreseeable future it is unlikely that the voltage phase angles will not be measured anywhere in the network. Thus the calculation of the power flow losses in the lines is not possible from measurements of the voltage levels and the data validation program must therefore calculate the power flow losses by calculating the voltage magnitude and phase angle at each end of the line, thereby turning the algorithm into a state estimation algorithm. The following sections of this chapter develop the theory of the method used in the state estimation process and chapter 5 details the implementation of the algorithm together with the linear programming technique used to obtain the solution of the estimates. A presentation and discussion of the results can be found in chapter 6.

4.2.1 Calculation of line losses and voltage drops

The mathematical model used to describe the network follows the conventions already mentioned in chapter 3 of the thesis but summarised here for convenience.

1. The power flow, S is defined as P - j Q
2. A power flow is said to be positive if the direction of flow is into the bus-section to which the device is connected.
3. A line power flow loss is regarded as a negative quantity.

It should be noted that in the four test networks used to evaluate the performance of the programs, each bus-section may be able to form a single electrical node depending on the status on the surrounding line and link switches. Strictly speaking two or more bus-section connected together by an active link form one electrical node since the links have been assumed to have zero resistance and the voltage levels of the bus-sections will then be identical. To avoid confusion in the following section the term bus-section will replace the term node used previously to define the point where two or more branches join because the state estimation algorithm allows for the possibility of any bus-section forming a true node with its own unique voltage levels. The state estimation algorithm has been developed using the polar notation to represent complex numbers.

The standard equations defining the power flows at the end of a line in terms

of the voltage levels at both ends of the line are as follows

$$P_s = V_s V_r Y_{ln} \cos(\theta_{sr}) - (V_s)^2 (Y_{ln} \cos(\sigma) + Y_{lg} \cos(\alpha)) \quad (4.12)$$

$$Q_s = V_s V_r Y_{ln} \sin(\theta_{sr}) + (V_s)^2 (Y_{ln} \sin(\sigma) + Y_{lg} \sin(\alpha)) \quad (4.13)$$

where

The subscripts r and s refer to the receiving and sending ends of the line respectively.

P = active power flow.

Q = reactive power flow.

V = voltage magnitude.

$\theta_{sr} = \theta_s - \theta_r - \sigma$.

θ = voltage phase angle.

Y_{ln} = series line admittance magnitude.

σ = series line admittance angle.

Y_{lg} = shunt admittance magnitude between the line and the ground.

α = shunt admittance angle between the line and the ground.

Inter-changing the terms for the sending and receiving ends of the line, the active and reactive power flows for the receiving end of the line are given by the equations

$$P_r = V_s V_r Y_{ln} \cos(\theta_{rs}) - (V_r)^2 (Y_{ln} \cos(\sigma) + Y_{lg} \cos(\alpha)) \quad (4.14)$$

$$Q_r = V_s V_r Y_{ln} \sin(\theta_{rs}) + (V_r)^2 (Y_{ln} \sin(\sigma) + Y_{lg} \sin(\alpha)) \quad (4.15)$$

where

$\theta_{rs} = \theta_r - \theta_s - \sigma$.

Adding equations 4.12 and 4.14 and defining C_f as

$$C_f = Y_{ln} \cos(\sigma) + Y_{lg} \cos(\alpha) \quad (4.16)$$

gives the active power flow loss in a line as

$$P_s + P_r = V_s V_r Y_{ln} [\cos(\theta_{sr}) + \cos(\theta_{rs})] - C_f [(V_s)^2 + (V_r)^2] \quad (4.17)$$

similarly adding equations 4.13 and 4.15 and defining S_f as

$$S_f = Y_{ln}\sin(\sigma) + Y_{lg}\sin(\alpha) \tag{4.18}$$

gives the reactive power flow line loss as

$$Q_s + Q_r = V_s V_r Y_{ln}[\sin(\theta_{sr}) + \sin(\theta_{rs})] - S_f[(V_s)^2 + (V_r)^2] \tag{4.19}$$

The four equations defining the power flows at both ends of the line 4.12 to 4.15 may be re-arranged to produce an equation which evaluates the difference between the voltage magnitude at the ends of the line. The equation is of the form

$$V_s - V_r = (f_1)^{0.5} + (f_2)^{0.5} \tag{4.20}$$

where

f_1 = function of the power flows at the sending end of the line and the voltage phase angle difference across the line.

f_2 = function of the power flows at the receiving end of the line and the voltage phase angle difference across the line.

The usage of the above equation was unlikely to result in a stable state estimation method because if the state estimates were not close to convergence the values of the two functions, f_1 and f_2 which are used as arguments for the square root function were likely to be negative. The most obvious way to avoid the problem of a negative argument to the square root function is to work in terms of the difference between the squares of the voltage magnitudes at the ends of the line.

Hence subtracting equation 4.14 from 4.12 and 4.15 from 4.13 gives

$$P_s - P_r = V_s V_r Y_{ln}[\cos(\theta_{sr}) - \cos(\theta_{rs})] - C_f[(V_s)^2 - (V_r)^2] \tag{4.21}$$

$$Q_s - Q_r = V_s V_r Y_{ln}[\sin(\theta_{sr}) - \sin(\theta_{rs})] + S_f[(V_s)^2 - (V_r)^2] \tag{4.22}$$

Re-arranging equation 4.22 and substituting $V_s V_r Y_{ln}$ into equation 4.21 gives

$$P_s - P_r = -C_f[(V_s)^2 - (V_r)^2] + \tag{4.23}$$

$$\frac{[Q_s - Q_r - S_f[(V_s)^2 - (V_r)^2]]\,[\cos(\theta_{sr}) - \cos(\theta_{rs})]}{\sin(\theta_{sr}) - \sin(\theta_{rs})}$$

which may be re-arranged to give

$$(P_s - P_r)\,[\sin(\theta_{sr}) - \sin(\theta_{rs})] = (Q_s - Q_r)\,[\cos(\theta_{sr}) - \cos(\theta_{rs})] \quad (4.24)$$

$$- S_f[\cos(\theta_{sr}) - \cos(\theta_{rs})]\,[(V_s)^2 - (V_r)^2]$$

$$- C_f[\sin(\theta_{sr}) - \sin(\theta_{rs})]\,[(V_s)^2 - (V_r)^2]$$

hence collecting the terms in the square of the voltage magnitude gives the difference between the squares of the voltage magnitudes across the line as

$$(V_s)^2 - (V_r)^2 = \quad (4.25)$$

$$\frac{(Q_s - Q_r)\,[\cos(\theta_{sr}) - \cos(\theta_{rs})] - (P_s - P_r)\,[\sin(\theta_{sr}) - \sin(\theta_{rs})]}{C_f[\sin(\theta_{sr}) - \sin(\theta_{rs})] + S_f[\cos(\theta_{sr}) - \cos(\theta_{rs})]}$$

An equation defining the difference between the voltage phase angles across a line can be derived as follows. Re-arranging equations 4.12 and 4.13 and dividing the first into the second gives

$$\frac{V_s V_r Y_{ln} \sin(\theta_{sr}) = Q_s - (V_s)^2 S_f}{V_s V_r Y_{ln} \cos(\theta_{sr}) \quad P_s - (V_s)^2 C_f} \quad (4.26)$$

which may be re-arranged to give

$$\tan(\theta_{sr}) = \tan(\theta_s - \theta_r - \sigma) = \frac{Q_s - (V_s)^2 S_f}{P_s - (V_s)^2 C_f} \quad (4.27)$$

similarly equations 4.14 and 4.15 give

$$\tan(\theta_{rs}) = \tan(\theta_r - \theta_s - \sigma) = \frac{Q_r - (V_r)^2 S_f}{P_r - (V_r)^2 C_f} \quad (4.28)$$

Taking the arctangent of equations 4.27 and 4.28 and subtracting gives

$$2(\theta_s - \theta_r) = \arctan\left[\frac{Q_s - (V_s)^2 S_f}{P_s - (V_s)^2 C_f}\right] - \arctan\left[\frac{Q_r - (V_r)^2 S_f}{P_r - (V_r)^2 C_f}\right] \quad (4.29)$$

Thus the difference between the voltage phase angles across a line is

$$\frac{\arctan\left[\frac{Q_s - (V_s)^2 S_f}{P_s - (V_s)^2 C_f}\right] - \arctan\left[\frac{Q_r - (V_r)^2 S_f}{P_r - (V_r)^2 C_f}\right]}{2} \qquad (4.30)$$

In most state estimation algorithms the most recent values of the estimates are used to calculate mis-match values and the new mis-match values are then used to re-calculate the estimates. The iterative process is terminated when the change in the value of the mis-match values or the estimates falls below a pre-defined tolerance. In the proposed algorithm for etate estimation the four equations defining the power flow losses in and the voltage differences across a line 4.17, 4.19, 4.25 and 4.30 are re-evaluated during every iteration using the most recent values of the power flow and voltage level estimates for the sending and receiving ends of the lines. No mis-match values as such are calculated, the iterative process being terminated when the change in the values of all the power flow and voltage estimates falls below a pre-defined tolerance.

The state estimation algorithm is divided into four distinct sub-estimation stages, each of which is solved once during every iteration. The four sub-estimation stages calculate estimates for the following network variables respectively. The active power flows in every device throughout the network, the reactive power flows throughout the network, the voltage magnitude at every bus-section in the network and the voltage phase angle at every bus-section. Each sub-estimation stage involves the solution of an over-determined set of linear equations formed in a similar way to those used in the data validation algorithm. The formation of the linear equations is described in the following section.

4.2.2 Formulation of the state estimation problem

In a similar manner to the problem definition of the substation data validation technique the problem definition of the active power sub-estimation stage can be formed using a set of linear equations derived from the bus-section sum check balance equations, the active power flow measurements, any open switch status measurements and the equations equating the sum of the line flows at each end of a line to the active power flow loss calculated using equation 4.17. The set of linear equations can be written as follows.

For each bus-section in the network the algebraic sum of the active power flows is zero, thus

$$\sum_{a \varepsilon I_b} P_{e_a} + E_b = 0 \quad \text{for all b} \tag{4.31}$$

where

b = bus-section number.

P_{e_a} = estimates power flow in branch a.

I_b = set of all branches connected to bus-section b.

E_b = error term associated with the nodal sum check equation b.

The active power flow estimate plus an error term is equal to the measured value, thus

$$P_{e_c} + E_d = P_{m_d} \quad \text{for all d} \tag{4.32}$$

where

d = flow measurement number.

P_{e_c} = estimate of flow in branch c.

P_{m_d} = value of flow measurement number d, associated with branch c.

E_d = error term associated with measurement number d.

The active power flow estimate plus an error term is equal to zero if a switch is open, thus

$$P_{e_f} + E_g = 0 \quad \text{for all g} \tag{4.33}$$

where

g = open switch measurement number.

P_{e_f} = estimated flow in branch f which contains the open switch.

E_g = error term associated with open switch measurement number g.

The sum of the active power flow estimates at the ends of a line plus an error term is equal to the calculated active power line loss, thus

$$P_{e_{sl}} + P_{e_{rl}} + E_l = L_l \quad \text{for all l} \tag{4.34}$$

where

l = line number.

$P_{e_{sl}}$ = branch flow estimated for the sending end of the line.

$P_{e_{rl}}$ = branch flow estimated for the recieving end of the line.

E_l = error term for the line loss equation l.

L_l = active power line loss, calculated from equation 4.1.

The objective function for the active power flow estimation stage is thus

$$\text{Min } \Phi_p = \sum_{\text{all } b}[w_b^+ e_b^+ + w_b^- e_b^-] + \sum_{\text{all } d}[w_d^+ e_d^+ + w_d^- e_d^-] \quad (4.35)$$

$$+ \sum_{\text{all } g}[w_g^+ e_g^+ + w_g^- e_g^-] + \sum_{\text{all } l}[w_l^+ e_l^+ + w_l^- e_l^-]$$

where

Φ_p = objective function for the active power flow estimation stage, the minimisation of which can be found by the application of linear programming.

The set of linear equations for the reactive power flow estimation stage are analogous to those for the active power flow except the reactive power flow measurements are used in equation 4.32 and equation 4.19 is used to calculate the reactive power flow line loss for equation 4.34. The application of a linear programming routine can also be used to minimise an objective function, Φ_q defined in a similar way to the objective function for the active power flow estimation stage.

The equation for the calculation of the voltage magnitude drop across a line, equation 4.25, is in terms of the difference between the square of the voltage magnitudes at the ends of the line. This necessitates the sub-estimation stage for the voltage magnitude estimates being implemented in terms of the square of the voltage magnitudes. The voltage magnitude measurements are thus squared prior to the sub-estimation problem and the square root of the voltage magnitude estimates calculated after each voltage magnitude sub-estimation has been obtained.

The set of linear equationns for the estimation of the square of the voltage magnitudes at every bus-section throughout the network are derived by considering the following constraints on the voltage levels.
The square of the voltage magnitude estimate is equal to the square of the voltage magnitude measurement, subject to an error term, thus

$$V_{2_a} + E_b = (V_{m_b})^2 \quad \text{for all } b \quad (4.36)$$

where

b = voltage magnitude measurement number.

V_{2_a} = estimate of the square of the voltage magnitude at bus-section a.

V_{m_b} = voltage magnitude measurement number b, associated with bus-section a.

E_b = error term associated with measurement number b.

The voltage magnitudes of any two bus-sections connected together by an active link are identical, thus the difference between the squares of the voltage magnitudes is zero subject to an error term to allow for the possibility that the link switch status is incorrect, thus

$$V_{2_{sc}} - V_{2_{rc}} + E_c = 0.0 \quad \text{for all c} \tag{4.37}$$

where

c = active link number.

$V_{2_{sc}}$ = estimate of the square of the voltage magnitude at the sending bus-section of link c.

$V_{2_{rc}}$ = estimate of the square of the voltage magnitude at the receiving bus-section of link c.

E_c = error term.

The difference between the squares of the voltage magnitudes at the sending and receiving ends of an active line is given by equation 4.25, including an error term to allow for the incorrect line status or an error in the calculated value of the difference leads to the formation of the following equation.

$$V_{2_{sd}} - V_{2_{rd}} + E_d = D_d \quad \text{for all d} \tag{4.38}$$

where

d = line number.

$V_{2_{sd}}$ = estimate of the square of the voltage magnitude at the sending bus-section of line d.

$V_{2_{rd}}$ = estimate of the square of the voltage magnitude at the receiving bus-section of line d.

D_d = difference between the squares of the voltage magnitudes calculated from equation 4.25.

E_d = error term.

The objective function for the voltage magnitude estimation stage is thus

$$\text{Min } \Phi_v = \sum_{\text{all b}} [w_b^+ e_b^+ + w_b^- e_b^-] + \sum_{\text{all c}} [w_c^+ e_c^+ + w_c^- e_c^-] + \sum_{\text{all d}} [w_d^+ e_d^+ + w_d^- e_d^-] \quad (4.39)$$

where

Φ_v = objective function for the voltage magnitude estimation stage, the minimisation of which is obtained by the use of linear programming.

After each solution of the estimation of the square of the voltage magnitudes the square root is taken to obtain the voltage magnitude estimate at each bus-section. Thus

$$V_e = \sqrt{V2_e} \quad \text{for all e} \quad (4.40)$$

where

e = bus-section number.

V_e = estimate of the voltage magnitude for bus-section e.

The formation of the linear equations for the estimation of the voltage phgase angles throughout the network is analogous to the formation of the voltage magnitude equations. However the voltage phase angles throughout a power system are obtained with reference to a fixed angle at a selected point in the network and the true voltage phase angle is estimated as opposed to the square of the value. The value of the fixed angle is usually taken to be zero and hence an equation can be formed equating the estimate of the voltage phase angle at the chosen reference bus-section to zero. The linear equation formed to apply this constraint has an error term which is forced to be zero by assigning large weighting factors to it, thus

$$\theta_{e_a} + E_a = 0.0 \quad (4.41)$$

where

θ_{e_a} = voltage phase angle estimate at the reference bus-section.

E_a = error term, forced to be zero by large weighting factors.

If any phase angle measurements were available then a set of linear equations equating the estimate of the phase angle at the relevant bus-section

to the measurement value would be formed, each equation having an error term to allow for measurement inaccuracies, thus

$$\theta_{e_b} + E_b = \theta_{m_c} \quad \text{for all c} \tag{4.42}$$

where

c = voltage phase angle measurement number.

θ_{e_b} = estimate of the voltage phase angle at bus-section b.

θ_{m_c} = voltage phase angle measurement number c, associated with bus-section number b.

E_c = error term.

The voltage phase angles of any two bus-sections connected together by an active link are identical, thus the difference between the angles is zero subject to an error term to allow for the possibility of the link switch status being incorrect, thus

$$\theta_{e_{sc}} - \theta_{e_{rc}} + E_d = 0.0 \quad \text{for all d} \tag{4.43}$$

where

d = active link number.

$\theta_{e_{sd}}$ = estimate of the voltage phase angle at the sending bus-section of link d.

$\theta_{e_{rd}}$ = estimate of the voltage phase angle at the receiving bus-section of link d.

E_d = error term.

The difference between the voltage phase angles at the sending and receiving ends of an active line is given by equation 4.30, hence a linear equation can be written equating the two voltage levels to the difference, subject to an error term to allow for the incorrect line status and an error in the calculated value of the difference, thus

$$\theta_{e_{se}} - \theta_{e_{re}} + E_e = D_e \quad \text{for all e} \tag{4.44}$$

where

e = line number.

$\theta_{e_{se}}$ = estimate of the voltage phase angle at the sending bus-section of line e.

$\theta_{e_{re}}$ = estimate of the voltage phase angle at the receiving bus-section of line e.

D_e = difference between the voltage phase angles calculated from equation 4.30.

The objective function for the voltage magnitude estimation stage is thus

$$\text{Min } \Phi_\theta = w_a^+ e_a^+ + w_a^- e_a^- + \sum_{\text{all } c} [w_c^+ e_c^+ + w_c^- e_c^-] \qquad (4.45)$$

$$+ \sum_{\text{all } d} [w_d^+ e_d^+ + w_d^- e_d^-] + \sum_{\text{all } e} [w_e^+ e_e^+ + w_e^- e_e^-]$$

where

Φ_θ = objective function for the voltage phase angle estimation the minimisation of which is obtained using linear programming.

The four test networks have no voltage phase angle measurements, hence no linear equations of the form 4.42 are included in the set. The four sets of linear equations formed for each of the sub-estimation problems can be solved separately using the revised Simplex method[87] or indeed by any standard linear programming technique.[32,106,126]

The implementation of the 4 stage state estimation algorithm is discussed in the following chapter.

Chapter 5
Implementation of the State Estimation Algorithm

5.1 Implementation of the techniques for solving a set of linear equations

A brief description of the implementation of both the Revised Simplex method and the least squares method is presented in this section. The two algorithms have been written in the form of subroutines which accept the data defining the set of linear equations in a sparse form. The sparse nature of a typical coefficient matrix used in power system state estimation has been previously illustrated in figure 4.2. Considerable computational gains can be made in both CPU times and storage requirements if only the non-zero elements of the coefficient matrix are processed. For example, the elements of the matrix A in the equation

$$AX = B \tag{5.1}$$

where

X = solution vector of length n.

B = input vector of length m.

A = m*n coefficient matrix.

are entered in three vectors, the first of which defines the value of the coefficient, the second and third vectors define the row and column number respectively of the position of the coefficient in the matrix. A summary of the mathematical theory of both the Revised Simplex method and the least squares method can be found in appendix 4.

5.1.1 The Revised Simplex method

The subroutine called SLPEST (Sparse Linear Programming Estimation) has been written to solve a set of over-determined linear equations as defined by the user on entry, subject to the minimisation of an objective function computed from the sum of the modulii of the weighted error terms. The reader is referred to appendix 4 for details on the mathematical procedure of the Revised Simplex process.

The implementation of the Revised Simplex method uses the Harwell library suite of subroutines LA05A, LA05B, LA05C, LA05E and MC20A to manage the formation and updating of the Simplex tableau together with the evaluation of

the solution vector, X. The function of these subroutines is briefly outlined below.

The subroutine LA05A is used to factorise a m*m non-singular coefficient matrix into a lower and upper triangular matrix of the form

$$A' = LU \tag{5.2}$$

where

A' = m*m coefficient matrix.

L = m*m lower triangular matrix.

U = m*m upper triangular matrix.

The matrices L and U are stored in a manner which maximises the use of the sparse nature of the coefficients. The subroutine also initialises some indexing arrays which enable the subroutine LA05C to update the matrices L and U to reflect the exchange of a basic variable for a non-basic variable. This feature is exploited in the main section of the Simplex method and reduces the number of elements in the Simplex tableau that need updating after each basis exchange from m^2 to the few elements of the appropriate rows and columns of the matrices L and U.

The Harwell subroutine LA05B uses the matrices L and U to solve the set of linearly independent equations defined as below

$$A'X' = LUX' = B \tag{5.3}$$

where

X' = solution vector of length m.

B = input vector of length m.

Forward substitution on the matrix L is used to evaluate an intermediate solution vector X'' as defined by equation 5.4

$$LX'' = B \tag{5.4}$$

where

X'' = intermediate solution vector of length m.

The solution to equation 5.3 is comleted by backward substitution on the matrix U to evaluate the solution vector X' as defined by equation 5.5

$$UX' = X'' \tag{5.5}$$

Thus the subroutine LA05B is used to evaluate the new solution vector after the subroutine LA05C has updated the matrices L and U to reflect the exchange of a basic variable for a non-basic variable. Subroutine LA05B is also able to define the non-basic column vector representing the non-basic variable chosen to enter the basis in terms of the column vectors already in the basis. This function of the subroutine LA05B is used during the Simplex process to determine which basis column representing a basic variable should leave the basis once a suitable variable has been found to enter the basis.

The remaining Harwell subroutines, LA05E and MC20A are support routines to the above subroutines and are used in the re-ordering and storing of the elements of the factorised matrices L and U. All the above subroutines exploit the sparse nature of the coefficient matrix to the full.

The linear programming problem as solved by the subroutine SLPEST is represented by the following matrix equation

$$AX + IE^{+} - IE^{-} = B \tag{5.6}$$

where

A = m*n coefficient matrix, m>n.

X = solution vector of length n.

B = input vector of length m.

I = m*m identity matrix.

E^{+}/E^{-} = vectors of length m of variables representing the positive/negative error terms.

The method minimises the following objective function.

$$\text{Min } \Phi = \sum_{i=1}^{m} (w^{+}_i e^{+}_i + w^{-}_i e^{-}_i) \tag{5.7}$$

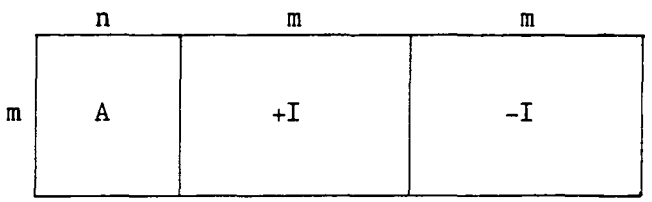
where

W^{+}/W^{-} = vectors of length m of weighting factors for the error terms.

m = number of equations.

The user supplies the subroutine with the coefficients of the matrix A in a sparse form as described at the beginning of section 5.1 together with the vectors W^{+}, W^{-} and B.

The matrix equation 5.6 is equivalent to a tableau of the form illustrated below



The elements of the solution vector, X are always constrained to be basic, they implicitly therefore have zero cost, thus only the error term variables may be brought into and out of the basis. The user must supply a vector which lists a set of n equations taken from the set m defined by the coefficient matrix A in which the rank of the subset of equations is n, i.e. the subset of n equations are linearly independent. These n equations will then be used to form an initial basic tableau for the Revised Simplex method in the following way.

Both the error terms in equation 5.6 for the subset of the n equations will be defined as being non-basic and thus equal to zero. The remaining m-n equations in equation 5.6 will thus have either a positive or negative error term which is non-zero and is thus said to be a basic variable. The error terms which are basic will be henceforth referred to as basic error variables.

It is worth emphasising at this point that the set of n equations defined on entry by the user are used to calculate initial values for the solution vector, X. Thus if dummy measurement equations in which the values of the input vector B are set to zero are used to set up the initial basis then the error terms on the remaining equations will be large and the total cost of the solution evaluated from the objective function described in equation 5.7 will be high. The dummy equations will quickly have an error term made basic because the cost coefficients, C_d are very small and hence the relative cost coefficient will be negative. This will reduce the values of the error terms of the remaining equations and hence the total cost. Further details on the use of dummy measurement equations for defining the initial basis can be found in section 5.2.

The values of the error terms of the m-n equations which are not used to form the initial basis are calculated on entry to the subroutine in the following way. The coefficients of the subset of the n equations are copied to a separate workspace array. Similarly the values of the input vector, B for the n equations are also copied to a second workspace array. These two

workspace arrays now store the coefficients and input values for a set of linearly independent equations in terms of the n variables originally defined by equation 5.6. The subroutines LA05A and LA05B are used to factorise and solve the equations respectively.

The solution obtained is then substituted into the m-n equations of those defined by equation 5.6 which were not selected to define the initial basis. This enables the value of the positive and negative error terms to be evaluated. Note that either the positive error term, the negative error term or both will have a value of zero, the value of the non-zero term depends on the value of the initial input vector, B. The non-zero error term or an arbitrary choice of either if both terms are zero, is defined as being a basic variable in the initial Simplex tableau. Thus the initial basic tableau is composed of the n solution variables of equation 5.6 plus m-n error terms of equation 5.6, the remaining m+n error terms of equation 5.6 are non-basic and hence have a value of zero. As the n solution variables are constrained to always be basic the Simplex tableau can be constructed by keeping an index of which of the 2m error term variables are basic. It should be noted that the inverse of the basis tableau is stored factorised, i.e. as its LU product, and the subroutine LA05C is used to update the factors on exchanging a basic and non-basic variable.

When the optimal basis has been obtained, there exists a set of n equations in the original set m defined by the coefficient matrix A which have both error terms as non-basic variables. These n equations will be linearly independent and maybe regarded as the equations which define the values of the solution vector X. An index of these n equations is maintained from which the initial basis may be constructed on a subsequent entry to the subroutine. The remaining m-n equations in the set m will have one or other of the error terms as a basic variable. The value of this error term is available and is the error associated with the value of the input vector B for that equation. Thus for a measurement equation which has a basic error variable, the degree of the error in the measured value is given by the value of the error term.

5.1.2 The least squares method

The implementation of the least squares method is a straight forward implementation of the equations given in appendix 4. The subroutine has been called CGOSL (Conjugate Gradient solution of Sparse Linear equations). The user supplies the coefficient matrix, A in sparse form as described at the beginning of section 5.1, together with the input vector, B and an initial

solution vector X_0. The following points concerning the usage of the subroutine should be noted. The initial solution may be set to zero but the solution times are reduced if the initial estimates are good. The algorithm is able to calculate estimates in the case of an under-determined set of equations. However, the initial solution will affect the overall solution point, the algorithm will converge to a solution at which the sum of the squares of the final estimates is a minimum provided the initial solution is zero. This feature is of use in the evaluation of the link power flows as described in chapter 3 on the simulator.

The subroutine has an initialisation section followed by an iterative loop which updates work space arrays representing the vectors, P, Q, R and S in the equations given in appendix 4. The iterative process is terminated in one of two ways. The values of all the elements of the vectors R and P are checked against a predefined tolerance immediately after they have been evaluated, if they are all below the required tolerance, then the process has converged and the most recent values of X are returned as the solution. Alternatively, the number of iterations exceeds a predefined limited and the process is deemed to have failed to converge.

The conditioning of the least squares problem can often be improved by scaling the columns of the coefficient matrix, A so that the value of the largest element is one. This can be achieved by dividing the elements of each column by largest element in that column. This value known as the scale factor needs to be stored and the initial estimates, X_0 supplied to the subroutine multiplied by the appropriate scale factor. Finally the values in the solution vector X need to be divided by the appropriate scale factor. The least squares problem returns the solution to a set of equations defined by

$$AX = B \tag{5.8}$$

It can be shown that a solution to the above equation satisfies the equation

$$A^tAX = A^tB \tag{5.9}$$

which implies that

$$X = (A^tA)^{-1}A^tB \tag{5.10}$$

Equations 5.8 to 5.10 may be modified to preferentially weight some of the

equations and thus result in a weighted least squares solution. Equation 5.4 is pre-multiplied by a weighting vector W, thus

$$WAX = WB \tag{5.11}$$

which gives the solution

$$X = ((WA)^t(WA))^{-1}(WA)^tB \tag{5.12}$$

$$= (A^tW^tWA)^{-1}A^tW^tB$$

The product W^tW is equivalent to the weight and hence the values of the vector W are set equal to the square root of the required weight.

5.1.3 Solution Times

A direct comparison between the solution times of both methods of solving a linear set of problems has not been made, however the overall solution time for both methods when applied to the entire estimation process indicates that under similar conditions the least squares method is likely to be between 2.0 and 2.5 times faster than the linear programming method. Further details on these comparisons can be found in chapter 6.

It is worth mentioning that the solution times for the 2 methods depend significantly on the initial basis for the Simplex method and the initial values of the estimates for the least squares method. Also the solution time for the Simplex method is further increased by the addition of 'dummy' equations adjoined to the system to facilitate the formation of the initial basis. Thus a direct comparison of the solution times is not very meaningful.

The only direct measurement of the solution times for the least squares method was made on the estimation of the link power flows in the simulator program. The initial estimates for the link flows are zero and there are 55 equations defining the flows in 68 links. The problem is under-determined, however as mentioned in the previous section, the subroutine returns a solution which minimises the sum of the squares of all the link flows. The solution times for this system vary between 0.18 seconds and 0.21 seconds depending on the values of the input vector B.

More detailed results on the solution times for the Revised Simplex method have been obtained to enable a comparison to be made between the use of linear programming techniques and network flow techniques for solving the active and

reactive power flow sub-estimation problems. Further details on the network flow techniques can be found in chapter 7. Table 5.1 lists the number of equations and variables in each of the four sub-estimation stages for the estimation of the states of the 30 substation test network. The top half of table 5.2 lists the solution times together with the number of Simplex iterations in brackets for each of the five iterations required to obtain convergence from a flat start. The bottom half of the table lists the solution times when line 7 was opened while the program was running, note in this case both the active and reactive sub-estimation stages have an additional equation which defines the power flow through the open breaker to be zero. Table 5.2 illustrates the reduction in the solution times when on the second and subsequent main iterations the subroutine has a good initial Simplex tableau from which to form the initial basis.

Tables 5.3 and 5.4 list a similar set of results obtained from the 118 substation test network. In the 30 substation example the generators each had two equations constraining both the active and reactive power flow estimates to lie within an upper and lower limit and the loads had one equation constraining the active power flow estimate to be less than zero. However, in the 118 substation example these equations were removed as their worth seems limited and they increase the size of the problem unnecessarily.

Unfortunately the solution times for both methods do not have a linear relationship with the problem size. The CPU time required for each iteration will increase with the problem size as will the total number of iterations required to obtain the solution. This can be seen by comparing the solution times for the voltage sub-estimation problems, the ratio between the number of variables in the 30 substation example and the 118 substation example is in the order of 1:1.6 while the ratio of the solution times is in the order of 1:3.5. A similar comparison may be made for the power flow sub estimation problems but an allowance must be made for the additional generator and load constraints on the 30 substation example.

5.1.4 Early termination of the Simplex method

The objective function of the Simplex method minimises the weighted sum of the modulii of all the error terms as explained in chapter 4. The selection of the most negative relative cost coefficient ought to cause the value of the objective function (the total cost) to fall rapidly initially and then gradually level out. A plot of the total cost against the iteration number would be expected to be of the form illustrated in figure 5.1 The analogy

Table 5.1: Comparison of the number of equations and variables for the four sub-estimation stages on the 30 substation test network

Sub-estimation type	Number of equations	Number of variables
Active power flow (P)	337	181
Reactive power flow (Q)	312	181
Voltage magnitude (V)	182	73
Voltage phase angle (θ)	183	73

Table 5.2: Solution times for the Revised Simplex method during the estimation of the states of the 30 substation test network

The times are in seconds and the number of Simplex iterations required to converge each sub-estimation stage is shown in brackets.

Solution times obtained from a flat start

Iteration number	Sub-estimation type P	Q	V	θ
1	14.67 (118)	15.05 (131)	8.07 (119)	11.12 (157)
2	1.58 (6)	2.86 (18)	1.23 (11)	0.85 (6)
3	4.00 (25)	2.33 (13)	0.71 (4)	0.73 (4)
4	2.90 (16)	3.63 (24)	0.72 (4)	0.73 (4)
5	2.48 (13)	2.97 (19)	0.74 (4)	0.72 (4)

Solution times obtained continuing from the above estimates with line 7 open

Iteration number	Sub-estimation type P	Q	V	θ
1	6.16 (43)	6.14 (46)	3.09 (36)	2.11 (22)
2	2.85 (16)	5.71 (43)	0.85 (6)	1.82 (17)
3	1.48 (5)	1.53 (6)	0.71 (4)	0.96 (7)
4	1.34 (4)	2.39 (14)	0.71 (4)	0.74 (4)

Table 5.3: Comparison of the number of equations and variables for the four sub-estimation stages on the 118 substation test network

Sub-estimation type	Number of equations	Number of variables
Active power flow (P)	814	517
Reactive power flow (Q)	814	517
Voltage magnitude (V)	297	118
Voltage phase angle (θ)	298	118

Table 5.4: Solution times for the Revised Simplex method during the estimation of the states of the 118 substation test network

The times are in seconds and the number of Simplex iterations required to converge each sub-estimation stage is shown in brackets.

Solution times obtained from a flat start

Iteration number	Sub-estimation type P	Q	V	θ
1	81.07 (281)	84.61 (297)	22.76 (205)	47.00 (367)
2	7.31 (18)	27.18 (85)	11.85 (95)	6.46 (46)
3	6.64 (15)	21.76 (67)	6.35 (48)	5.18 (36)
4	18.02 (53)	6.81 (16)	3.73 (25)	4.84 (32)
5	11.48 (31)	5.40 (11)	3.31 (22)	2.45 (15)
6	9.58 (25)	5.35 (11)	5.15 (37)	2.88 (18)

Solution times obtained continuing from the above estimates with line 50 open

Iteration number	Sub-estimation type P	Q	V	θ
1	23.07 (70)	21.09 (68)	9.01 (71)	6.86 (50)
2	12.27 (33)	14.43 (41)	8.36 (65)	3.49 (23)
3	7.04 (16)	5.43 (11)	3.03 (20)	2.37 (14)

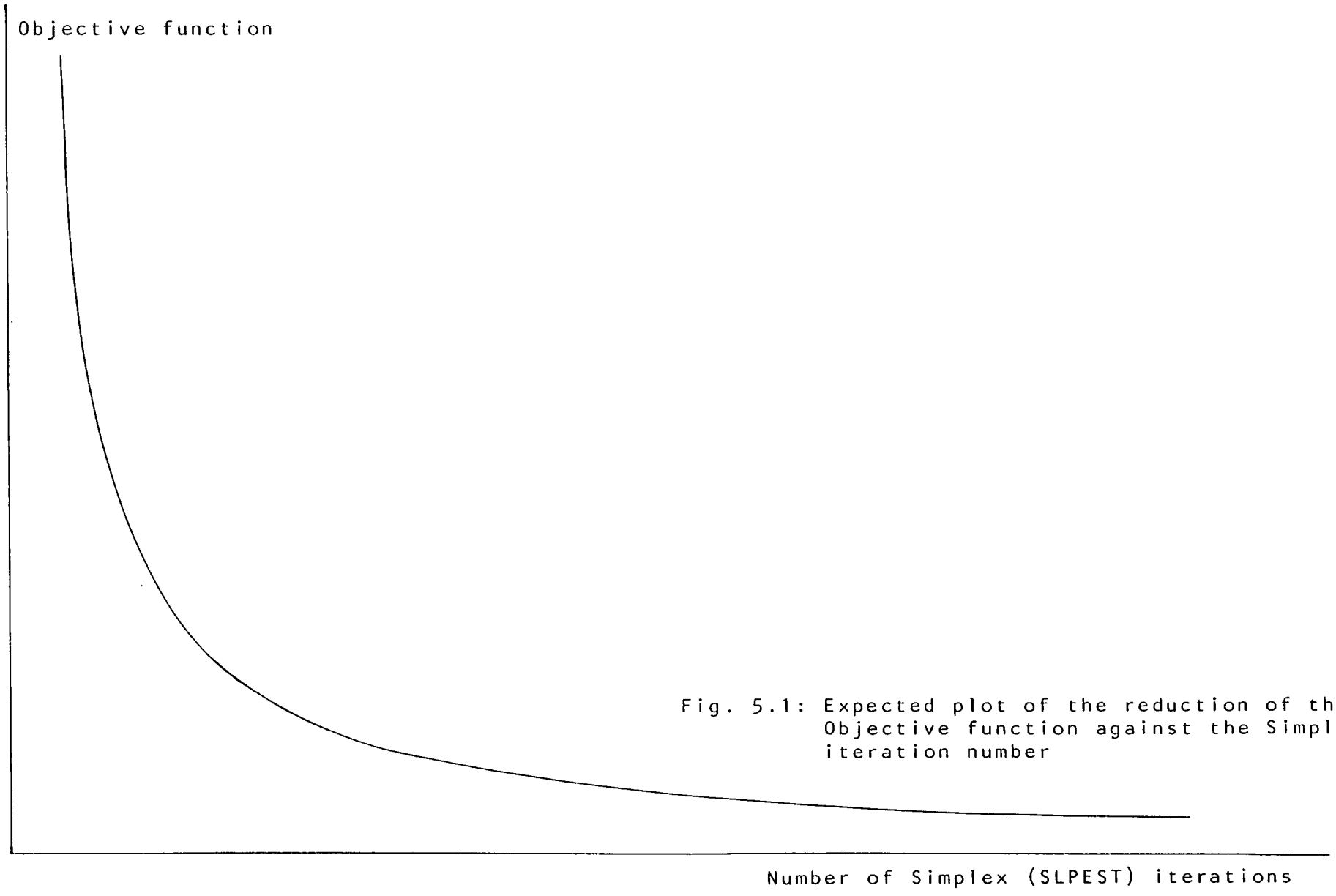


Fig. 5.1: Expected plot of the reduction of the Objective function against the Simplex iteration number

between the expected curve and the equations defining the estimates of the network is as follows. Initially the measurement equations with a corrupt measurement value would have an error term variable brought into the basis if one was not already present. The definition of a bad measurement equation includes the dummy equations introduced to set up the initial basis. This would cause the total cost to fall significantly with each iteration. Then the remaining error term variables would be swapped in and out of the basis as the method selects the least noisy set of equations to define the estimates. During this stage the magnitude of the change of the estimates would be small and hence the change in the value of the objective function should also be small. Thus it was hoped that the Simplex process could be terminated early once the change in the objective function fell below a pre-defined limit for three successive iterations, especially during the first solution of each of the 4 sub-estimation stages.

The estimates produced during the first solution of the sub-estimation stages are only approximate as the equations relating the estimates at the opposite ends of a line are not generally valid on the first iteration. Hence once the measurement equations with large errors have been identified the estimates will be fairly accurately defined and the Simplex process could be halted. However, the plots of the total cost against the iteration number of the Simplex process were obtained for several different estimation problems. The four plots shown in figures 5.2 to 5.5 were obtained from the first solution of each of the four sub-estimation stages for the state estimation program on the 30 substation test network in its steady state. As can be seen, the shapes of the curves differ widely and follow no general trend. The results for the second and third solutions of the active power sub-estimation stage (ie the second and third main iterations) are shown in figures 5.6 and 5.7. The magnitude of the total cost has changed in each case because the errors on the active power flow line loss equations have been reduced as convergence is approached. There is still no general trend even for an individual sub-estimation type and the random nature of the curves has made selecting a suitable point at which to halt the Simplex process before the normal termination point very difficult. The final solution to this problem was to set the limit on the rate of change of the total cost to 0.0001 times the total cost at the end of the previous solution for that stage. It should be noted that the total cost is initialised to zero on the first main iteration. The number of successive iterations for which the change in the total cost must be less than the tolerance defined above has also had to be set to allow for the large flat regions found during the early stages of some

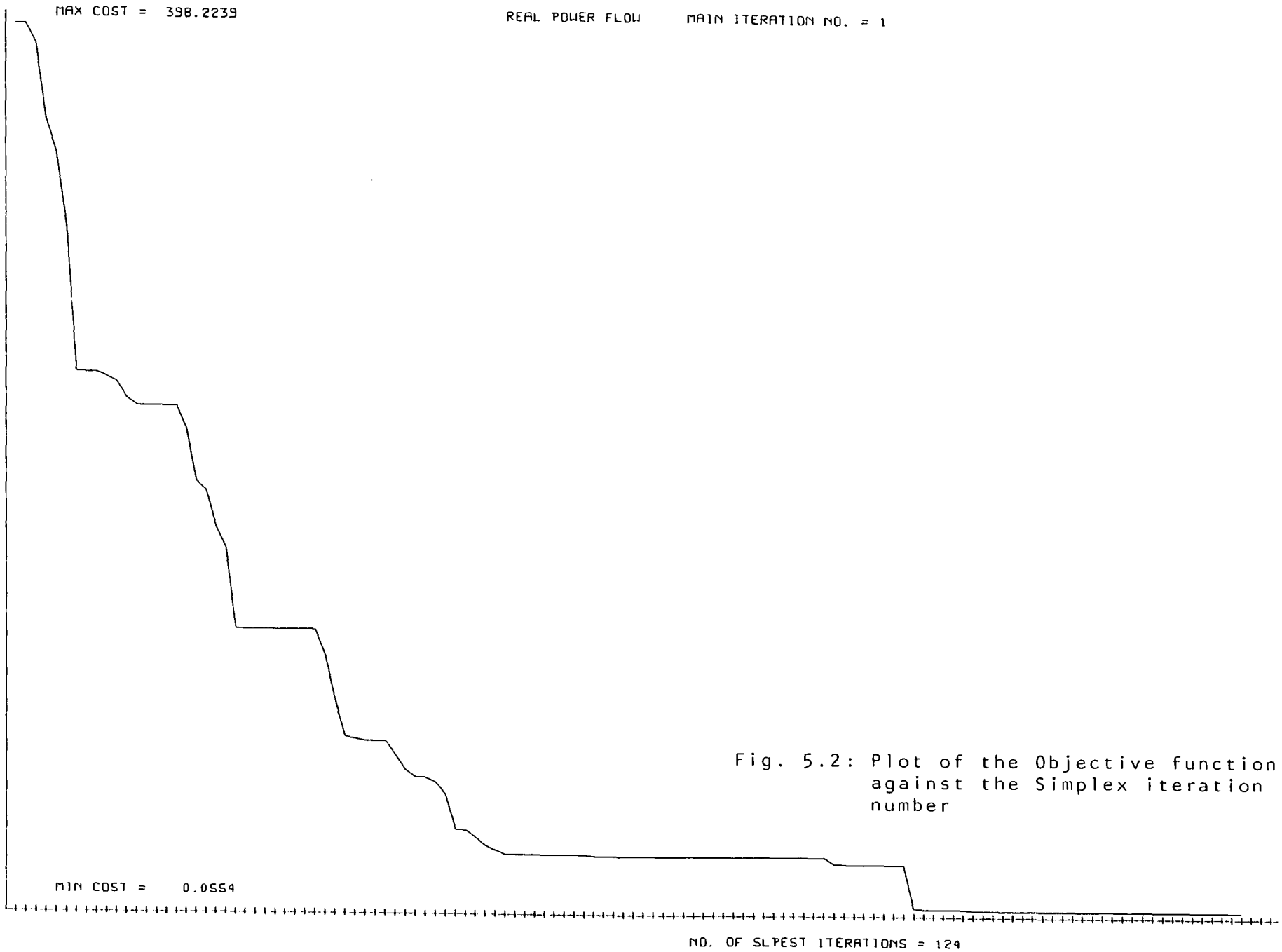


Fig. 5.2: Plot of the Objective function against the Simplex iteration number

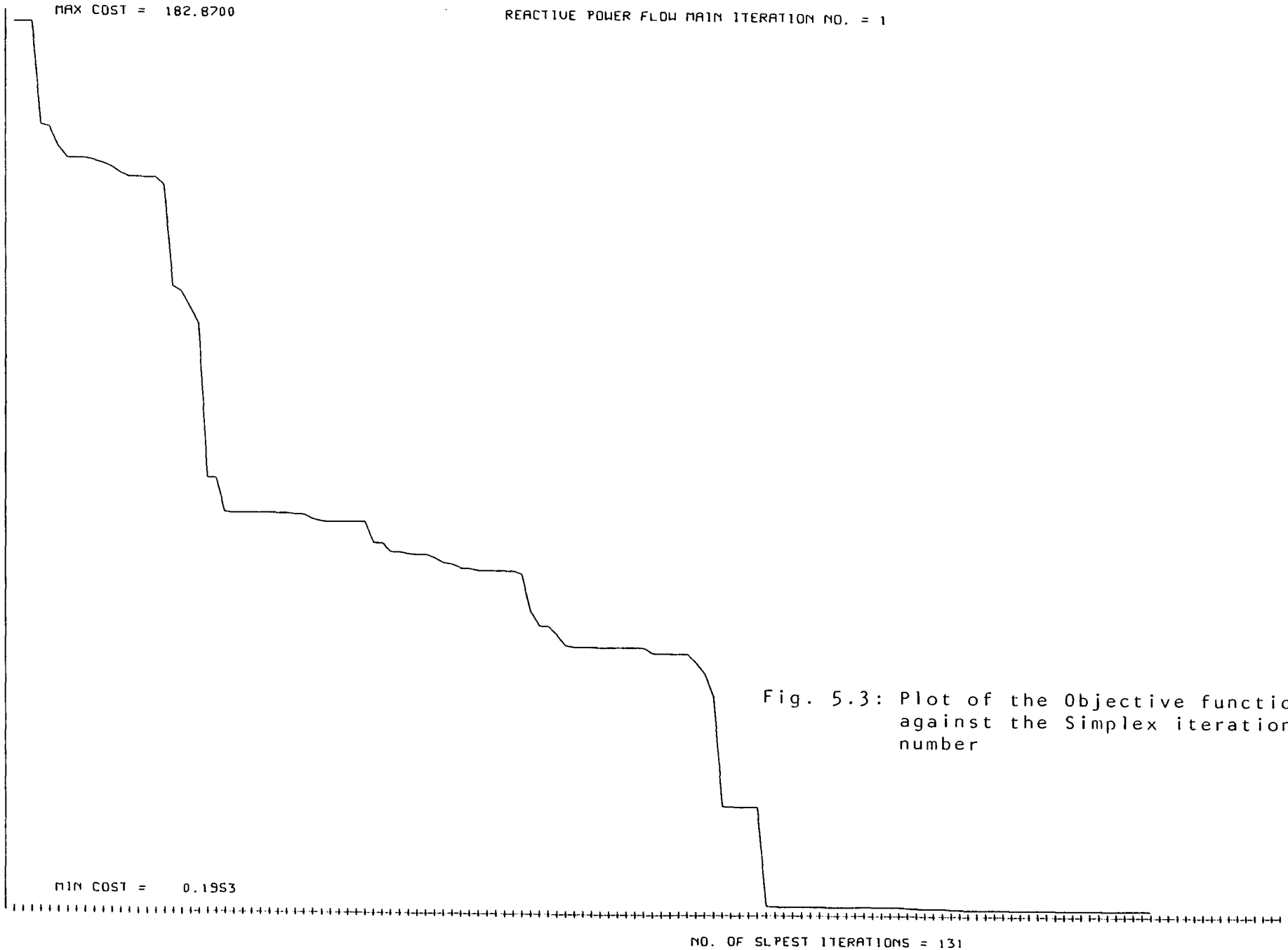


Fig. 5.3: Plot of the Objective function against the Simplex iteration number

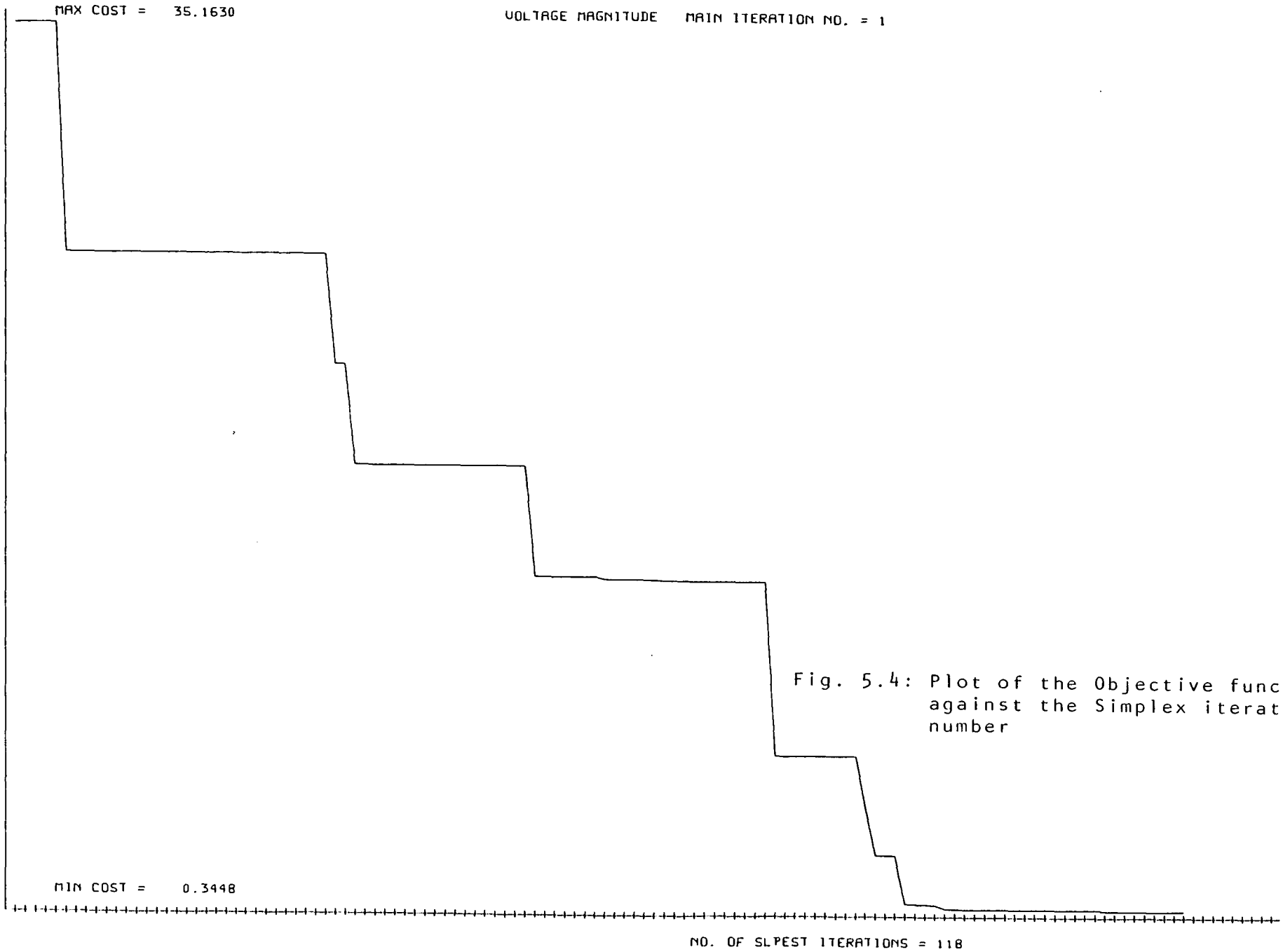


Fig. 5.4: Plot of the Objective function against the Simplex iteration number

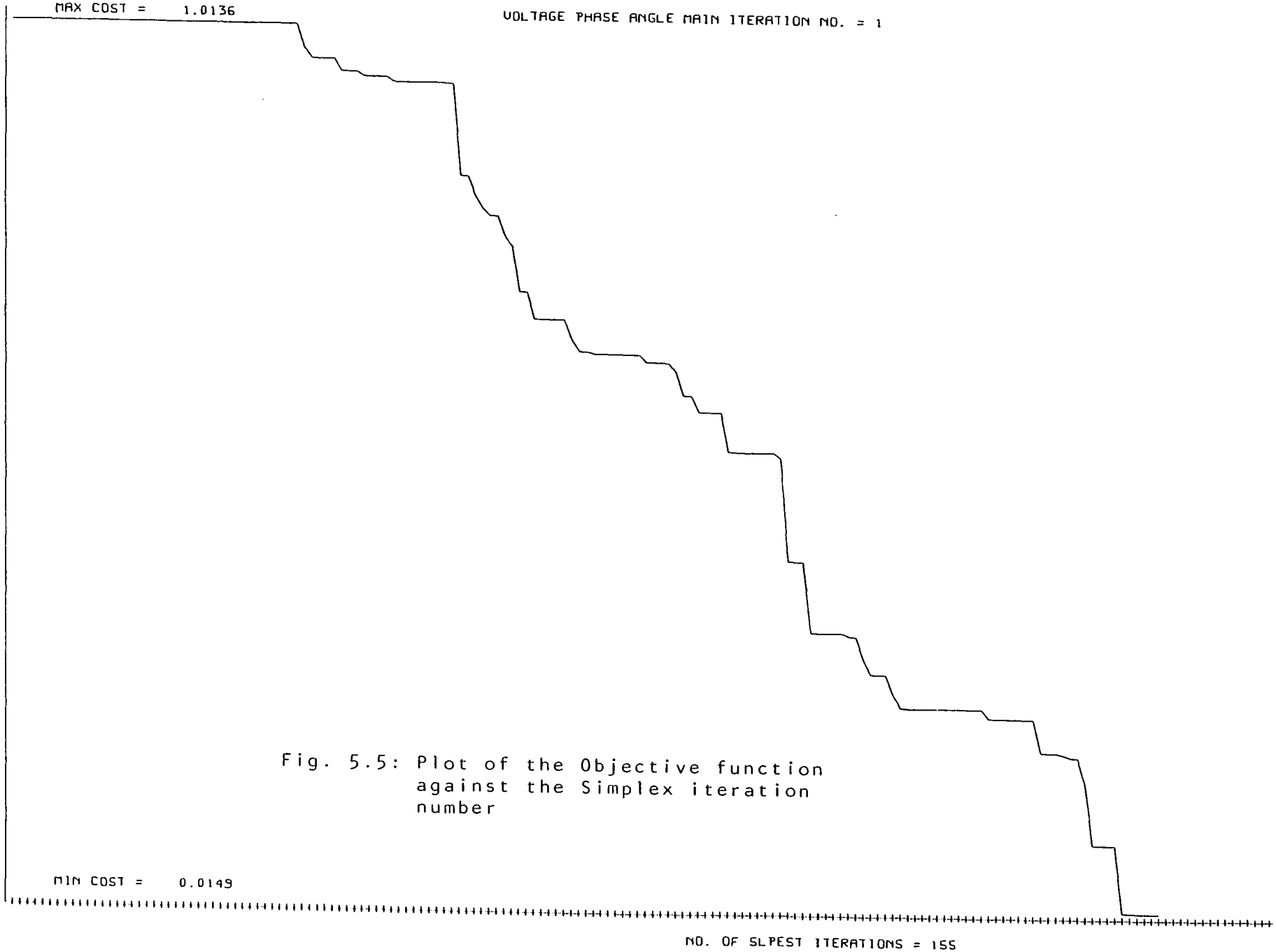


Fig. 5.5: Plot of the Objective function against the Simplex iteration number

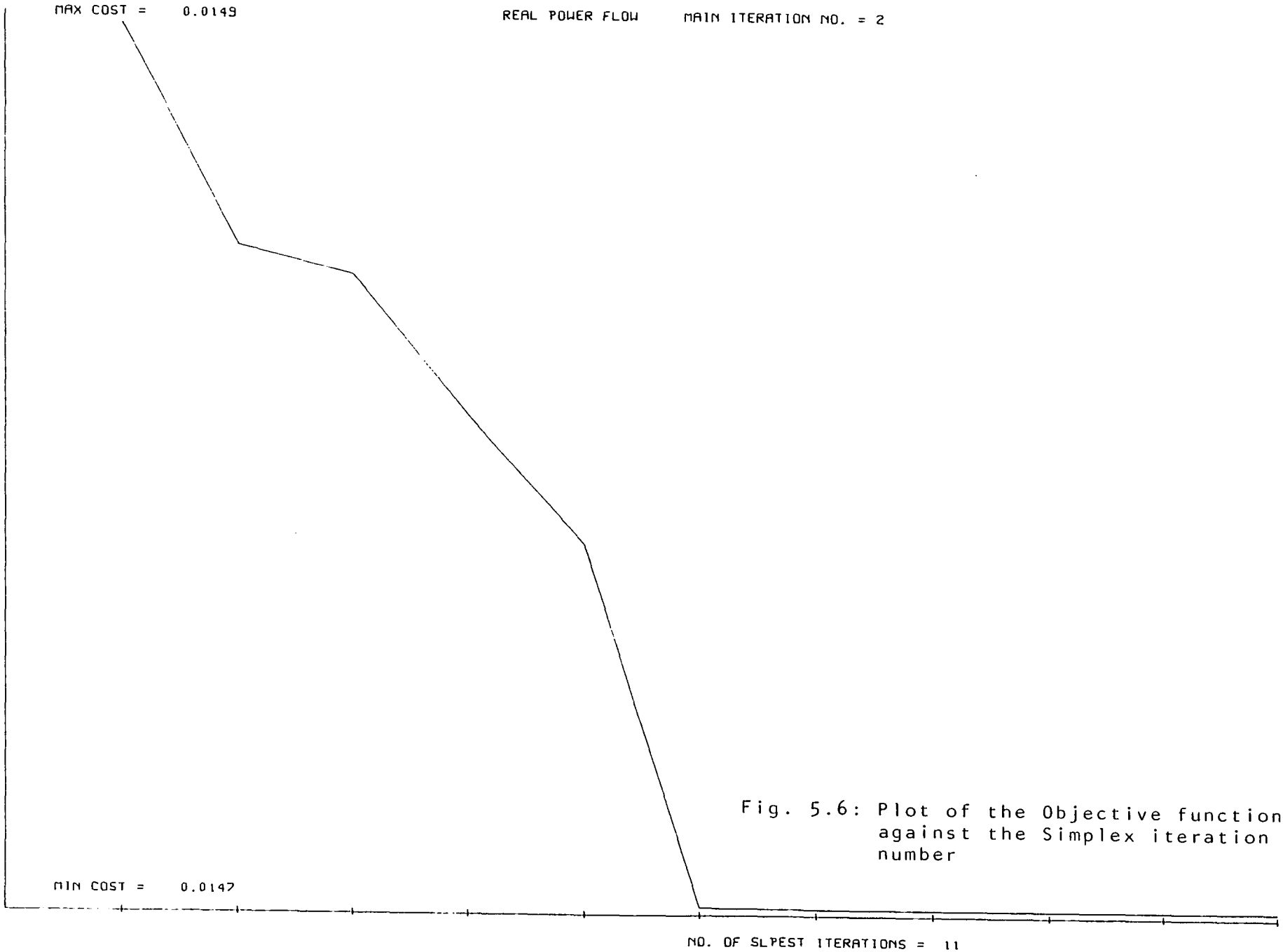


Fig. 5.6: Plot of the Objective function against the Simplex iteration number

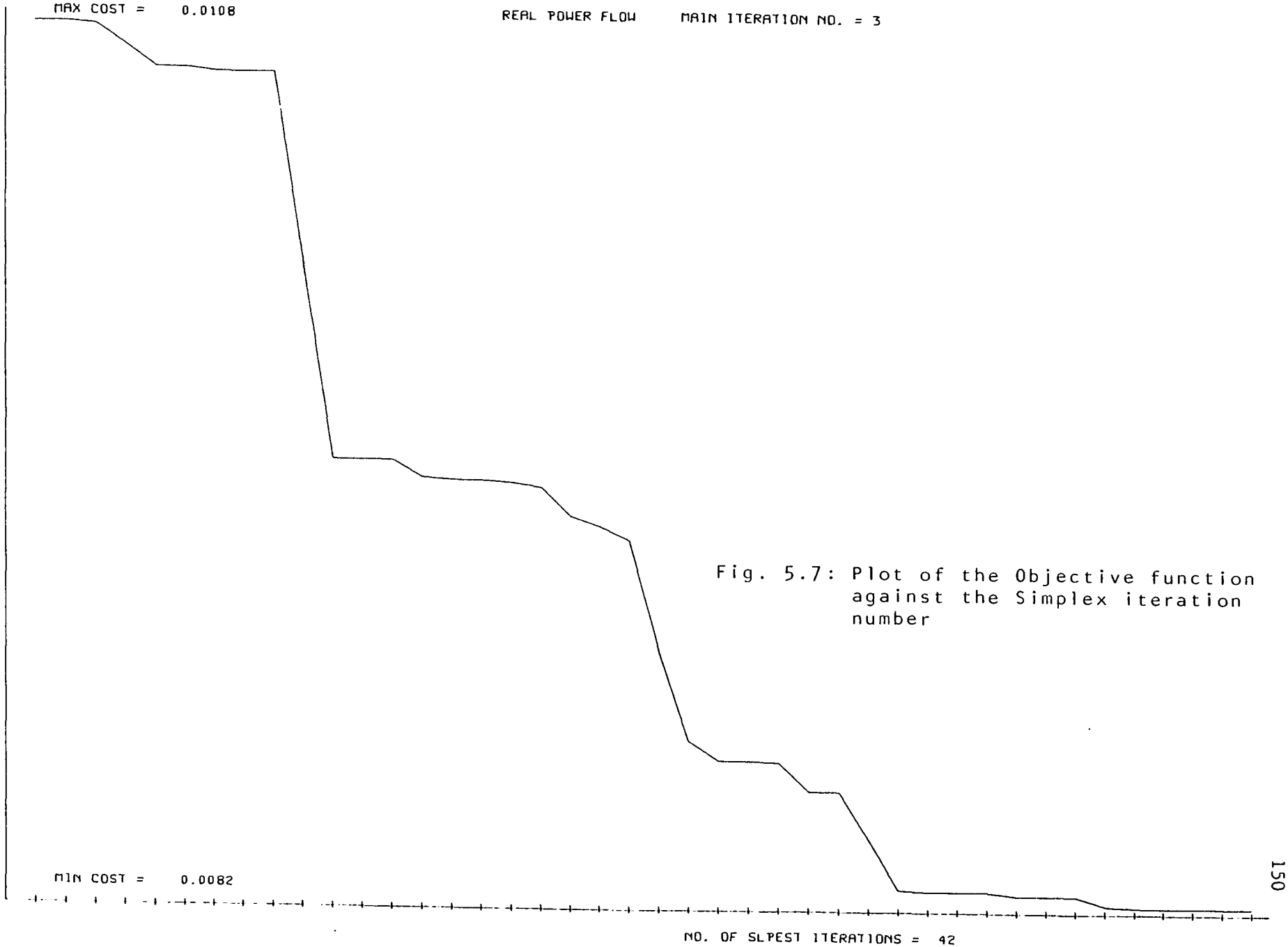


Fig. 5.7: Plot of the Objective function against the Simplex iteration number

of the graphs. To try and improve the performance of the early termination section of the Simplex method this value is set according to the sub-estimation type, ie active power, reactive power etc. and the main iteration number of the overall estimation process, ie the number of times each sub-estimation stage has been invoked. The required number of iterations for early termination has been set according to table 5.5.

The success of the early termination section has been rather limited although it does reduce the number of Simplex iterations on the first main iteration from a flat start by about 15%. Further reductions in the number of Simplex iterations are not advised as experimental evidence suggests that additional main iterations would be required if the Simplex process is terminated too soon.

The large flat regions of the graphs for the voltage magnitude and phase angle sub-estimation problems arise because of the large difference between the weighting factors of the dummy measurement equations and the line difference equations. This allows the Simplex process to readily exchange the basic and non-basic error variables of these equations, however the initial input values of the line difference equations are close to zero which means the values of the estimates do not change significantly upon the basis exchanges and hence the total cost does not change.

5.2 Implementation of the four sub-estimation stages

As is the case with all the programs, the state estimation algorithm has been programmed in Fortran 77[7], compiled using an optimising compiler and run on a Perkin-Elmer 3230 minicomputer. The processor uses 32 bit words and has a floating point arithmetic hardware unit.

The state estimation algorithm proposed in the thesis comprises of 4 distinct sub-estimation stages, each of which involves the solution of an overdetermined set of linear equations. Further details on the formulation of the linear equations can be found in chapter 4. The four sub-estimation stages are solved consecutively in an iterative process, 1 iteration being defined as the single solution of each of the 4 stages.

The process is said to have converged when in any given iteration the change in magnitude of all the estimates in each of the four stages is less than a pre-defined tolerance.

Table 5.5: Number of successive Simplex iterations with little change in the objective function value required for early termination.

Sub-estimation type	Main iteration No. 1	2	3 or more
active power flow	10	5	3
reactive power flow	10	5	3
voltage magnitude	30	5	3
voltage phase angle	50	5	3

5.2.1 Selection of the initial basis for the Simplex process

The linear programming algorithm solves an over-determined set of linear equations by minimising an objective function which is calculated from the weighted error terms associated with each equation. The subroutine SLPEST which implements the Revised Simplex method for solving the linear programming problem requires that the user supplies a vector defining a subset of the equations which are linearly independent and hence may be used to form the initial basic tableau. Further details on the Revised Simplex method and its implementation can be found in section 5.1.1 and appendix 4.

The selection of a linearly independent set of equations from a typical set found in the four sub-estimation stages is unfortunately a non-trivial process. The method initially adopted facilitates the initial selection of the required set of equations and also the replacement of an open switch status measurement equation should one have been selected as being the subset of n equations which will be used to reform the basis at a later date and the status subsequently changing to closed, thus making the equation invalid and resulting in the basis matrix no longer being of full rank.
The easiest way to select a linearly independent subset of equations from the original over determined set of equations defined by the product

$$AX \tag{5.13}$$

where

A = m*n coefficient matrix, m>n.

X = solution vector of length n.

is to select the coefficients such that the subset of equations is defined by

$$BX \tag{5.14}$$

where

B = n*n diagonal matrix.

The resulting subset of equations can be interpreted as defining each state by an equation with only one term in it. The only type of equation amongst those available which complies with the above condition are the measurement equations, eg. equation 4.31. It is unlikely that all the states will have a measurement available, especially in the voltage phase angle sub-estimation problem, however the addition of artificial or dummy

measurement equations in which the variable is equated to either zero or to a pseudo measurement value overcomes this difficulty. It should be noted that the value of all the dummy measurement equations has been set to zero in the program. The weighting factors of these equations must be low so the error terms may be allowed to become basic without incurring a large increase to the objective function. Thus the initial basic tableau can be found from the measurement equations together with the required number of dummy measurement equations associated with the states which do not have a real measurement available.

The additional burden this method puts on the linear programming method can be seen in tables 5.2 and 5.4 which show the solution times for the Simplex process for each of the four sub-estimation stages as the program approaches the solution point from a flat start. In the first main iteration, a considerable number of Simplex iterations and hence time is spent in exchanging the error terms of those dummy measurement equations which were initially non-basic for error terms of other equations which were defined as being basic in the initial tableau. On subsequent main iterations the previous tableau is used as the starting point and hence these error terms are already basic and are not reprocessed. However, time is still spent in checking to see whether they are suitable to leave the basis and furthermore the dimensions of all the program arrays are increased to allow for the additional equations which thus increases the time taken to update the arrays. Thus a method of selecting the subset of equations to form the initial basic tableau avoiding the use of dummy meassurement equations would not only reduce the solution times, especially on the first solution of each of the four stages, but also the storage requirements of the overall state estimation program.

5.2.2 Addition of the open switch measurement equations

An open switch provides additional information for the estimation of the active and reactive power flows by constraining the flow through the switch to be zero, however if the switch is in a link or line it decreases the information available for the estimation of the voltage levels because the equations defining the difference between the voltage levels at each end of the link or line is no longer valid.

In the majority of cases, a power system is operated with the majority of the busbars and lines connected to the system. This makes the system more secure and more able to survive a large transient disturbance without

interrupting the supply. The number of open switches in the system at any one time is thus likely to be low. Therefore in the voltage sub-estimation problems should a link or line have an open switch then the weighting factors on the error terms for that link or line are set very low so that the equation may easily be violated and not affect the estimates or the value of the objective function. The value of the weighting factors for these links or lines (typically 0.00001) has been set lower than those of the dummy voltage measurement equations (typically 0.001) so that in the event that the bus-section becomes isolated, the program will either assign the value of the dummy measurement equation to that estimate, that is zero, or the value of a genuine measurement equation which ought to be zero. If the measurement equation does not have the value of zero, then the program will not be able to correct the value in this case.

If a generator, load, link or line has an open switch, then an equation of the appropriate form is adjoined to those defining the estimates of the active and reactive power flow. The implementation of the Simplex method allows the coefficients of the equations to be entered in any order, thus the coefficients of the open switch equations can be added to the ends of the arrays storing the information on the other equations. When the switch status returned to closed the weighting factors on the link or line equations for the voltage level sub-estimations problems are returned to their normal value (typically 1.0) and the additional equation for the power flow sub-estimation problems deleted from the ends of the vectors storing the information. The state estimation program has been dimensioned to allow for approximately ten percent of all the branches to have an open switch measurement. A check is made to ensure that the limit is not exceeded and a warning is issued should this occur, the program will continue to operate but the additional information on the power flow estimates is not included in the active and reactive power flow sub-estimation problems. This may cause a degradation in the performance of the program in the event that there are measurement errors associated with the branches where the open switch measurement equations have not been included.

5.2.3 Solution order of the four sub-estimation stages

The state estimation algorithm consists of 4 sub-estimation stages, the solution order of which might be expected to affect the overall convergence characteristics. As in the case of a load flow solution, the choice arises as to the point at which the estimates are updated with the newly calculated

values, the options being all together at the end of all 4 sub-estimation solutions. (as in a Gauss load flow solution[124]), progressively as each new set of values becomes available after each sub-estimation solution (as in a Gauss-Seidel load flow solution[124]) or a combination of both the above options whereby 2 of the 4 sub-estimations problems are solved and their estimates then updated with the new values before the remaining two sub-estimation problems are solved. The total number of combinations of sub-estimation solution order and the update points is large and hence only a few of the possibilities have been investigated. These are summarised in the following list, the closing of a pair of brackets signifies the point at which the estimates are updated with the newly calculated values, thus a single sub-estimation stage in brackets means the estimates are updated as the new values become available while a pair of sub-estimation stages in the brackets signifies that both stages are solved before the estimates are updated.

1. (P) (Q) (V) (θ)
2. (PQV θ)
3. (P) (θ) (Q) (V)
4. (θ) (P) (V) (Q)
5. (P) (θ) (V) (Q)
6. (P θ) (QV)
7. (P) (V) (Q) (θ)
8. (PV) (Q θ)

where

P = active power flow sub-estimation problem.
Q = reactive power flow sub-estimation problem.
V = voltage magnitude sub-estimation problem.
θ = voltage phase angle sub-estimation problem.

The solution order of two or more of the sub-estimation problems before the update of the estimates is immaterial, that is the solution and estimate update of (P θ) is identical to that of (θ P). The various choices investigated were derived by considering the natural coupling between the active power flows and voltage phase angles together with the coupling between the reactive power flows and voltage magnitudes. This coupling is made use of in the fast decoupled load flow technique.[130] The ordering or grouping together of the sub-estimation problems in such a way so as to either follow the natural coupling or alternatively to oppose it was expected to have an

effect on the overall convergence characteristics. Furthermore, it was noticed during the trials that on successive iterations a few of the reactive power flow estimates would oscillate between the values differing by about 0.0005 p.u. under certain circumstances and hence the solution order of the voltage magnitude sub-estimation problem and the reactive power flow sub-estimation problem were interchanged to try and cure this phenomenon. However, the various choices of the solution order and update point of these two sub-estimation problems did not prevent this small oscillation.
Fortunately, the method adopted for testing for convergence does not explicitly compare values of the new and old estimates and hence this oscillation is not generally observed. Further details on the method of testing for convergence can be found at the end of the section. The results of the various combinations listed above indicated that the choice made no difference to the reliability of the estimates but in some cases, such as the second an additional iteration was required when calculating the estimates from a flat start.

The final choice adopted was that of number one for the following two reasons. Firstly the options in which the estimates were updated as the values became available generally converged in fewer iterations than those where two or more of the sub-estimation problems were solved before an update. Secondly in the test networks the active and reactive power flow sub-estimation problems have a higher ratio of the number of measurements to the number of estimates than the voltage magnitude sub-estimation problem, the voltage phase angle sub-estimation problem has no measurements at all. Thus should there be any section of the network from which the measurements have been lost due to a telemetry failure, then the active and reactive power flow sub-estimation problems ought to produce more reliable estimates on the first iteration than the voltage sub-estimation problems which should then enable to program to converge to the solution point quickly.

The choice of the weighting factors of the error terms did make a small difference on the convergence characteristics of the algorithm. It has been observed that the algorithm is slightly better at detecting bad measurements if the weighting factors of the error terms for the measurement equations were less than those for the equations defining the power flow loss in and voltage drop across a line. An arbitrary reduction of 40 percent gave good results, the weighting factors for the measurement equations being set at 0.6 and those of the line equations at 1.0. The reason for this is that smaller weighting factors facilitates a less punitive violation of the equation, thus a bad measurement equation may more easily be violated if the weighting factors are

small allowing the correct estimate to be calculated from other equations defining that state. An interpretation of this situation is that instead of using the line equations to verify that the measurements conform to the laws governing power flow in a network, the measurements are used as a form of constraint on a load flow solution.

The final point worth noting on the convergence of the four sub-estimation problems is the method of determining when convergence has been achieved. Two alternative methods are available, the first is to compare the newly calculated value of every estimate with the previous value and check that the change in the value is below the required tolerance, the second is to monitor the change in the value of the objective function for each of the four sub-estimation problems.

The second alternative was adopted as it involves less computation and a good correlation exists between the change in the value of the objective function and the change in the values of the estimates. If the value of the objective function changes by less than 5 percent of the previous value then the sub-estimation problem was said to have converged, if all four sub-estimnation problems have converged in any one iteration, then the state estimation algorithm was said to have converged. This method also has the advantage of masking any small oscillations in the values of some of the estimates which occurs under certain circumstances. The reason for this being that a small change in the values of the estimates does not significantly affect the value of the objective function.

It is believed that these small oscillations are caused by the degenerate nature of the equations which arise under certain circumstances when the measurements have no noise or bad data added to them. In this situation a few of the estimates in each of the sub-estimation problems are defined by several equations which equate the estimates to similar values. During one iteration the Simplex process swaps the error terms of these equations in and out of the Simplex basis and thus slightly alters the values of the estimates. The natural coupling between the reactive power flow estimates and the voltage magnitude estimates means that a small change in the values of a few of the reactive power estimates gives rise to a small change in a few of the corresponding voltage magnitude estimates. On subsequent iterations, the small change in those voltage magnitude estimates allows the original reactive power flow equation error terms to re-enter the Simplex basis and hence the values of the estimates oscillate from one iteration to the next. This phenomenon has not been observed with the active power flow and voltage phase angles estimates because the voltage phase angle sub-estimation problem has no

measurements and the degenerate definition of the estimates does not arise. The magnitude of oscillations was originally only noticed on the reactive power flow estimates because the voltage magnitude estimates are solved in terms of the square of the voltage magnitude, hence a small change in the square of the estimate gives approximately half this change in the estimate. The magnitude of the oscillations is less than 0.0005 per unit and is thus small enough to be neglected.

Unfortunately the problem of degenerate equations seriously affected the performance of a linear programming version of the of the Newton Raphson State estimation algorithm. Further details on this problem can be found at the start of chapter 6.

5.2.4 A summary of the sequence of steps in the state estimation algorithm

The following sequence of steps summarises the operation of the state estimation program.

1. Declare all arrays and initialise as appropriate.
2. Initialise the estimates of active and rective power flow and the voltage phase angle to zero, the voltage magnitude estimates to one.
3. Set up the measurement equations for all the sub-estimation stages.
4. Set up the line power flow loss equations and the voltage drop equations for all the lines. Set up the equations defining the difference between the squares of the voltage magnitude estimates at the terminating bus sections of all the links to be zero. Similarly for the voltage phase angle estimates.
5. If a switch status change has occurred, go to step 9.
6. If the operator requests an update of the estimates, go to step 9.
7. If the time limit since the last estimate update has expired, go to step 9.
8. Wait for 1 second, go to step 5.
9. Read all new measurement values and switch status information.
10. Amend the open switch equations for the active and reactive power flow sub-estimation stages as appropriate.
11. Reduce the weighting factors on the voltage drop equations for any lines or links which has an open switch, reset the weighting factors for those lines or links which have been re-connected to the network.
12. Calculate the active and reactive power flow losses in all the lines from

the present estimates of the voltage magnitude and phase angle.

13. Solve the set of linear equations for the active power flow sub-estimation stage, update the estimates and check for convergence.
14. Solve the set of linear equations for the active power flow sub-estimation stage, update the estimates and check for convergence.
15. Calculate the difference between the squares of the voltage magnitude at the terminating nodes of all the lines from the present estimates of the active and reactive power flows together with the voltage phase angle estimates.
16. Solve the set of linear equations for the square of the voltage magnitude sub-estimation stage, evaluate the square root and update the voltage magnitude estimates, check for convergence.
17. Calculate the voltage phase angle drop across all the lines from the present estimates of active and reactive power flow together with the voltage magnitude estimates.
18. Solve the set of linear equations for the voltage phase angle sub-estimation stage, update the estimates, check for convergence.
19. If all four sub-estimation stages have converged, go to step 21.
20. Go to step 12.
21. Output the values of the new estimates, go to step 5.

5.2.5 Reduced linear programming problem size

The implementation of the Revised Simplex method requires on input an index to a linearly independent sub-set of equations from which the initial basic tableau can be constructed. The selection of a linearly independent sub-set of equations is a non-trivial process and the method used involved the addition of a dummy measurement equation for every unmeasured point in the network. The initial basic tableau was then formed using both the real and dummy measurement equations. Further details on the implementation of the Revised Simplex method and the use of dummy measurement equations can be found in sections 5.1.1 and 5.2.1 respectively.

The additional dummy measurement equations increase the solution times of the method for two reasons. Firstly, the extra equations require additional CPU time when searching for suitable variables to enter and exit the basic tableau since the dimensions of the storage arrays are larger, more time is also required to update the inverse of the basic tableau which is stored in a factorised form. Secondly, on the first main iteration the additional dummy

measurement equations have both the error term components stored as non-basic variables. The low weighting factors of these error term components means that they are good candidates for entering the basis. Thus during the concluding stages of the research, an investigation was carried out to ascertain the gains in solution times if the dummy measurement equations were removed from each of the four sub-estimation problems.

A considerable amount of research has been conducted into network observability using graph theory techniques[34,43] and it may be possible to adapt one of these methods to the problems of forming the initial basic tableau for each of the four sub-estumation problems, however in the present work the following methods were used.

Providing that at least one voltage magnitude measurement exists, then both the voltage magnitude and the voltage phase angle sub-estimation problems are observable and do not require any dummy measurement equations. A linearly independent sub-set of equations can be selected from the full set for each of the voltage sub-estimation stages by following the procedure outlined below.

1. Initialise a workspace array of length equal to the number of bus sections to zero.
2. Scan through all the measurement equations for the voltage sub-estimation problem under consideration. If the value of the element in the workspace array corresponding to the bus section at which the measurement is being made is zero, then enter the measurement equation into the sub-set and set the value of the element in the workspace array to one.
3. Scan through all the link and line voltage difference equations. If the values of the elements of the workspace array corresponding to the terminating bus sections of the link or line are one and zero (or zero and one), then enter the equation into the sub-set and set the values of both the elements of the workspace array to one.
4. Repeat step 3 until all the values of the elements of the workspace array are one.

The selection of the sub-set of equations for the power flow sub-estimation problems is complicated by the fact that the problems are not observable on the 30 substation test network and a few dummy measurement equations are required to define the link power flows in a number of the substations where there are insufficient link power flow measurements. The bus-section power flow sum check equations which equate the algebraic sum of the power flows in two or more elements make it difficult to ensure that the

sub-set of equations are linearly independent.

The method adopted for selecting the subset of equations follows the procedure below.

1. Set up the sub-set of equations in the usual way using dummy measurement equat ions for all unmeasured points.
2. Read a set of power flow measurements for the power flow sub-estimation problem under consideration from the task common block.
3. Solve the problem in the usual way.
4. The dummy measurement equations which have a basic error term are not required and can be deleted from the full set of equations. Those which have both error terms as non basis variables are required to maintain observability and cannot be deleted from the full set of equations.

This method of selecting the equations from which to build the initial basic tableau for the power flow sub-estimation is very inefficient and hence time consuming. However, it was quickly implemented and enabled the gains in solution times on the reduced problem size to be determined.

The results of the implementation of the state estimation program with the unnecessary dummy measurement equations removed can be found in appendix 6 and the implications of this technique are discussed in the concluding chapter, chapter 9.

Chapter 6
Results of the state estimation algorithm

6.1 Presentation of the results

The implementation of the 4 stage decomposed linear state estimation program has been described in the previous chapter. This chapter presents a discussion of the results and solution times of the program in a selection of operating conditions. The program was developed and initially tested using a development compiler but was re-compiled using the Fortran 77 'Z' optimising compiler for the final testing and time trials. The sections of code to be timed were placed between two Fortran pause statements which temporarily halts the execution of the program. The operating systems accounting facility was the interrogated to display the CPU time taken to reach the present pause statement.

The difference in values is thus the CPU time used to execute the code between the two pause statements, the time is displayed in milliseconds. A run-time Fortran library extension allows a program to interrogate the system clock directly from the code, however the time is returned in seconds which is not sufficiently accurate to enable CPU time comparisons to be made between programs on any of the 4 test networks except the 118 substation test network.

The States of a power system are defined as the voltage magnitude and phase angle at every node in the network, the phase angles value being calculated with respect to an arbitrarily fixed value at one of the nodes. The nodal power injections and line flows may then be calculated from the state estimates and the network parameters. The majority of state estimation algorithms calculate values for the states of the network in an iterative process from all the available measurements of voltage magnitude, nodal injections and line flows. If estimates of the nodal injections or line flows are required, they are then calculated separately at the end of the estimation process. The 4 stage decomposed linear programming state estimation algorithm presented in the thesis differs from conventional methods in that the estimates of nodal injections and line flows are evaluated explicitly during the estimation process. To include the estimates of all the power flows in all the test cases mentioned would create long lists of numbers in which the advantages of the technique would be lost, thus only the voltage estimates are listed and compared to the results from other state estimation techniques where relevant. The detection and correction of any bad measurements or switch status information is mentioned in the text.

It was initially intended to compare the performance of the proposed state

estimation algorithm with the performance of two other algorithms. Both these algorithms are based on the Newton Raphson method, one of which uses the least squares method to solve the linearised network equations, the other the Revised Simplex method. The sub-routines used to implement the Revised Simplex method are identical to that used in the implementation of the proposed algorithm and the sub-routine implementing the least squares method is the same as the one used to evaluate the link power flows in the simulator program. Both these state estimators work at a nodal level and require a small topology program to form a list of which bus sections are connected together by active links and hence constitute an electrical node. The topology program also constructs a list of active transmission lines, i.e. those connected to 2 nodes. Thus any comparison of the voltage estimates between the different state estimation algorithms must be at a nodal level while comparisons between the proposed algorithm and the simulator may be at a bus-section level. The latter is generally not necessary as the voltage levels of the bus-sections forming an electrical node are identical and it is hence sufficient to highlight any voltage magnitude measurements which have been rejected.

Unfortunately the Newton Raphson state estimation algorithm using the linear programming technique to solve the equations would not converge on the 30 substation test network. The problem was traced to a combination of factors. The algorithm worked at a nodal level and hence several of the electrical nodes had two or more voltage magnitude measurements arising from measurements on different bus sections within that electrical node. This degeneracy led to the situation that the Simplex algorithm selected an error term of one of these degenerate voltage magnitude measurement equations to enter the basis, on the next iteration the method of selecting an error term to enter the basis chose an error term of one of the other degenerate equations while the method of selecting the error term to leave the basis chose the first degenerate equation. The Simplex algorithm thus ended up in a loop continually swapping the error terms of these equations in and out of the basis. This situation known as cycling arises partly by chance in that the coefficients of all the equations allow the process to fall into a region where the cost function has reached an intermediate minimum.

A similar problem is likely to occur in the nodal version of the 4 stage decomposed linear programming state estimation algorithm described in section 6.5, although to date it has not been observed. The degenerate problem is usually removed if the measurements are subjected to the addition of noise as the magnitude of the error terms for each of the degenerate equations is then

different and the methods of selecting which error terms are to enter and leave the basis will select alternative variables. This however is not a satisfactory way to solve the problem as it does not guarantee that the situation will not arise. Possible solution methods include abandoning the Simplex process after a given number of iterations and continuing with the next iteration in the state estimation algorithm, or averaging the degenerate measurements into one.

The former alternative has the advantage that it is easy to program but may affect the reliability of the estimates as the linear programming technique may not minimise the objective function as far as possible. The later alternative has the disadvantage that a corrupt measurement will, when averaged with the others, produce a value which is still corrupt, although to a lesser degree. An alternative is to only average the measurement values which are similar to form additional equations for those measurement values which differ from the average value. This alternative however would be difficult to implement. None of the above mentioned modifications have yet been implemented and thus the results from this stage estimation algorithm have not been included in the thesis.

To illustrate the improvement of the 4 stage decomposed linear state estimation program in comparison with the data validation program the estimates of the line flows etc. have been listed for the two cases originally described in section 4.1.2 and discussed section 6.3 which is entitled 'Results for the 5 substation test network'. However, before the discussion of the state estimate results, it is worth mentioning the measurement redundancy of the test networks.

6.2 Measurement redundancy

The measurement redundancy of a power system is a measure of the number of measurements which could be removed before the network as a whole becomes unobservable. The redundancy however does not specify which of the measurements may be removed before the network becomes unobservable. It maybe such that removing one measurement from a section of the network where there are few measurements will cause that section of the network to become unobservable, while a different section of the network may remain observable even if several measurements are removed.

The measurement redundancy may be defined in two ways. A network which is just observable may be defined as having a redundancy of zero, i.e. there are no redundant measurements or it may be defined as having a redundancy of one.

The latter definition may be regarded as a measure of the measurement density and is the definition adopted in thesis.

The measurement redundancy of the network is one of the outputs of the least squares state estimator and it is calculated as follows. The total number of voltage magnitude measurements (including all degenerate measurements) are summed together with the total number of active and reactive line flow measurements and the total number of active and reactive power flow injection measurements to produce the total number of measurements. It should be noted that the injection measurements refer to the total injection at the electrical nodes in the network. Thus if one of the generators or loads connected to a node is unmeasured, then the injection at that node is said to be unmeasured but if there are no generators or loads connected to a node then the value of the injection measurement is set to zero. The total number of measurements is then divided by 2n-1 where n is the number of nodes in the network. The voltage phase angle at the reference node is pre-defined and hence it does not have to be estimated, thus the number of voltage angle estimates is one less than the number of nodes. The number of injection measurements and the number of nodes in the network depends on the present topology and may alter upon a switch state change. The measurement redundancy for each of the networks is listed below, the values have been evaluated when each of the networks was in its initial steady state, that is with all the switches closed.

Test Network	Measurement redundancy
5 substation	4.111
30 substation	3.305
57 substation	2.487
118 substation	2.957

The above values are not strictly the redundancy levels for the 4 stage decomposed linear programming state estimator because this state estimator works at the bus section level and also considers the links. Furthermore, the linear programming method is able to use the power flow injection measurements on the individual loads and generators even if there are two such elements connected to the same bus station and one does not have a measurement available. The true measurement redundancy for the linear programming state estimator has not been evaluated as most readers will be familiar with the conventional method of evaluating the redundancy.

It would be expected that changing the measurement redundancy would change

both the solution times and the reliability of the estimates. A discussion of the effects of measurement redundancy on the performance of the 4 stage decomposed linear programming state estimator can be found in section 6.4.4.

6.3 Results for the 5 substation test network

The estimates from the 4 stage decomposed linear programming state estimation program on the 5 substation test network are listed in tables 6.1 and 6.2.

The figures in the column headed 'true values' represent the actual state of the network. These values are calculated by the simulator and are the values from which the measurement values are calculated where appropriate. Thus in the ideal state the estimates as calculated by the state estimation programs would be identical to the true values.

The results are produced from the same set of active power flow measurements as those used to produce the estimates of the active power flows from the data validation algorithm listed in tables 4.2 and 4.3. In the first case, the measurements reflect the true state of the network while in the second case the flow measurement on line 7 has been set to zero. The 4 stage development linear programming state estimation program is able to use its estimates of the voltage levels etc. to improve the estimates of the flows in substation 5 where the measurement redundancy is low. This low redundancy prevents the data validation algorithm from calculating good estimates or correctly identifying the bad measurement in the second example. The 4 stage decomposed linear programming state estimation program rejects the active flow measurement in line 7 but still has sufficient information available from making full use of all the other measurements to calculate accurate estimates of all the power flows and voltage levels throughout the network. The results of the Newton Raphson least squares state estimator using the measurement set with the active power flow measurement on line 7 set to zero are listed in table 6.3, a table of the results of this program using the measurement set with no errors has not been presented because the results are virtually identical to those of table 6.1. It should be noted that the least squares state estimator works at a nodal level and therefore does not produce estimates for the link power flows.

It can be seen from table 6.3 that the magnitude of the estimate errors is generally small but that the single error has affected the value of the majority of the power flows and voltage levels throughout the network. The least squares method is aided by the relatively high redundancy of the

Table 6.1: Estimates from the state estimation program on the 5 substation test network with no measurement errors

Values are in P.U.
(S) => sending end of a line, (R) => receiving end of line.
Error = Estimate - True value

Power flow estimates

Generators

Number		True value	Estimate	Error
1	Active	1.2956	1.2956	0.0000
1	Reactive	-0.0748	-0.0748	0.0000
2	Active	0.4000	0.4000	0.0000
2	Reactive	0.3000	0.3000	0.0000

Loads

Number		True value	Estimate	Error
1	Active	-0.2000	-0.1999	0.0001
1	Reactive	-0.1000	-0.0998	0.0002
2	Active	-0.4500	-0.4499	0.0001
2	Reactive	-0.1500	-0.1496	0.0004
3	Active	-0.4000	-0.4000	0.0000
3	Reactive	-0.0500	-0.0500	0.0000
4	Active	-0.6000	-0.6000	0.0000
4	Reactive	-0.1000	-0.1000	0.0000

Links

Number		True value	Estimate	Error
1	Active	0.6388	0.6386	-0.0002
1	Reactive	0.0980	0.0978	-0.0002
2	Active	0.2267	0.2267	0.0000
2	Reactive	-0.0169	-0.0169	0.0000
3	Active	0.2069	0.2069	0.0000
3	Reactive	0.0007	0.0006	-0.0001
4	Active	0.1939	0.1939	0.0000
4	Reactive	-0.0809	-0.0810	-0.0001
5	Active	-0.0198	-0.0198	0.0000
5	Reactive	0.0176	0.0176	0.0000
6	Active	-0.0328	-0.0328	0.0000
6	Reactive	-0.0640	-0.0640	0.0000
7	Active	-0.0129	-0.0129	0.0000
7	Reactive	-0.0816	-0.0816	0.0000

Lines

Number			True value	Estimate	Error
1	(S)	Active	-0.8884	-0.8885	-0.0001
1	(S)	Reactive	0.0862	0.0862	0.0000
1	(R)	Active	0.8743	0.8744	0.0001
1	(R)	Reactive	-0.0619	-0.0619	0.0000
2	(S)	Active	-0.4071	-0.4071	0.0000
2	(S)	Reactive	-0.0114	-0.0114	0.0000
2	(R)	Active	0.3952	0.3952	0.0000
2	(R)	Reactive	0.0300	0.0300	0.0000
3	(S)	Active	0.2434	0.2434	0.0000
3	(S)	Reactive	0.0678	0.0678	0.0000
3	(R)	Active	-0.2469	-0.2469	0.0000
3	(R)	Reactive	-0.0354	-0.0354	0.0000
4	(S)	Active	0.2749	0.2749	-0.0001
4	(S)	Reactive	0.0593	0.0591	-0.0001
4	(R)	Active	-0.2794	-0.2793	0.0001
4	(R)	Reactive	-0.0296	-0.0295	0.0001
5	(S)	Active	-0.5483	-0.5483	0.0001
5	(S)	Reactive	-0.0735	-0.0734	0.0002
5	(R)	Active	0.5371	0.5370	-0.0001
5	(R)	Reactive	0.0718	0.0716	-0.0002
6	(S)	Active	-0.1890	-0.1888	0.0002
6	(S)	Reactive	0.0518	0.0518	0.0000
6	(R)	Active	0.1886	0.1884	-0.0002
6	(R)	Reactive	-0.0319	-0.0319	-0.0001
7	(S)	Active	-0.0634	-0.0633	0.0001
7	(S)	Reactive	0.0228	0.0228	0.0000
7	(R)	Active	0.0631	0.0630	-0.0001
7	(R)	Reactive	0.0284	0.0284	0.0000

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0600	1.0600	0.0000
2	2	1.0242	1.0242	0.0000
3	2	1.0242	1.0242	0.0000
4	3	1.0236	1.0236	0.0000
5	4	1.0475	1.0475	0.0000
6	4	1.0475	1.0475	0.0000
7	4	1.0475	1.0475	0.0000
8	4	1.0475	1.0475	0.0000
9	5	1.0179	1.0180	0.0000

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0872	-0.0872	0.0000
3	2	-0.0872	-0.0872	0.0000
4	3	-0.0930	-0.0930	0.0000
5	4	-0.0490	-0.0490	0.0000
6	4	-0.0490	-0.0490	0.0000
7	4	-0.0490	-0.0490	0.0000
8	4	-0.0490	-0.0490	0.0000
9	5	-0.1073	-0.1073	0.0000

Table 6.2: Estimates from the state estimation program on the 5 substation test network with the active power flow measurement on line 7 set to zero

Values are in P.U.
(S) => sending end of a line, (R) => receiving end of line.
Error = Estimate - True value

Power flow estimates

Generators

Number		True value	Estimate	Error
1	Active	1.2956	1.2956	0.0000
1	Reactive	-0.0748	-0.0748	0.0000
2	Active	0.4000	0.4001	0.0001
2	Reactive	0.3000	0.3000	0.0000

Loads

Number		True value	Estimate	Error
1	Active	-0.2000	-0.1999	0.0001
1	Reactive	-0.1000	-0.0999	0.0001
2	Active	-0.4500	-0.4500	0.0000
2	Reactive	-0.1500	-0.1496	0.0004
3	Active	-0.4000	-0.4000	0.0000
3	Reactive	-0.0500	-0.0500	0.0000
4	Active	-0.6000	-0.5999	0.0001
4	Reactive	-0.1000	-0.1000	0.0000

Links

Number		True value	Estimate	Error
1	Active	0.6388	0.6386	-0.0002
1	Reactive	0.0980	0.0978	-0.0002
2	Active	0.2267	0.2267	0.0000
2	Reactive	-0.0169	-0.0170	0.0000
3	Active	0.2069	0.2069	0.0000
3	Reactive	0.0007	0.0007	0.0000
4	Active	0.1939	0.1939	0.0000
4	Reactive	-0.0809	-0.0810	-0.0001
5	Active	-0.0198	-0.0198	0.0000
5	Reactive	0.0176	0.0176	0.0000
6	Active	-0.0328	-0.0328	0.0000
6	Reactive	-0.0640	-0.0640	0.0000
7	Active	-0.0129	-0.0129	0.0000
7	Reactive	-0.0816	-0.0816	0.0000

Lines

Number			True value	Estimate	Error
1	(S)	Active	-0.8884	-0.8885	-0.0001
1	(S)	Reactive	0.0862	0.0862	0.0000
1	(R)	Active	0.8743	0.8744	0.0001
1	(R)	Reactive	-0.0619	-0.0619	0.0000
2	(S)	Active	-0.4071	-0.4071	0.0000
2	(S)	Reactive	-0.0114	-0.0114	0.0000
2	(R)	Active	0.3952	0.3952	0.0000
2	(R)	Reactive	0.0300	0.0300	0.0000
3	(S)	Active	0.2434	0.2434	0.0000
3	(S)	Reactive	0.0678	0.0678	0.0000
3	(R)	Active	-0.2469	-0.2469	0.0000
3	(R)	Reactive	-0.0354	-0.0354	0.0000
4	(S)	Active	0.2749	0.2749	-0.0001
4	(S)	Reactive	0.0593	0.0591	-0.0001
4	(R)	Active	-0.2794	-0.2793	0.0001
4	(R)	Reactive	-0.0296	-0.0295	0.0002
5	(S)	Active	-0.5483	-0.5483	0.0000
5	(S)	Reactive	-0.0735	-0.0733	0.0002
5	(R)	Active	0.5371	0.5371	0.0000
5	(R)	Reactive	0.0718	0.0716	-0.0002
6	(S)	Active	-0.1890	-0.1886	0.0003
6	(S)	Reactive	0.0518	0.0518	0.0000
6	(R)	Active	0.1886	0.1883	-0.0003
6	(R)	Reactive	-0.0319	-0.0319	-0.0001
7	(S)	Active	-0.0634	-0.0631	0.0003
7	(S)	Reactive	0.0228	0.0228	0.0000
7	(R)	Active	0.0631	0.0628	-0.0002
7	(R)	Reactive	0.0284	0.0284	0.0000

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0600	1.0600	0.0000
2	2	1.0242	1.0242	0.0000
4	3	1.0236	1.0236	0.0000
5	4	1.0475	1.0475	0.0000
9	5	1.0179	1.0180	0.0000

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0872	-0.0872	0.0000
4	3	-0.0930	-0.0930	0.0000
5	4	-0.0490	-0.0490	0.0000
9	5	-0.1073	-0.1073	0.0001

Table 6.3: Estimates from the Newton-Raphson least squares state estimator on the 5 substation test network with the active power flow measurement on line 7 set to zero

Values are in P.U.
(S) => sending end of a line, (R) => receiving end of line.
Error = Estimate - True value

Power flow estimates

Generators

Number		True value	Estimate	Error
1	Active	1.2956	1.2962	0.0005
1	Reactive	-0.0748	-0.0746	0.0002
2	Active	0.4000	0.4005	0.0005
2	Reactive	0.3000	0.3004	0.0004

Loads

Number		True value	Estimate	Error
1	Active	-0.2000	-0.2002	-0.0002
1	Reactive	-0.1000	-0.1001	-0.0001
2	Active	-0.4500	-0.4547	-0.0047
2	Reactive	-0.1500	-0.1500	0.0000
3	Active	-0.4000	-0.4081	-0.0081
3	Reactive	-0.0500	-0.0503	-0.0003
4	Active	-0.6000	-0.5879	0.0121
4	Reactive	-0.1000	-0.1000	0.0000

Links
Link power flow estimates are not calculated by the least squares state estimator.

Lines

Number			True value	Estimate	Error
1	(S)	Active	-0.8884	-0.8872	0.0013
1	(S)	Reactive	0.0862	0.0864	0.0001
1	(R)	Active	0.8743	0.8731	-0.0012
1	(R)	Reactive	-0.0619	-0.0619	0.0000
2	(S)	Active	-0.4071	-0.4090	-0.0019
2	(S)	Reactive	-0.0114	-0.0118	-0.0003
2	(R)	Active	0.3952	0.3970	0.0018
2	(R)	Reactive	0.0300	0.0300	0.0000
3	(S)	Active	0.2434	0.2462	0.0028
3	(S)	Reactive	0.0678	0.0679	0.0001
3	(R)	Active	-0.2469	-0.2498	-0.0029
3	(R)	Reactive	-0.0354	-0.0358	-0.0003
4	(S)	Active	0.2749	0.2777	0.0028
4	(S)	Reactive	0.0593	0.0593	0.0000
4	(R)	Active	-0.2794	-0.2822	-0.0029
4	(R)	Reactive	-0.0296	-0.0299	-0.0003
5	(S)	Active	-0.5483	-0.5413	0.0070
5	(S)	Reactive	-0.0735	-0.0727	0.0008
5	(R)	Active	0.5371	0.5303	-0.0068
5	(R)	Reactive	0.0718	0.0718	0.0001
6	(S)	Active	-0.1890	-0.1885	0.0004
6	(S)	Reactive	0.0518	0.0521	0.0003
6	(R)	Active	0.1886	0.1882	-0.0004
6	(R)	Reactive	-0.0319	-0.0322	-0.0003
7	(S)	Active	-0.0634	-0.0578	0.0056
7	(S)	Reactive	0.0228	0.0232	0.0004
7	(R)	Active	0.0631	0.0576	-0.0055
7	(R)	Reactive	0.0284	0.0282	-0.0002

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0600	1.0600	0.0000
2	2	1.0242	1.0241	-0.0002
4	3	1.0236	1.0235	-0.0001
5	4	1.0475	1.0475	0.0001
9	5	1.0179	1.0183	0.0004

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0872	-0.0876	-0.0004
4	3	-0.0930	-0.0934	-0.0004
5	4	-0.0490	-0.0489	0.0001
9	5	-0.1073	-0.1065	0.0009

measurement set on the network, the value for the 5 substation test network being 4.111.

The high redundancy means that the single measurement error when processed with the large number of valid measurements does not seriously corrupt the estimates. However a reduction in the level of the redundancy would cause a corresponding rise in the degree of the errors produced by the least squares state estimator. The linear programming state estimator is not sensitive to the measurement redundancy in the same way, a discussion on this subject can be found in section 6.4.4.

Typical solution times for the least squares state estimator and the linear programming state estimator are 0.36 seconds and 3.7 seconds respectively. These times were obtained from a flat start in which all the estimates were initialised to zero, except the voltage magnitude estimates which were initialised to 1.0. The solution times when starting from a previous valid set of estimates are typically 0.23 seconds and 2.9 seconds for the least squares and linear programming programs respectively. A more detailed assessment of the solution times has been obtained by the 30 substation test network, the results of which have been presented in the next section.

6.4 Results for the 30 substation test network

6.4.1 Solution times

A comparison of the solution times between the 4 stage decomposed linear programming and the least squares state estimation programs has been presented in table 6.4. The table compares the solution times for a sequence of events using two measurement sets, one which reflects the true network state, the second in which the measurements have been perturbed by the addition of a systematic and a random noise component. Further details on the method of calculating the perturbed measurements can be found at the end of chapter 3. The extent to which the addition of a systematic and random noise component alters the measurement values is illustrated in appendix 5. The appendix compares two sets of measurement values for the 30 substation network in its initial steady state. The first column of measurement values have had no gross errors or noise added to them and the second column of measurement values have been subject to the addition of 0.2% systematic noise component and a 1.5 random noise component. The difference between the true measurement value and the value after the addition of the noise components is listed in the final

Table 6.4: Comparison between the solution times of the 4 stage linear programming and the Newton-Raphson least squares state estimation programs

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	7.34 (2)	72.37 (5)
Force to run	0.14 (0)	12.00 (1)
Open line 7	7.62 (2)	30.71 (3)
Open link 6	4.01 (1)	37.39 (3)
Close all switches	7.21 (2)	41.16 (4)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	8.84 (3)	74.08 (4)
Force to run	0.47 (1)	4.70 (1)
Open line 7	8.17 (3)	33.70 (4)
Open link 6	7.74 (3)	26.10 (3)
Close all switches	8.69 (3)	30.89 (4)
Measurement update	7.37 (2)	28.21 (3)

column. The values chosen for the magnitude of the noise components reflects the levels noise to be expected from a telemetry system and are typical of the values used by other researchers[44,84] working in the field of state estimation. The two measurement sets listed in appendix 5 are those used at the start of the sequence of events listed in table 6.4.

In order to ensure that both programs used identical values for the perturbed measurements the values are stored in a file and copied into the measurement common block when required. This procedure of storing the values of the perturbed measurements in a file also applies to the other 5 events listed in the second half of table 6.4. It should be noted that it is possible to reproduce the values by running the simulator program again as the random number generator uses a constant as an initial seed and hence produces the same series of random numbers each time the program is run. However, care has to be taken to ensure that the simulator undergoes the same number of time steps in each case. The measurement values for the other 5 cases have not been included in the thesis. The final point to note on the measurement values is that the set of measurement values for the case entitled measurement update at the bottom of table 6.4 were produced by allowing the simulator to proceed one time step from the previous point, thus the random component of each of the measurements has changed which in turn alters the final values of the measurements. When the measurements are subject to the addition of no noise then running the simulator does not change the values of the measurements under unless the system has been disturbed by a change in the total load and is hence in an oscillatory state.

The solution times in table 6.4 clearly illustrate the large amount of CPU time used by the linear programming state estimator on the first solution run. As has been explained in chapter 5 this extra time arises because of the poor initial basis tableau. On subsequent runs, both state estimators generally have a fairly constant solution time, although as would be expected this time depends upon the extent of the change in the operating conditions of the network. The least squares state estimator has a subroutine which compares the present measurement set with the previous measurement set. If the change in all the values is less than a specified tolerance then the estimation process is not required and the estimates are not updated. This feature explains the very fast solution time for the least squares state estimator in the second step of the examples where the measurements are subject to no noise. At this step the operator has forced the program to run. However the measurements have not changed since the last run and the estimation process is not required. It would appear that the tolerance specified for the above test

on the change in the measurement values is too tight because when the same test is performed in the second half of table 6.4 it fails. The likely cause of this failure is numerical round-off in the machine coupled with a tolerance that is too small. The 32-bit words used by the machine generally only give a reliability of 6 significant figures depending on the number of calculations and memory updates performed on the variables.

The least squares method has two places at which a check for convergence is made. Firstly after the evaluation of the Jacobian elements, if all the elements are below a specified tolerance then the process has convered because solving the Jacobian matrix equation would produce no significant change in the values of the estimates. Secondly if after solving the Jacobian matrix equation the change in the values of all the estimates is below a specified tolerance then the process is said to have converged. Thus although the least squares method fails its test on the change of the measurement values in the second step of the example with measurements subject to the addition of noise, the values of the resulting Jacobian elements are all small; hence the process terminates without solving the Jacobian matrix equation. This explained the relatively fast solution time for this example.

The 4 stage linear programming state estimation program only has one point at which a test for convergence is made. This point is after the solution of all 4 sub-estimation stages, further details on the implementation of the program can be found in chapter 5. Thus the 4 stage decomposed linear programming state estimation program always undergoes 1 full iteration which accounts for the high solution times in the examples where the measurement values have not changed. Notwithstanding this difference in the methods for testing for convergence the linear programming method is 3 to 4 times slower than the least squares method when starting from a previos valid solution point. However, the linear programming method shows an improvement in solution times when running in the more realistic conditions of measurements subject to noise fluctuations, the least squares method on the other hand returns faster solution times when in the ideal situation of noise free measurements.

The reduction in the solution times for the linear programming method arises from a reduction in the solution time in each of the 4 sub-estimation proglems, since for a given step in the examples in table 6.4, the same number of main iterations are required whether the measurements are subject to the addition noise or not. The reduction in solution time of the linear problem arises because the addition of noise to the measurments reduces the number of non-basic variables eligible to enter the basis at any given Simplex

iteration.

When the measurements are noise free, there exists at any point in time, a number of adjacent measurement equations which may be combined with the network constraint equations such as the bus-section power flow sum check equations to define the estimates at a given point in the network. If the measurements are subject to the addition of noise then only one of the measurements will give the greatest reduction in the cost function. The addition of noise to the measurement not only reduces the number of non-basic variables eligible to enter the basis at any time but lends itself to forming a definite solution point. At the solution point, a basis exchange of any of the error term variables associated with the measurements would result in an increase in the value of the cost function. In the case of noise free measurements, only numerical round-off in the machine dictates whether a basis exchange of the measurement equation error term variables results in an approvement in the value of the cost function. In this situation it is more difficult to determine when convergence has been achieved and hence more Simplex iterations are usually required.

The situation is reversed for the least squares state estimator because the least squares algorithm considers all the linear equations formed from the Jacobian matrix and hence the addition of noise to the measurements causes contention between the equations, which in practice requires slightly more least squares iterations to resolve. Furthermore an additional main iteration is usually required to reduce the magnitude of the Jacobian elements and the magnitude of the change in the values of the estimates to within the specified tolerance. This arises because the least squares method smooths the values of the estimates defined by each of the linear equations of the Jacobian matrix equation; thus none of the equations are ever exactly satisfied at the solution point.

6.4.2 Estimates calculated from measurements free from gross errors

The voltage magnitude and voltage phase angle estimates calculated by the 4 stage decomposed state estimation program have been listed in table 6.5. These estimates were produced from a measurement set which reflects the network in its initial steady state and were not subject to the addition of any noise or gross erros. The correlation between the true values and the estimated values is excellent, the same degree of correlation extends to both the power flow injection estimates and also the line power flow estimates. The estimates calculated by the least squares state estimator show a similar

Table 6.5: Estimates from the state estimation program on the 30 substation test network with no measurement errors

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0438	0.0000
2	2	1.0301	1.0301	0.0000
4	3	1.0105	1.0105	0.0000
5	4	1.0025	1.0025	0.0000
7	5	0.9851	0.9851	0.0000
10	6	0.9951	0.9951	0.0000
16	7	0.9829	0.9829	0.0000
17	8	0.9894	0.9894	0.0000
25	9	1.0071	1.0071	0.0000
26	10	0.9928	0.9928	0.0000
36	11	1.0486	1.0486	0.0000
37	12	0.9889	0.9889	0.0000
39	13	1.0011	1.0011	0.0000
40	14	0.9752	0.9752	0.0000
41	15	0.9721	0.9721	0.0000
45	16	0.9828	0.9828	0.0000
46	17	0.9841	0.9841	0.0000
47	18	0.9664	0.9664	0.0000
48	19	0.9664	0.9664	0.0000
49	20	0.9722	0.9722	0.0000
50	21	0.9790	0.9790	0.0000
51	22	0.9793	0.9793	0.0000
52	23	0.9646	0.9646	0.0000
53	24	0.9637	0.9637	0.0000
55	25	0.9602	0.9602	0.0000
56	26	0.9415	0.9415	0.0000
57	27	0.9673	0.9673	0.0000
63	28	0.9899	0.9899	0.0000
64	29	0.9462	0.9462	0.0000
69	30	0.9340	0.9340	0.0000

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0375	0.0000
4	3	-0.0728	-0.0728	0.0000
5	4	-0.0870	-0.0870	0.0000
7	5	-0.1310	-0.1310	0.0000
10	6	-0.1028	-0.1028	0.0000
16	7	-0.1241	-0.1241	0.0000
17	8	-0.1067	-0.1067	0.0000
25	9	-0.1317	-0.1317	0.0000
26	10	-0.1690	-0.1690	0.0000
36	11	-0.0922	-0.0923	0.0000
37	12	-0.1482	-0.1482	0.0000
39	13	-0.1199	-0.1199	0.0000
40	14	-0.1666	-0.1666	0.0000
41	15	-0.1698	-0.1698	0.0000
45	16	-0.1626	-0.1626	0.0000
46	17	-0.1712	-0.1712	0.0000
47	18	-0.1832	-0.1832	0.0000
48	19	-0.1874	-0.1874	0.0000
49	20	-0.1839	-0.1839	0.0000
50	21	-0.1780	-0.1780	0.0000
51	22	-0.1778	-0.1778	0.0000
52	23	-0.1798	-0.1798	0.0000
53	24	-0.1864	-0.1864	0.0000
55	25	-0.1855	-0.1855	0.0000
56	26	-0.1938	-0.1938	0.0000
57	27	-0.1799	-0.1799	0.0000
63	28	-0.1111	-0.1111	0.0000
64	29	-0.2040	-0.2040	0.0000
69	30	-0.2213	-0.2213	0.0000

degree of correlation and hence have not been listed in a tabular form. The table provides a reference standard with which the correlation between the true values and the estimates produced using a different set of measurement values may be compared. It should be noted that unless otherwise stated, all the tables of estimates have been produced by the state estimation programs from a flat start.

The following two tables, that is tables 6.6 and 6.7, were produced by the linear programming and least squares state estimators respectively, using a measurement set subject to the addition of 0.2% systematic noise and 1.5% random noise. The measurement set still however reflects the network in its initial steady state and the measurement values used can be found in appendix 5. As can be seen from the tables, the magnitude of the errors on the estimates calculated by the least squares state estimator are approximately half those from the linear programming state estimator. This is to be expected as the least squares state estimator produces estimates calculated by smoothing all of the measurements while the linear programming state estimator produces estimates based on a sub-set of the measurements. The sub-set of measurements may be regarded as the least noisy sub-set of measurements taken from the entire set of measurements, subject to the constraint that each of the four sub-estimation processes is still observable.

The above example illustrates that although the linear programming state estimator is not as adept at smoothing the random noise on the measurements, it is still able to produce acceptable results under these conditions.

6.4.3 Estimates calculated from measurements subject to gross errors

The papers published by Irving et al. on substation data validation[66] and state estimation[65] using linear programming have shown that their method performs well in the presence of gross measurement errors and incorrect switch status information. It would be expected therefore that as the 4 stage decomposed linear programming state estimation program is based on the data validation program that it too would perform well under these conditions. The following three examples illustrate that this is indeed the case and highlights the failings of the least squares method under these conditions. As with the perturbed measurement values, a set of gross measurement errors was adopted as a standard and used in all the test cases. As explained in detail in chapter 3, a small separate program allows the user to interact with the simulator in order to define which measurement points will have a gross error applied to them and the degree to which the measurement values are corrupted.

Table 6.6: Estimates from the state estimation program on the 30 substation test network with 0.2% systematic noise and 1.5% random noise

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0375	-0.0063
2	2	1.0301	1.0239	-0.0062
4	3	1.0105	1.0041	-0.0064
5	4	1.0025	0.9961	-0.0065
7	5	0.9851	0.9786	-0.0064
10	6	0.9951	0.9886	-0.0065
16	7	0.9829	0.9763	-0.0066
17	8	0.9894	0.9829	-0.0065
25	9	1.0071	1.0008	-0.0063
26	10	0.9928	0.9863	-0.0065
36	11	1.0486	1.0430	-0.0056
37	12	0.9889	0.9824	-0.0065
39	13	1.0011	0.9947	-0.0064
40	14	0.9752	0.9684	-0.0067
41	15	0.9721	0.9654	-0.0067
45	16	0.9828	0.9762	-0.0066
46	17	0.9841	0.9776	-0.0065
47	18	0.9664	0.9596	-0.0068
48	19	0.9664	0.9595	-0.0069
49	20	0.9722	0.9653	-0.0069
50	21	0.9790	0.9723	-0.0067
51	22	0.9793	0.9726	-0.0067
52	23	0.9646	0.9579	-0.0068
53	24	0.9637	0.9568	-0.0069
55	25	0.9602	0.9534	-0.0068
56	26	0.9415	0.9342	-0.0073
57	27	0.9673	0.9607	-0.0066
63	28	0.9899	0.9833	-0.0066
64	29	0.9462	0.9395	-0.0067
69	30	0.9340	0.9274	-0.0066

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0373	0.0002
4	3	-0.0728	-0.0741	-0.0013
5	4	-0.0870	-0.0886	-0.0016
7	5	-0.1310	-0.1330	-0.0020
10	6	-0.1028	-0.1045	-0.0016
16	7	-0.1241	-0.1261	-0.0020
17	8	-0.1067	-0.1084	-0.0017
25	9	-0.1317	-0.1334	-0.0018
26	10	-0.1690	-0.1724	-0.0034
36	11	-0.0922	-0.0941	-0.0018
37	12	-0.1482	-0.1511	-0.0029
39	13	-0.1199	-0.1225	-0.0026
40	14	-0.1666	-0.1698	-0.0033
41	15	-0.1698	-0.1731	-0.0033
45	16	-0.1626	-0.1657	-0.0031
46	17	-0.1712	-0.1744	-0.0032
47	18	-0.1832	-0.1864	-0.0032
48	19	-0.1874	-0.1909	-0.0035
49	20	-0.1839	-0.1874	-0.0035
50	21	-0.1780	-0.1815	-0.0035
51	22	-0.1778	-0.1813	-0.0035
52	23	-0.1798	-0.1832	-0.0034
53	24	-0.1864	-0.1901	-0.0037
55	25	-0.1855	-0.1893	-0.0037
56	26	-0.1938	-0.1982	-0.0044
57	27	-0.1799	-0.1834	-0.0035
63	28	-0.1111	-0.1129	-0.0018
64	29	-0.2040	-0.2076	-0.0037
69	30	-0.2213	-0.2250	-0.0037

Table 6.7: Estimates from the Newton-Raphson least squares state estimator on the 30 substation test network with 0.2% systematic noise and 1.5% random noise

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0408	-0.0030
2	2	1.0301	1.0271	-0.0030
4	3	1.0105	1.0075	-0.0030
5	4	1.0025	0.9995	-0.0031
7	5	0.9851	0.9819	-0.0031
10	6	0.9951	0.9920	-0.0030
16	7	0.9829	0.9798	-0.0030
17	8	0.9894	0.9863	-0.0031
25	9	1.0071	1.0042	-0.0028
26	10	0.9928	0.9898	-0.0030
36	11	1.0486	1.0462	-0.0024
37	12	0.9889	0.9860	-0.0030
39	13	1.0011	0.9981	-0.0029
40	14	0.9752	0.9721	-0.0031
41	15	0.9721	0.9691	-0.0030
45	16	0.9828	0.9798	-0.0030
46	17	0.9841	0.9812	-0.0029
47	18	0.9664	0.9635	-0.0030
48	19	0.9664	0.9634	-0.0031
49	20	0.9722	0.9691	-0.0031
50	21	0.9790	0.9759	-0.0030
51	22	0.9793	0.9763	-0.0030
52	23	0.9646	0.9618	-0.0029
53	24	0.9637	0.9610	-0.0027
55	25	0.9602	0.9574	-0.0028
56	26	0.9415	0.9383	-0.0032
57	27	0.9673	0.9645	-0.0028
63	28	0.9899	0.9868	-0.0031
64	29	0.9462	0.9434	-0.0027
69	30	0.9340	0.9312	-0.0028

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0375	0.0000
4	3	-0.0728	-0.0735	-0.0006
5	4	-0.0870	-0.0878	-0.0008
7	5	-0.1310	-0.1326	-0.0015
10	6	-0.1028	-0.1036	-0.0008
16	7	-0.1241	-0.1251	-0.0010
17	8	-0.1067	-0.1076	-0.0009
25	9	-0.1317	-0.1326	-0.0009
26	10	-0.1690	-0.1697	-0.0007
36	11	-0.0922	-0.0928	-0.0005
37	12	-0.1482	-0.1496	-0.0013
39	13	-0.1199	-0.1211	-0.0012
40	14	-0.1666	-0.1682	-0.0016
41	15	-0.1698	-0.1712	-0.0015
45	16	-0.1626	-0.1638	-0.0011
46	17	-0.1712	-0.1721	-0.0009
47	18	-0.1832	-0.1844	-0.0012
48	19	-0.1874	-0.1886	-0.0012
49	20	-0.1839	-0.1850	-0.0011
50	21	-0.1780	-0.1788	-0.0008
51	22	-0.1778	-0.1787	-0.0008
52	23	-0.1798	-0.1813	-0.0014
53	24	-0.1864	-0.1877	-0.0013
55	25	-0.1855	-0.1875	-0.0020
56	26	-0.1938	-0.1966	-0.0028
57	27	-0.1799	-0.1818	-0.0019
63	28	-0.1111	-0.1120	-0.0009
64	29	-0.2040	-0.2062	-0.0022
69	30	-0.2213	-0.2237	-0.0023

The following 8 gross measurement errors where chosen at random as the standard set for the 30 substation test network.

Meas. type	Element number	Sub. number	Multi. factor	True value	Meas. value
V	bus 5	4	0.6	1.0025	0.6015
V	bus 61	27	0.0	0.9673	0.0
P	gen.1	1	0.0	1.1469	0.0
P	line 5	2	-1.0	-0.5101	0.5101
P	line -32	24	0.5	0.0195	0.0098
Q	load 10	12	1.3	-0.0750	-0.0975
Q	line 5	2	0.0	-0.1130	0.0
Q	line 38	27	1.2	-0.0170	-0.0204

A - sign indicates the receiving end of a line
The true value reflects the network in its initial steady state.

Table 6.8 presents the results for the 4 stage decomposed linear programming state estimator using a measurement set corrupted by the application of the above multiplication factors. The initial measurement values reflected the system in its steady state starting condition and were not subject to the addition of any noise components. As can be seen from the results, the gross measurement errors have not affected the estimates to any significant extent. All of the active and reactive power flow estimates agreed with the true power flow values and the gross measurement errors were rejected. A small error exists on the voltage magnitude estimate at bus section 39. However, the magnitude of the error is unlikely to be of concern to a power system operator as it is less than half of one percent. Bus section 39 is in substation 13 and when all the switches in the network are closed, this bus section is also node 13. Substation 13 is a single bus section with one generator and one line (which represents a transformer) connected to it. The other end of the line connects to substation 12 at which there is an incorrect reading on the reactive power flow of load 10. The small error on the voltage magnitude estimate arises because of the following points: Substation 13 is isolated and hence there is no additional information available to verify the estimates, the natural laws governing power flows in a power system means that the voltage magnitude estimates depend to a significant extent on the reactive power flow estimates. The reactive power flow estimates agree with the true values to within 4 decimal places. However, the sensitivity of the network is such that a small error in

Table 6.8: Estimates from the state estimation program on the 30 substation test network with 8 severely corrupted analogue measurements

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0438	0.0000
2	2	1.0301	1.0301	0.0000
4	3	1.0105	1.0105	0.0000
5	4	1.0025	1.0025	0.0000
7	5	0.9851	0.9851	0.0000
10	6	0.9951	0.9951	0.0000
16	7	0.9829	0.9829	0.0000
17	8	0.9894	0.9894	0.0000
25	9	1.0071	1.0071	0.0000
26	10	0.9928	0.9928	0.0000
36	11	1.0486	1.0486	0.0000
37	12	0.9889	0.9889	0.0000
39	13	1.0011	1.0043	0.0032
40	14	0.9752	0.9752	0.0000
41	15	0.9721	0.9721	0.0000
45	16	0.9828	0.9828	0.0000
46	17	0.9841	0.9841	0.0000
47	18	0.9664	0.9664	0.0000
48	19	0.9664	0.9664	0.0000
49	20	0.9722	0.9722	0.0000
50	21	0.9790	0.9790	0.0000
51	22	0.9793	0.9793	0.0000
52	23	0.9646	0.9646	0.0000
53	24	0.9637	0.9637	0.0000
55	25	0.9602	0.9602	0.0000
56	26	0.9415	0.9415	0.0000
57	27	0.9673	0.9673	0.0000
63	28	0.9899	0.9899	0.0000
64	29	0.9462	0.9462	0.0000
69	30	0.9340	0.9340	0.0000

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0375	0.0000
4	3	-0.0728	-0.0728	0.0000
5	4	-0.0870	-0.0870	0.0000
7	5	-0.1310	-0.1310	0.0000
10	6	-0.1029	-0.1029	0.0000
16	7	-0.1241	-0.1241	0.0000
17	8	-0.1067	-0.1067	0.0000
25	9	-0.1317	-0.1317	0.0000
26	10	-0.1690	-0.1690	0.0000
36	11	-0.0923	-0.0923	0.0000
37	12	-0.1483	-0.1483	0.0000
39	13	-0.1200	-0.1201	-0.0001
40	14	-0.1666	-0.1666	0.0000
41	15	-0.1698	-0.1698	0.0000
45	16	-0.1627	-0.1627	0.0000
46	17	-0.1712	-0.1712	0.0000
47	18	-0.1832	-0.1832	0.0000
48	19	-0.1874	-0.1874	0.0000
49	20	-0.1840	-0.1840	0.0000
50	21	-0.1780	-0.1780	0.0000
51	22	-0.1779	-0.1779	0.0000
52	23	-0.1799	-0.1799	0.0000
53	24	-0.1864	-0.1864	0.0000
55	25	-0.1856	-0.1856	0.0000
56	26	-0.1938	-0.1938	0.0000
57	27	-0.1799	-0.1799	0.0000
63	28	-0.1111	-0.1111	0.0000
64	29	-0.2040	-0.2040	0.0000
69	30	-0.2214	-0.2214	0.0000

the reactive power estimates manifests itself in a larger error on the voltage magnitude estimates.

Using the same measurement set as above, the least squares state estimator failed to converge in 4 iterations. A Newton Raphson iterative process is generally likely to be diverging if a solution has not been obtained within a few iterations, hence a limit of 4 iterations has been applied to the program to prevent numerical overflow occurring within the computer which would result in the operating system aborting the program. The output of the least squares state estimator is suppressed if the program fails to converge, thus no estimates have been presented in a tabular form.

The second and third examples illustrate the ability of the 4 stage decomposed linear programming state estimator to identify incorrect switch status measurements. As with the analogue measurements, two standard sets of corrupt switch status measurements were chosen at random and used throughout all the tests. In the first set of corrupt switch status measurements only the switch at the sending end of transmission line 5, which is connected to substations two and 5, was corrupted. In the second set all the transmission lines connected to substation two have had the switch status measurements corrupted, that is the receiving end of line 1 and the sending ends of lines 3, 5 and 6. In both cases, the analogue measurement values have had no errors or noise components added to them and reflected the network in its steady state starting condition. All the switches were initially closed and hence the corrupted switch status measurements indicate that the switches were open. This means that in the second case substation 2, according to the switch status measurements, is isolated from the rest of the network.

The estimates from the 4 stage decomposed linear programming state estimator have not been listed in tabular form for the case where the status of the switch at the sending end of line 5 has been corrupted because they are identical to those of table 6.5. The program has rejected the incorrect switch status measurement and the estimates of the power flows in line 5 were also correctly calculated.

The program has a small routine at the end of the main estimation process which performs a logical check on the switch status measurements and the power flow estimates throughout the network. If a power system element has an open switch element but the estimate of either the active or reactive power flow is not zero, then the switch status measurement is deemed to be incorrect. The converse however is not true as the element may have no power flowing through it but may still be connected to the system. The routine will indicate that a possible error may exist on a line or link switch measurement if the switch

measurement is closed but there is no power flow and the voltage levels across the element differ by a significant amount. This routine could be improved upon if it also considered the error terms of the linear equations associated with each of the elements where a possible switch status measurement error exists. This enhancement has not been implemented due to a shortage of time but a further discussion on its implications can be found in the concluding chapter, chapter 9. This logical check on the switch status measurements and the power flow estimates enabled the program to identify the corrupt switch measurement and in this case, to correct the measurement.

The estimates from the least squares state estimator for the above case have been listed in table 6.9. As can be seen, the estimates for the nodes adjacent to the corrupt switch measurement have some errors on them which are as large as 0.11 per unit for the voltage estimates and 0.027 per unit for the voltage phase angle estimates. The effect is fairly local for the voltage magnitude estimates and does not propogate through the rest of the network. However, the voltage phase angle estimates have been calculated with respect to the node which contains the reference bus-section. The reference bus-section has been set to bus-section 1 which will have been assigned node number 1. Studying the overall layout of the 30 substation test (figure 3.10) reveals that there are only two transmission lines from substation 1 which contains bus-section 1. Thus any error in the estimates of the voltage phase angle in the substations connected directly to substation 1 is likely to propagate through a large portion of the network. The first of the lines from substation 1 is connected to substation two which also has line 5 connected to it. The switch status measurement error on line 5 causes an error in the estimation of the voltage phase angles at the terminating bus-sections which is then propagated through the rest of the network. As can be seen from table 6.9 themagnitude of the error on the voltage phase angle estimates from bus-station 26 (substation 10) onwards is fairly constant. If the reference bus-section had been a long way from line 5, for example bus-section 73 then the effect would not have been propagated through such a large portion of the network. It must be pointed out that the topology program which defines the number of nodes and active lines in the network for the least squares state estimator has flagged line 5 as being inactive and hence the estimates of the power flows were automatically set to zero. The least squares estimator then has to reconcile the discrepancies in adjacent power flow measurements which results in errors up to 0.3 per unit in the power flow estimates for the adjacent lines, loads and generators.

The estimates from the least squares state estimator when 4 switch status

Table 6.9: Estimates from the Newton-Raphson least squares state estimator on the 30 substation test network with 1 line switch status error

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0487	0.0049
2	2	1.0301	1.0382	0.0081
4	3	1.0105	1.0111	0.0006
5	4	1.0025	1.0030	0.0004
7	5	0.9851	0.9735	-0.0116
10	6	0.9951	0.9951	0.0000
16	7	0.9829	0.9808	-0.0021
17	8	0.9894	0.9894	0.0000
25	9	1.0071	1.0068	-0.0003
26	10	0.9928	0.9924	-0.0004
36	11	1.0486	1.0484	-0.0002
37	12	0.9889	0.9888	-0.0001
39	13	1.0011	1.0009	-0.0001
40	14	0.9752	0.9747	-0.0005
41	15	0.9721	0.9717	-0.0004
45	16	0.9828	0.9824	-0.0004
46	17	0.9841	0.9837	-0.0004
47	18	0.9665	0.9660	-0.0004
48	19	0.9665	0.9660	-0.0004
49	20	0.9722	0.9718	-0.0004
50	21	0.9790	0.9786	-0.0004
51	22	0.9793	0.9789	-0.0004
52	23	0.9646	0.9642	-0.0004
53	24	0.9637	0.9633	-0.0004
55	25	0.9602	0.9599	-0.0003
56	26	0.9415	0.9411	-0.0004
57	27	0.9673	0.9671	-0.0002
63	28	0.9899	0.9898	-0.0001
64	29	0.9462	0.9459	-0.0003
69	30	0.9340	0.9336	-0.0003

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0322	0.0054
4	3	-0.0728	-0.0800	-0.0072
5	4	-0.0870	-0.0944	-0.0073
7	5	-0.1310	-0.1584	-0.0274
10	6	-0.1029	-0.1110	-0.0082
16	7	-0.1241	-0.1363	-0.0122
17	8	-0.1067	-0.1152	-0.0084
25	9	-0.1317	-0.1435	-0.0118
26	10	-0.1690	-0.1817	-0.0126
36	11	-0.0923	-0.1050	-0.0127
37	12	-0.1483	-0.1604	-0.0122
39	13	-0.1200	-0.1326	-0.0127
40	14	-0.1666	-0.1794	-0.0128
41	15	-0.1698	-0.1824	-0.0126
45	16	-0.1627	-0.1753	-0.0127
46	17	-0.1712	-0.1839	-0.0127
47	18	-0.1833	-0.1960	-0.0127
48	19	-0.1874	-0.2001	-0.0127
49	20	-0.1840	-0.1967	-0.0127
50	21	-0.1780	-0.1906	-0.0126
51	22	-0.1779	-0.1905	-0.0126
52	23	-0.1799	-0.1925	-0.0126
53	24	-0.1864	-0.1989	-0.0125
55	25	-0.1856	-0.1969	-0.0113
56	26	-0.1938	-0.2051	-0.0113
57	27	-0.1799	-0.1906	-0.0106
63	28	-0.1111	-0.1197	-0.0086
64	29	-0.2040	-0.2149	-0.0108
69	30	-0.2214	-0.2322	-0.0109

measure4ment errors were applied to the measurement set have been listed in table 6.10. The first point to note is that the topology program has set the number of nodes to 29 because the switch status measurements indicate that substation two is completely isolated from the network. The correlation between the bus number listed in column 1 of table 6.10 and the node number listed in column two is now invalid because the program which produces the tables was not written to allow for an error in the number of nodes. The program reads the bus-section number and the corresponding true voltage values from a list which is produced by the simulator. The simulator uses a valid model of the network and hence produces a list of true values for the 30 nodes. For simplicity the program producing the table uses the same value for the number of nodes as the least squares state estimator and compares the estimates with the relevant number of true values from the top of the list. The topology program produces a list of the bus-sections in each of the nodes and the bus-sections which are isolated are assigned a fictitious node number of zero. A more sophisticated display program would be able to compare the correct true values with the estimates, however this program would be complex to program and the benefit gained minimal. This means that in reality, because substation two has been omitted from the state estimator, the estimates ought to be compared with the true value for the following node except for node 1 where the topology error has had no effect. This reduces the magnitude of the errors for the voltage magnitude estimates and the voltage phase angle estimates to the order of 0.005 per unit and 0.017 per unit respectively. The errors in the voltage estimates gives rise to errors in the estimates of the line flows as high as 0.4 per unit on some lines and of course the flow estimates for lines 1,3,5 and 6 were set to zero. The majority of the power flow injection imbalance arising from the missing power flow injections at substation two was accounted for by suitably altering the injection estimates at substation 3.

The estimates from the 4 stage decomposed linear programming state estimator produced from a measurement set with 4 corrupt switch status measurements have been listed in table 6.11. As in the previous case where only 1 switch measurement was corrupted, the program has been able to identify and correct the invalid switch status measurements. The table further illustrates how an incorrect voltage phase angle estimate near the reference node tends to propagate through a large portion of the network, all of the nodes apart from node 1 have a small error on the estimate which is constant throughout the majority of the network. However, the estimates are more than accurate enough for use by a power system operator or a second analysis

Table 6.10: Estimates from the Newton-Raphson least squares state estimator on the 30 substation test network with 4 line switch status errors

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0541	0.0103
2	2	1.0301	1.0119	-0.0183
4	3	1.0105	1.0022	-0.0083
5	4	1.0025	0.9766	-0.0260
7	5	0.9851	0.9950	0.0099
10	6	0.9951	0.9830	-0.0120
16	7	0.9829	0.9903	0.0074
17	8	0.9894	1.0114	0.0220
25	9	1.0071	0.9983	-0.0088
26	10	0.9928	1.0534	0.0606
36	11	1.0486	0.9925	-0.0560
37	12	0.9889	1.0051	0.0162
39	13	1.0011	0.9801	-0.0209
40	14	0.9752	0.9769	0.0018
41	15	0.9721	0.9881	0.0160
45	16	0.9828	0.9897	0.0069
46	17	0.9841	0.9718	-0.0123
47	18	0.9664	0.9720	0.0055
48	19	0.9664	0.9777	0.0113
49	20	0.9722	0.9845	0.0123
50	21	0.9790	0.9848	0.0058
51	22	0.9793	0.9698	-0.0095
52	23	0.9646	0.9689	0.0043
53	24	0.9637	0.9646	0.0009
55	25	0.9602	0.9464	-0.0138
56	26	0.9415	0.9705	0.0290
57	27	0.9673	0.9914	0.0241
63	28	0.9899	0.9506	-0.0392
64	29	0.9462	0.9386	-0.0076

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.1080	-0.0705
4	3	-0.0728	-0.1263	-0.0534
5	4	-0.0870	-0.1834	-0.0964
7	5	-0.1310	-0.1423	-0.0113
10	6	-0.1029	-0.1631	-0.0602
16	7	-0.1241	-0.1442	-0.0201
17	8	-0.1067	-0.1492	-0.0425
25	9	-0.1317	-0.1821	-0.0504
26	10	-0.1690	-0.1034	0.0656
36	11	-0.0923	-0.1688	-0.0766
37	12	-0.1483	-0.1394	0.0089
39	13	-0.1200	-0.1853	-0.0653
40	14	-0.1666	-0.1883	-0.0217
41	15	-0.1698	-0.1787	-0.0089
45	16	-0.1627	-0.1850	-0.0223
46	17	-0.1712	-0.1995	-0.0283
47	18	-0.1832	-0.2022	-0.0190
48	19	-0.1874	-0.1983	-0.0109
49	20	-0.1840	-0.1913	-0.0073
50	21	-0.1780	-0.1912	-0.0132
51	22	-0.1779	-0.1970	-0.0191
52	23	-0.1799	-0.2016	-0.0217
53	24	-0.1864	-0.2055	-0.0191
55	25	-0.1856	-0.2134	-0.0278
56	26	-0.1938	-0.2033	-0.0095
57	27	-0.1799	-0.1474	0.0325
63	28	-0.1111	-0.2258	-0.1147
64	29	-0.2040	-0.2428	-0.0388

Table 6.11: Estimates from the state estimation program on the 30 substation test network with 4 line switch status errors

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0438	0.0000
2	2	1.0301	1.0301	0.0000
4	3	1.0105	1.0105	0.0000
5	4	1.0025	1.0025	0.0000
7	5	0.9851	0.9851	0.0000
10	6	0.9951	0.9951	0.0000
16	7	0.9829	0.9829	0.0000
17	8	0.9894	0.9894	0.0000
25	9	1.0071	1.0071	0.0000
26	10	0.9928	0.9928	0.0000
36	11	1.0486	1.0486	0.0000
37	12	0.9889	0.9889	0.0000
39	13	1.0011	1.0011	0.0000
40	14	0.9752	0.9752	0.0000
41	15	0.9721	0.9721	0.0000
45	16	0.9828	0.9828	0.0000
46	17	0.9841	0.9841	0.0000
47	18	0.9664	0.9664	0.0000
48	19	0.9664	0.9665	0.0000
49	20	0.9722	0.9722	0.0000
50	21	0.9790	0.9790	0.0000
51	22	0.9793	0.9793	0.0000
52	23	0.9646	0.9646	0.0000
53	24	0.9637	0.9637	0.0000
55	25	0.9602	0.9602	0.0000
56	26	0.9415	0.9415	0.0000
57	27	0.9673	0.9673	0.0000
63	28	0.9899	0.9899	0.0000
64	29	0.9462	0.9462	0.0000
69	30	0.9340	0.9340	0.0000

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0377	-0.0002
4	3	-0.0728	-0.0731	-0.0002
5	4	-0.0870	-0.0873	-0.0002
7	5	-0.1310	-0.1316	-0.0005
10	6	-0.1029	-0.1031	-0.0002
16	7	-0.1241	-0.1247	-0.0005
17	8	-0.1067	-0.1069	-0.0002
25	9	-0.1317	-0.1319	-0.0002
26	10	-0.1690	-0.1692	-0.0002
36	11	-0.0923	-0.0925	-0.0002
37	12	-0.1483	-0.1485	-0.0002
39	13	-0.1200	-0.1202	-0.0002
40	14	-0.1666	-0.1668	-0.0002
41	15	-0.1698	-0.1700	-0.0002
45	16	-0.1627	-0.1629	-0.0002
46	17	-0.1712	-0.1715	-0.0002
47	18	-0.1832	-0.1835	-0.0002
48	19	-0.1874	-0.1876	-0.0002
49	20	-0.1840	-0.1842	-0.0002
50	21	-0.1780	-0.1782	-0.0002
51	22	-0.1779	-0.1781	-0.0002
52	23	-0.1799	-0.1801	-0.0002
53	24	-0.1864	-0.1866	-0.0002
55	25	-0.1856	-0.1858	-0.0002
56	26	-0.1938	-0.1940	-0.0002
57	27	-0.1799	-0.1801	-0.0002
63	28	-0.1111	-0.1113	-0.0002
64	29	-0.2040	-0.2042	-0.0002
69	30	-0.2214	-0.2216	-0.0002

program such as security assessment. The active power flow estimates for the lines, loads and generators were all correct but the reactive power estimates for lines 1,6 and 9 had a small error, the magnitude of which was 0.025 per unit. A few of the adjacent lines had even smaller errors, (magnitude in the order of 0.005 per unit) but none of the reactive power flow estimates for the loads or generators were incorrect. The exact cause of the errors is not clear but it is likely to be caused by terminating the solution too soon. As explained in chapter 5 the termination of the iterative process is controlled by the change in the total cost of the solution from one iteration to the next. The change in the total cost correlates well with the change in any of the estimates but it is not necessarily true that all of the estimates have remained static. It is likely that given a further iteration the reactive power flow estimates would have been correct. This arises because on the first iteration the estimator had to consider the implications of the corrupt switch status measurements, however on subsequent iterations the effects of these errors is not so great and the program progresses towards the correct solution.

The above examples only serve to illustrate the potential advantages of using the 4 stage decomposed linear programming state estimator. Further tests have been performed but the results have not been presented in tabular form because they would not further enlighten the reader to the ability of the technique to calculate reliable estimates in the presence of gross measurement errors and incorrect switch status information. However, a discussion of some of the features encountered during the testing of the program has been presented below.

6.4.4 The effects of measurement redundancy

It would be expected that the level of measurement redundancy would significantly affect the performance of a state estimator. A detailed study of the performance with respect to the measurement redundancy has not been possible but the following points are worth mentioning.

The accuracy and reliability of the estimates is bound to increase as as the measurement redundancy increases. The noise filtering ability of the technique will increase as the measurement redundancy increases because the program has a greater number of measurements from which to select the least noisy set of measurements required to define each of the sub-estimation problems.

The requirement of the program to detect a bad measurement is that an

alternative equation exists to define the estimate corresponding to the measurement equation which is not already defining the value of an estimate at any other point in the power system. Furthermore a suitable path exists to allow the Simplex algorithm to perform a basis exchange for the error term components of each of the equations. When the measurement redundancy is low a large number of the non-measurement equations will be required to define the estimates for those points in the network without a measurement. In this situation measurement errors are likely to go undetected and it is likely that a minimum threshold value for the measurement redundancy exists below which little detection of bad measurements will occur.

Above this first threshold value the program will be able to detect single bad measurement values, a second threshold value will exist which would enable the program to detect two adjacent bad measurement values. Naturally the ability of the program to detect bad measurements will also depend on the structure of the network and the distribution of the measurements throughout the network.

The level of the measurement redundancy generally affects the overal solution times of a state estimation program, since a reduction in the redundancy level reduces the number of linear equations and hence the overall problem size. However the implementation of the 4 stage decomposed linear programming state estimator is such that the measurement redundancy will have no significant effect on the solution times. The reason for this is as follows: the implementation of the linear programming method, the Revised Simplex method, is such that it requires the user to supply a sub-set of the linear equations from which an initial feasible basic tableau can be formed. The 4 stage decomposed state estimation program does this by assuming that every point in the network has a measurement available. The correct measurement value is assigned to the points that have a measurement available and the value of zero to those points which do not have a measurement available. Thus unless there are numerous points with two or more measurements available then changing the overall measurement redundancy will not change the number of linear equations and hence will not change the solution times.

If the 4 stage decomposed linear programming method was implemented in such a manner that did not necessitate the use of the dummy measurement equations then the overall measurement redundancy would affect the solution times. A further discussion of this point can be found in the concluding chapter, chapter 9 where the evaluation of an attempt to remove the 'dummy' measurement equations is discussed.

The effect of the measurement redundancy on the Newton Raphson least squares state estimator has not been investigated but it is known that both the least squares and the linear programming routines exhibit a quadratic relationship between the solution times and the problem size.

6.4.5 Other features of the of the 4 stage decomposed state estimator

The ability of the 4 stage decomposed linear programming state estimator to detect bad measurements and incorrect switch status measurements is not seriously affected by the addition of noise to the measurements. It has been observed that under certain conditions whereby a switch status measurement is corrupted to indicate open instead of closed and the values of the true active and reactive power flowing through the switch are very low then a high level of noise on the measurements may cause the switch error to do undetected. The failure of the program arises because the noise level is such that the power flow measurements and hence the power flow estimates are below the tolerance required for a power system element to be deemed active. In these situations the estimates of the voltage levels will not be seriously affected as the true power flows are so small in any case and in a subsequent estimation run the noise may change such that the power flow estimate is above the tolerance and the switch status error will be detected.

The 4 stage decomposed linear programming state estimator is not as efficient at detecting measurement errors associated with the generators and loads. This is to be expected when the sub-estimation problems for the estimation of the active and reactive power flows are examined more closely. In addition to the measurement equation the power flow estimate may also be evaluated from the power flow sum check equation at the bus-section to which the power system element is connected. This is true for any element in the network, however the links are connected to two bus-sections and hence the estimate could be evaluated from either of the power sum check equations associated with the two bus-sections. Similarly the estimate of the power flow at the end of a line could also be evaluated from the power flow sum check equation or the equation relating the power flow estimate to the estimate of the power flow at the other end of the line. Thus in the case of a link or a line, there is a higher level of information available from which to evaluate the estimate of the power flow than in the case of a generator or load. The problem is exagerated if the bus-section has an unmeasured generator or load connected to it. In this case the only equation available

from which to evaluate the power flow estimate is the power flow sum check equation at the relevant bus-section. If there is a measurement error adacent to the generator or load then there is a chance that the measurement may be accepted and the estimate for the power flow in the unmeasured generator or load adjusted to compensate for the incorrect measurement. It should be noted that should this occur then only two or three estimates in the local vicinity will be wrong while in the case of the least squares state estimator the error could be smeared over a larger number of estimates.

During the testing of the program it was noticed that if a line switch measurement was corrupted so that it indicated open instead of closed and both the real and reactive power flow measurements for that end of the line were corrupted so that the values equalled zero then it was possible that the measurement on an adjacent generator or load would be rejected and switch switch status measurement accepted as being valid. The estimate of the power flow in the generator or load would be set to agree with the zero line flow estimate. This problem was minimised by increasing the weights on the power flow measurement equations for the generators and loads above those of the remaining measurement equations, typical values thus being 0.8 for the measurement equations associated with the generators and loads and 0.6 for the link and line power flow measurement equations.

6.5 The 4 stage nodal decomposed linear programming state estimator

The solution time of a state estimation program is almost as important as the accuracy of the calculated estimates. The state estimation program is likely to be providing estimates of the state of the network for both the power system operator and also other network analysis programs such as a continency analysis program. When the power system is operating under abnormal conditions the operator will require precise and up-to-date information about the state of the network. Unfortunately under these conditions the state of the network will be changing rapidly and the previous state estimate solution will not be such a good starting point for the next evaluation of the estimates as when the network is operating under normal conditions. The solution time is therefore going to be greater in these situations and it is important that the state estimator is able to converge to the solution in as short a time as possible.

The solution times for the 4 stage linear programming state estimator have been presented in table 6.4 along with those for the least squares state estimator. The table illustrates that in the majority of cases the linear

programming state estimator is over 4 times as slow as the least squares state estimator. However it must be pointed out that since the 4 stage linear programming method considers the individual bus-sections in the system as opposed to the electrical nodes then the linear programming method is calculating a greater number of estimates. The solution times presented in table 6.4 have been obtained when all the switches were closed, thus the number of electrical nodes is equal to the number of substations in the network, that is 30. The least squares program thus calculates voltage magnitude and phase angles for 30 nodes whereas the linear programming program calculates estimates for the 73 bus-sections. To obtain a more accurate comparison of the solution times both methods will need to be calculating the same number of estimates.

It is not possible for the Newton Raphson least squares state estimator to operate at the bus-section level because the algorithm is unable to include the zero impedance links which inter-connect the bus-sections. The reason for this being that the equations defining the power flow between 2 points in terms of the voltage differences, are invalid if the impedance is zero (see chapter 3 for details on the equations). Thus in order to obtain a more accurate comparison between the solution times a version of the 4 stage linear programming state estimator was written which operated at a nodal level.

This version of the linear programming method did not consider the links in the network and thus required the topology program to evaluate the number of nodes in the network and form a list of the nodal inter-connections of the transmission lines. The program did not therefore produce estimates of the link power flows and like the least squares method was also susceptible to switch status measurement errors. A discussion on the performance of the 4 stage nodal decomposed linear programming state estimation program now follows.

Since this nodal version of the program does not consider the links in the network each of the 4 sub-estimation problems is smaller and thus less time is required to solve the overall problem. This is reflected in the solution times listed in table 6.12 in which the solution times of the 4 stage nodal decomposed linear programming state estimator are compared with those of the conventional Newton Raphson least squares state estimator. The conditions under which the programs were run were indentical to those used to produce the timing results of the original program. These results have been presented in table 6.4 and thus a comparison may be made between the last columns of these tables to evaluate the effect on the solution times on removing the link equations from each of the 4 sub-estimation stages.

Table 6.12: Comparison between the solution times of the 4 stage nodal linear programming and the Newton-Raphson least squares state estimation programs

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	Nodal 4 Stage Linear programming
Start estimator	7.34 (2)	29.35 (5)
Force to run	0.14 (0)	6.09 (1)
Open line 7	7.62 (2)	19.93 (4)
Open link 6	4.01 (1)	17.44 (3)
Close all switches	7.21 (2)	19.41 (4)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	Nodal 4 Stage Linear programming
Start estimator	8.84 (3)	26.07 (4)
Force to run	0.47 (1)	1.54 (1)
Open line 7	8.17 (3)	13.19 (3)
Open link 6	7.74 (3)	9.93 (3)
Close all switches	8.69 (3)	12.25 (3)
Measurement update	7.37 (2)	10.65 (3)

In the majority of cases the nodal version of the linear programming state estimation program is twice as fast as the original program, however as table 6.12 illustrates the 4 stage nodal decomposed state estimator is still considerably slower than the Newton Raphson least squares method. Working on measurement sets which have not been perturbed by the addition of any noise the linear programming method is over twice as slow as the least squares method. However upon the addition of the random and systematic noise components the linear programming method only requires 50% additional CPU time to obtain a solution as compared with the least squares method. This improvement in the solution times when the measurements are subject to the addition of noise arises for the same reasons as the similar improvement obtained from the original program discussed earlier.

The estimates produced by the 4-stage nodal decomposed linear programming state estimator are generally the same as those produced by the original version of the program, thus tables of the results for this version have not been included except for those obtained when the measurement set was subject to the 4 line switch status measurement errors in which there is a significant difference. The estimates produced using this measurement set have been presented in table 6.13.

As has already been explained this version of the program, like the Newton Raphson least squares method is susceptable to switch status measurement errors. However in the case when only line 1 has a switch status measurement error the program is able to correctly calculate voltage estimates for all the nodes and power flow estimates for all the generators, loads and lines. The reason for this is because the 4 stage nodal decomposed linear programming state estimator does not rely on the list of active transmission produced by the topology program and the error has not lead to the formation of an incorrect nodal inter-connection list. In the case when there are 4 line switch status measurement errors the program only calculates estimates for 29 nodes instead of the correct number which is 30. The mis-alignment of the estimates and the true values caused by the program which producted the tables can be more clearly seen in this example. The reason for this mis-alignment has been explained earlier in the chapter. If the estimates for node number 2 onwards are compared with the true value for the following node then the estimates are in fact correct. Allowing for this mis-alignment the least squares state estimate still calculated estimates with errors in the order of 0.005 per unit. Since the 4 stage nodal linear programming state estimator does not rely on the topology program for the status of the transmission lines it was able to calculate estimates for the power flows in lines 1,3,5 and 6

Table 6.13: Estimates from the nodal version of the state estimation program on the 30 substation test network with 4 line switch status errors

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0438	0.0000
2	2	1.0301	1.0105	-0.0196
4	3	1.0105	1.0025	-0.0080
5	4	1.0025	0.9851	-0.0175
7	5	0.9851	0.9951	0.0100
10	6	0.9951	0.9829	-0.0122
16	7	0.9829	0.9894	0.0065
17	8	0.9894	1.0071	0.0177
25	9	1.0071	0.9928	-0.0143
26	10	0.9928	1.0486	0.0558
36	11	1.0486	0.9889	-0.0597
37	12	0.9889	1.0011	0.0122
39	13	1.0011	0.9752	-0.0259
40	14	0.9752	0.9721	-0.0031
41	15	0.9721	0.9828	0.0107
45	16	0.9828	0.9841	0.0013
46	17	0.9841	0.9664	-0.0177
47	18	0.9664	0.9664	0.0000
48	19	0.9664	0.9722	0.0057
49	20	0.9722	0.9790	0.0068
50	21	0.9790	0.9793	0.0003
51	22	0.9793	0.9646	-0.0147
52	23	0.9646	0.9637	-0.0010
53	24	0.9637	0.9602	-0.0034
55	25	0.9602	0.9415	-0.0188
56	26	0.9415	0.9673	0.0258
57	27	0.9673	0.9899	0.0226
63	28	0.9899	0.9462	-0.0437
64	29	0.9462	0.9340	-0.0122

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0728	-0.0353
4	3	-0.0728	-0.0870	-0.0142
5	4	-0.0870	-0.1310	-0.0440
7	5	-0.1310	-0.1029	0.0282
10	6	-0.1029	-0.1241	-0.0213
16	7	-0.1241	-0.1067	0.0174
17	8	-0.1067	-0.1317	-0.0250
25	9	-0.1317	-0.1690	-0.0373
26	10	-0.1690	-0.0923	0.0768
36	11	-0.0923	-0.1483	-0.0560
37	12	-0.1483	-0.1200	0.0283
39	13	-0.1200	-0.1666	-0.0466
40	14	-0.1666	-0.1698	-0.0032
41	15	-0.1698	-0.1627	0.0072
45	16	-0.1627	-0.1712	-0.0086
46	17	-0.1712	-0.1832	-0.0120
47	18	-0.1832	-0.1874	-0.0042
48	19	-0.1874	-0.1840	0.0035
49	20	-0.1840	-0.1780	0.0060
50	21	-0.1780	-0.1779	0.0001
51	22	-0.1779	-0.1799	-0.0020
52	23	-0.1799	-0.1864	-0.0065
53	24	-0.1864	-0.1856	0.0009
55	25	-0.1856	-0.1938	-0.0082
56	26	-0.1938	-0.1799	0.0139
57	27	-0.1799	-0.1111	0.0688
63	28	-0.1111	-0.2040	-0.0929
64	29	-0.2040	-0.2214	-0.0174

which contain the invalid switch status measurements. However, the generators and loads connected to substation 2 have been deemed to be inactive since the topology program has assigned them to an isolated node with a node number of zero. These false assumptions imposed on the estimator led to the failure of program to correctly estimate the reactive power flows in the 4 lines. The reactive power flow in line 1 was set to zero and the error on the remaining 3 lines was 0.0097 per unit. As a result of this a few of the surrounding lines had errors in the order of 0.0005 per unit. The active power flow estimates for all the lines were correct , the overall power flow balance was maintained because the implementation of the program was such that the set of linear equations contained a nodal sum check equation for node zero. Thus in effect the original node 2 had merely been renumbered to node zero, however if other generators, loads or lines had been assigned a node number of zero due to switch status measurement errors then further errors in the estimates would be likely.

The nodal 4 stage decomposed linear programming state estimator is thus considerably quicker than the original method which considers the inter-connections between the bus-sections, i.e. the links, however the gain in solution times is at the expense of the ability to detect and correct multiple switch status measurement errors. As stated by Johnson, Potts, Wrubel and Schulte[69] switch status measurement errors are the worst major problem for state estimators, thus the gain in solution times may prove to be worthless in a realistic on-line environment.

6.6 The 4 stage nodal decomposed least squares state estimator

The estimation of the states of a power system using the Newton Raphson method involves the formation of a set of over-determined linear equations from the non-linear system equations by taking the partial differentials of the non-linear equations. A choise arises as to the method of solving the set of linear equations. The majority of state estimation programs currently in use today[4,9,41,63,94,103] use a least squares method of one form or another although several authors have considered using linear programming methods.[9,44,50,65,66,76,77]

The choice therefore arises as to the method of solving the set of linear equations in each of the 4 sub-estimation stages of the 4 stage decomposed state estimation algorithm. Since the method was developed from the substation data validation algorithm[66] which was a linear programming technique the first choice was to use the same linear programming subroutines.

However the set of linear equations formed by the 4 stage decomposed state estimation program were readily adaptable for solution by the least squares state estimator. Thus the 4 stage nodal decomposed state estimator described in section 6.4.1 was modified to use the least squares subroutines. This program was called the 4 stage decomposed least squares state estimator.

The linear equations were weighted using the method described in chapter 5 to reflect the weightings used in the original method. Thus the nodal sum check equations were given a higher weighting than the line difference equations and the measurement equations were given the lowest weighting of all. Considering the solution times obtained from the Newton Raphson least squares state estimator it was expected that the 4 stage nodal decomposed least squares state estimator would have faster solutions times than the equivalent linear programming method and show an improvement in the ability to smooth noisy measurements whilst still retaining its ability to reject bad data. However, as the following discussion outlines, the results obtained were generally disappointing.

The solution times for the 4 stage nodal decomposed least squares state estimator have been listed in table 6.14 alongside those of the conventional Newton Raphson least squares state estimator. A direct comparison may be made between the two columns of solution times since both programs work at the nodal level and are hence solving problems of equivalent sizes. Similarly a direct comparison may be made between the solution times for the 4 stage nodal decomposed least squares state estimator in table 6.14 and those of the 4 stage nodal decomposed linear programming state estimator listed in table 6.12. A comparison of these times illustrates the factor solution times of the least squares method as compared with the linear programming method on a similar set of linear equations.

The 4 stage decomposed least squares state estimator, as expected, showed an improvement in the values of the estimates when working on a set of measurements subject to the addition of noise. The estimates produced by the program when the measurements are subject to the addition of no noise or errors were identical to those of table 6.5 and therefore have not been listed separately. The estimates produced by the program on the set of measurements subject to the addition of 0.2% systematic noise and 1.5% random noise have been listed in table 6.15. The magnitude of the errors in the voltage estimates are similar to those of the conventional Newton Raphson least squares state estimator which have been listed in table 6.7. The magnitude of the errors in the voltage estimates produced by the 4 stage decomposed linear programming state estimator are approximately double those of the least

Table 6.14: Comparison between the solution times of the 4 stage nodal least squares and the Newton-Raphson least squares state estimation programs

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Least squares
Start estimator	7.34 (2)	18.88 (4)
Force to run	0.14 (0)	0.48 (1)
Open line 7	7.62 (2)	12.57 (3)
Open link 6	4.01 (1)	4.60 (3)
Close all switches	7.21 (2)	16.98 (4)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Least squares
Start estimator	8.84 (3)	18.41 (4)
Force to run	0.47 (1)	0.49 (1)
Open line 7	8.17 (3)	12.23 (3)
Open link 6	7.74 (3)	7.02 (3)
Close all switches	8.69 (3)	16.42 (4)
Measurement update	7.37 (2)	4.28 (3)

Table 6.15: Estimates from the nodal least squares version of the state estimation algorithm on the 30 substation test network with 0.2% systematic noise and 1.5% random noise

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0416	-0.0022
2	2	1.0301	1.0275	-0.0027
4	3	1.0105	1.0077	-0.0028
5	4	1.0025	0.9991	-0.0034
7	5	0.9851	0.9837	-0.0014
10	6	0.9951	0.9926	-0.0025
16	7	0.9829	0.9809	-0.0020
17	8	0.9894	0.9840	-0.0054
25	9	1.0071	1.0042	-0.0028
26	10	0.9928	0.9894	-0.0034
36	11	1.0486	1.0465	-0.0021
37	12	0.9889	0.9856	-0.0033
39	13	1.0011	0.9975	-0.0036
40	14	0.9752	0.9718	-0.0034
41	15	0.9721	0.9689	-0.0032
45	16	0.9828	0.9794	-0.0034
46	17	0.9841	0.9808	-0.0033
47	18	0.9664	0.9632	-0.0032
48	19	0.9664	0.9632	-0.0033
49	20	0.9722	0.9689	-0.0033
50	21	0.9790	0.9756	-0.0033
51	22	0.9793	0.9761	-0.0031
52	23	0.9646	0.9617	-0.0029
53	24	0.9637	0.9610	-0.0027
55	25	0.9602	0.9575	-0.0027
56	26	0.9415	0.9381	-0.0033
57	27	0.9673	0.9647	-0.0026
63	28	0.9899	0.9864	-0.0035
64	29	0.9462	0.9436	-0.0026
69	30	0.9340	0.9315	-0.0025

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0377	-0.0002
4	3	-0.0728	-0.0735	-0.0006
5	4	-0.0870	-0.0876	-0.0006
7	5	-0.1310	-0.1324	-0.0014
10	6	-0.1029	-0.1035	-0.0007
16	7	-0.1241	-0.1251	-0.0010
17	8	-0.1067	-0.1074	-0.0007
25	9	-0.1317	-0.1321	-0.0004
26	10	-0.1690	-0.1699	-0.0009
36	11	-0.0923	-0.0924	-0.0001
37	12	-0.1483	-0.1496	-0.0013
39	13	-0.1200	-0.1211	-0.0011
40	14	-0.1666	-0.1681	-0.0015
41	15	-0.1698	-0.1713	-0.0014
45	16	-0.1627	-0.1639	-0.0013
46	17	-0.1712	-0.1724	-0.0011
47	18	-0.1832	-0.1846	-0.0014
48	19	-0.1874	-0.1889	-0.0014
49	20	-0.1840	-0.1851	-0.0011
50	21	-0.1780	-0.1790	-0.0010
51	22	-0.1779	-0.1789	-0.0011
52	23	-0.1799	-0.1813	-0.0015
53	24	-0.1864	-0.1880	-0.0016
55	25	-0.1856	-0.1874	-0.0018
56	26	-0.1938	-0.1962	-0.0024
57	27	-0.1799	-0.1816	-0.0017
63	28	-0.1111	-0.1118	-0.0007
64	29	-0.2040	-0.2060	-0.0020
69	30	-0.2214	-0.2235	-0.0021

squares methods. However, the magnitude of the errors produced by the linear programming method is only half of one percent of the true value for the voltage magnitude estimates and 2 percent for the voltage phase angle estimates.

The advantages of the gains in the solution times and the performance in the presence of noisy measurements of the 4 stage decomposed least squares state estimator are outweighted by the disadvantages of the performance in the presence of gross measurement errors and corrupt switch status measurements. The estimates produced by the 4-stage nodal decomposed least squares state estimator from a measurement set with 8 corrupted values have been listed in table 6.16. It should be noted that the conventional Newton Raphson least squares method failed to converge on this measurement set. However, the estimates produced from the program are subject to large errors which are above 0.075 per unit for the voltage magnitude estimates and in excess of 0.050 per unit for the majority of the voltage phase angle estimates. The values of the 8 corrupted measurements together with the true value and the estimated value have been listed below.

Measurement type	Element number	Measurement value	True value	Estimated value
V	bus 5	0.6015	1.0025	0.9560
V	bus 61	0.0	0.9673	0.8902
P	gen 1	0.0	1.1469	0.5766
P	line 5	0.5101	-0.5101	-0.0429
P	line -32	0.0098	0.0195	0.0133
Q	load 10	-0.0975	0.0750	-0.0912
Q	line 5	0.0	-0.1130	-0.0488
Q	line 38	-0.0204	-0.0170	-0.0164

A - sign indicates the receiving end of a line

The values are in per unit

The program has been able to produce good estimates for the two corrupt voltage magnitude measurements. However, the estimates for the corrupt power flow measurements are generally half way between the corrupt measurement value and the true value. It should also be noted that several other of the line flow estimates had errors as large as 0.4 per unit. Thus the least squares implementation of the 4 stage decomposed state estimation algorithm is unable to reject the bad measurements and smears the error arising from the corrupt

Table 6.16: Estimates from the nodal least squares version of the state estimation algorithm on the 30 substation test network with 8 severely corrupted measurements

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0008	-0.0430
2	2	1.0301	0.9942	-0.0359
4	3	1.0105	0.9696	-0.0409
5	4	1.0025	0.9560	-0.0465
7	5	0.9851	0.9708	-0.0142
10	6	0.9951	0.9608	-0.0342
16	7	0.9829	0.9592	-0.0236
17	8	0.9894	0.9569	-0.0324
25	9	1.0071	0.9722	-0.0348
26	10	0.9928	0.9535	-0.0393
36	11	1.0486	1.0187	-0.0299
37	12	0.9889	0.9451	-0.0438
39	13	1.0011	0.9622	-0.0389
40	14	0.9752	0.9312	-0.0439
41	15	0.9721	0.9279	-0.0442
45	16	0.9828	0.9406	-0.0422
46	17	0.9841	0.9433	-0.0408
47	18	0.9664	0.9232	-0.0433
48	19	0.9664	0.9243	-0.0422
49	20	0.9722	0.9312	-0.0409
50	21	0.9790	0.9380	-0.0410
51	22	0.9793	0.9371	-0.0422
52	23	0.9646	0.9189	-0.0457
53	24	0.9637	0.9173	-0.0464
55	25	0.9602	0.8982	-0.0620
56	26	0.9415	0.8784	-0.0631
57	27	0.9673	0.8902	-0.0771
63	28	0.9899	0.9424	-0.0475
64	29	0.9462	0.8679	-0.0783
69	30	0.9340	0.8546	-0.0794

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0165	0.0210
4	3	-0.0728	-0.0429	0.0299
5	4	-0.0870	-0.0619	0.0252
7	5	-0.1310	-0.0538	0.0772
10	6	-0.1029	-0.0721	0.0308
16	7	-0.1241	-0.0689	0.0552
17	8	-0.1067	-0.0761	0.0306
25	9	-0.1317	-0.1046	0.0271
26	10	-0.1690	-0.1462	0.0228
36	11	-0.0923	-0.0627	0.0295
37	12	-0.1483	-0.1262	0.0221
39	13	-0.1200	-0.0953	0.0247
40	14	-0.1666	-0.1462	0.0204
41	15	-0.1698	-0.1497	0.0202
45	16	-0.1627	-0.1412	0.0215
46	17	-0.1712	-0.1496	0.0217
47	18	-0.1832	-0.1639	0.0194
48	19	-0.1874	-0.1678	0.0196
49	20	-0.1840	-0.1633	0.0207
50	21	-0.1780	-0.1562	0.0219
51	22	-0.1779	-0.1562	0.0217
52	23	-0.1799	-0.1604	0.0194
53	24	-0.1864	-0.1661	0.0204
55	25	-0.1856	-0.1652	0.0203
56	26	-0.1938	-0.1748	0.0190
57	27	-0.1799	-0.1589	0.0211
63	28	-0.1111	-0.0806	0.0305
64	29	-0.2040	-0.1877	0.0163
69	30	-0.2214	-0.2084	0.0130

value over the surrounding estimates. Attempts to improve the ability of the method to reject the bad measurements by altering the relative weighting of the measurement equations did not yield any significant change in the performance of the program.

The situation was similar in the case of switch status measurement errors, although the magnitude of the errors on the estimates was well within acceptable limits. The voltage estimates produced by the 4 stage nodal decomposed least squares state estimator when the status of the switch at the sending end of line 5 was corrupted from closed to open have been listed in table 6.17. As expected the estimates of the voltages at the terminating nodes of line 5 (nodes 2 and 5) have suffered the worst errors and the effect has spread out through the rest of the estimates. The values of the power flow estimates through line 5 were below the true values but again were within acceptable limits. Although the program produced acceptable estimates in the presence of 1 switch status measurement error, the program failed to converge when the 4 line switch status measurements at substation 2 were corrupted. The failure of the algorithm to converge arises from the fact that too many line flow estimates were initially very low. The low line flow estimates were caused by the equations defining the power flows through the open breakers as zero and as a result of this the initial voltage estimates were wrong and the program is unable to converge the estimates to realistic values.

The least squares implementation of the 4 stage decomposed state estimation algorithm thus has no real advantages over the conventional Newton Raphson least squares state estimation method or the linear programming implementation of the 4 stage decomposed state estimation method. However, a combination of the linear programming implementation and the least squares implementation of the 4 stage decomposed state estimation algorithm could result in a state estimation method with an improved ability to reject bad measurements and smooth noisy measurements. A further discussion of this idea can be found in the concluding chapter, chapter 9.

The least squares implementation would have one significant advantage over the linear programming implementation when applied to the original program in which the link flow estimates are calculated. This advantage is discussed in section 6.8.

6.7 Results from the 57 and 118 substation test networks

The reliability of the estimates calculated by all of the state estimation programs on the 57 and 118 substation test networks were generally the same as

Table 6.17: Estimates from the nodal least squares version of the state estimation algorithm on the 30 substation test network with 1 line switch status error

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0459	0.0021
2	2	1.0301	1.0347	0.0045
4	3	1.0105	1.0110	0.0005
5	4	1.0025	1.0024	-0.0001
7	5	0.9851	0.9828	-0.0023
10	6	0.9951	0.9949	-0.0001
16	7	0.9829	0.9827	-0.0002
17	8	0.9894	0.9892	-0.0002
25	9	1.0071	1.0063	-0.0008
26	10	0.9928	0.9912	-0.0016
36	11	1.0486	1.0484	-0.0002
37	12	0.9889	0.9882	-0.0007
39	13	1.0011	1.0004	-0.0007
40	14	0.9752	0.9742	-0.0010
41	15	0.9721	0.9711	-0.0010
45	16	0.9828	0.9816	-0.0012
46	17	0.9841	0.9827	-0.0014
47	18	0.9664	0.9653	-0.0011
48	19	0.9664	0.9652	-0.0012
49	20	0.9722	0.9708	-0.0014
50	21	0.9790	0.9775	-0.0015
51	22	0.9793	0.9779	-0.0014
52	23	0.9646	0.9636	-0.0010
53	24	0.9637	0.9627	-0.0010
55	25	0.9602	0.9596	-0.0006
56	26	0.9415	0.9406	-0.0009
57	27	0.9673	0.9670	-0.0003
63	28	0.9899	0.9897	-0.0002
64	29	0.9462	0.9459	-0.0003
69	30	0.9340	0.9338	-0.0002

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0334	0.0040
4	3	-0.0728	-0.0758	-0.0029
5	4	-0.0870	-0.0911	-0.0041
7	5	-0.1310	-0.1411	-0.0101
10	6	-0.1028	-0.1071	-0.0042
16	7	-0.1241	-0.1297	-0.0056
17	8	-0.1067	-0.1115	-0.0048
25	9	-0.1317	-0.1398	-0.0081
26	10	-0.1690	-0.1794	-0.0104
36	11	-0.0922	-0.1013	-0.0091
37	12	-0.1482	-0.1564	-0.0081
39	13	-0.1199	-0.1283	-0.0083
40	14	-0.1666	-0.1752	-0.0087
41	15	-0.1698	-0.1786	-0.0088
45	16	-0.1626	-0.1717	-0.0091
46	17	-0.1712	-0.1810	-0.0098
47	18	-0.1832	-0.1925	-0.0093
48	19	-0.1874	-0.1971	-0.0098
49	20	-0.1839	-0.1941	-0.0101
50	21	-0.1780	-0.1882	-0.0102
51	22	-0.1778	-0.1878	-0.0100
52	23	-0.1798	-0.1889	-0.0091
53	24	-0.1864	-0.1958	-0.0094
55	25	-0.1855	-0.1937	-0.0082
56	26	-0.1938	-0.2020	-0.0082
57	27	-0.1799	-0.1871	-0.0071
63	28	-0.1111	-0.1160	-0.0049
64	29	-0.2040	-0.2116	-0.0076
69	30	-0.2213	-0.2290	-0.0077

those on the 30 substation test network. The solution times on the larger networks is obviously greater and a comparison of the solution times of the Newton Raphson and the linear programming state estimators for both the 57 and 118 substation test networks have been listed in tables 6.18 and 6.19 respectively.

The Newton Raphson least squares state estimator was the same as the one used in all the previous examples, however the dimension statements of all the data arrays were increased to accommodate the additional network information. The solution times for the 4 stage decomposed linear programming state estimator were produced by the nodal version of the program, however since neither of the two large test networks have any links in them, then the solution times will be identical to those of the original 4 stage decomposed linear programming state estimation program. It must be pointed out that the nodal version of the linear programming estimator would still be susceptible to switch status measurement errors which lead to errors in the nodal lists generated by the topology program. Thus ideally the original program should be run on these networks, but this program contains data arrays for the link information which would unnecessarily increase the memory storage requirements of the executable code. The solution to this problem of storing unused data arrays would be to change the value of the parameter defining the number of links in the master dimension file. However this would affect other users of the master dimension file and as only solution times were required on the larger networks it was considered easier to use the nodal version of the program.

The results in tables 6.18 and 6.19 demonstrate further that the linear programming estimator returns faster solution times when operating on a set of noisy measurement values than when operating on an idealistic set of noise free measurement values. These solution times also illustrate the degree of the non-linearity of the problem size verses the solution time of both the least squares and the linear programming methods of solving a set of linear equations.

No attempt has been made to draw any precise conclusions from the solution times from each of the four test networks because the solution time of a set of linear equations not only depends on the number of equations but also on the number of terms in each equation. Each of the four test networks has a different ratio of the number of nodes to lines, generators and loads, thus the number of terms in any given equation varies for each of the four test networks. The results show however that the increase in solution time is a quadratic function of the network size. It should also be noted that apart

Table 6.18: Comparison between the solution times of the 4 stage linear programming and the Newton-Raphson least squares state estimation programs on the 57 substation test network

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	28.03 (2)	82.17 (5)
Force to run	0.53 (0)	4.75 (1)
Open line 1	23.54 (2)	29.00 (3)
Open load 7	25.47 (2)	21.43 (3)
Close all switches	31.07 (3)	26.90 (3)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	32.65 (3)	66.55 (4)
Force to run	0.97 (1)	2.75 (1)
Open line 1	26.69 (3)	25.31 (4)
Open load 7	27.50 (3)	23.70 (3)
Close all switches	31.38 (3)	25.04 (3)
Measurement update	24.95 (3)	21.21 (3)

Table 6.19: Comparison between the solution times of the 4 stage linear programming and the Newton-Raphson least squares state estimation programs on the 118 substation test network

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	202.94 (3)	416.08 (6)
Force to run	0.68 (0)	32.63 (1)
Open line 50	111.29 (2)	161.70 (4)
Open load 7	118.68 (2)	124.83 (3)
Close all switches	141.92 (3)	190.64 (4)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	213.56 (4)	441.93 (6)
Force to run	4.70 (1)	7.27 (1)
Open line 50	151.09 (3)	109.29 (3)
Open load 7	169.41 (3)	101.63 (3)
Close all switches	145.71 (3)	105.18 (3)
Measurement update	145.47 (3)	94.47 (3)

from the initial solution time the solution times of the linear programming method on the 118 substation test network are comparable with, or even less than those of the least squares method. This implies that the value of the quadratic term in the relationship between the solution times and the network size is considerably less for the linear programming method than for the least squares method.

No estimates have been listed in table form because as already mentioned, the reliability of the estimates were similar to those obtained on the 30 substation test network and hence no further benefit can be gained from including the estimates.

6.8 Estimation of the link power flows

The original version of the 4 stage decomposed linear programming state estimator operates at the bus-section level as opposed to the nodal level of conventional state estimation techniques. This means that the algorithm considers the power flows through the links which inter-connect the bus-sections within a substation. The algorithm is therefore able to estimate both the real and reactive power flows for the links. A conventional state estimator using the standard equations given in chapter 3 is unable to operate at the bus-section level because these equations cannot be applied to a link which is assumed to have zero impedence.

A recent paper by Rossier & Germond[108] suggests that there are benefits to be gained by monitoring the power flow through circuit breakers and bus couplers and altering the status of adjacent switch gear to minimise the possibilities of overloading any of the equipment. Thus an estimate of the present power flow values through the links would be of use to both a power system operator and an automatic switch status optimisation program.

The link power flow estimates calculated by the 4 stage decomposed linear programming state estimator for the 30 substation test network in its initial steady state have been listed in table 6.20. The estimates were calculated from a measurement set subject to the addition of no noise or gross errors.

As is the case withany estimation process the system must be observable. That is there must be sufficient measurements to allow the estimate for every point in the system to be calculated either directly from a measurement or indirectly using a property of the system in conjunction with an adjacent measurement. There are insufficient link power flow measurements to allow a direct estimation of the link power flows at the following substations: 8, 10, 12, 27, 29 and 30. However only substations 8 and 12 have unobservable link

Table 6.20: Link power flow estimates from the state estimation program on the 30 substation test network with no measurement errors

Values are in P.U.
Error = Estimate - True value

Link Number		True value	Estimate	Error
1	Active	-0.4702	-0.4702	0.0000
1	Reactive	-0.1191	-0.1191	0.0000
2	Active	-0.4702	-0.4702	0.0000
2	Reactive	-0.1191	-0.1191	0.0000
3	Active	0.0904	0.0904	0.0000
3	Reactive	-0.0040	-0.0040	0.0000
4	Active	-0.1277	-0.1277	0.0000
4	Reactive	-0.0335	-0.0335	0.0000
5	Active	-0.1277	-0.1277	0.0000
5	Reactive	-0.0335	-0.0335	0.0000
6	Active	-0.2643	-0.2643	0.0000
6	Reactive	-0.0732	-0.0732	0.0000
7	Active	0.1366	0.1366	0.0000
7	Reactive	0.0397	0.0397	0.0000
8	Active	0.0793	0.0793	0.0000
8	Reactive	0.0018	0.0018	0.0000
9	Active	0.0753	0.0753	0.0000
9	Reactive	0.0320	0.0320	0.0000
10	Active	-0.0483	-0.0483	0.0000
10	Reactive	-0.0056	-0.0056	0.0000
11	Active	-0.2237	-0.2237	0.0000
11	Reactive	-0.0362	-0.0362	0.0000
12	Active	-0.1276	-0.1276	0.0000
12	Reactive	-0.0073	-0.0073	0.0000
13	Active	-0.0040	-0.0040	0.0000
13	Reactive	0.0302	0.0302	0.0000
14	Active	0.0720	0.0720	0.0000
14	Reactive	0.0343	0.0342	0.0000
15	Active	0.1236	0.1236	0.0000
15	Reactive	0.0376	0.0376	0.0000
16	Active	-0.1754	-0.1754	0.0000
16	Reactive	-0.0306	-0.0306	0.0000
17	Active	0.0760	0.0760	0.0000
17	Reactive	0.0040	0.0040	0.0000
18	Active	-0.0593	-0.1186	-0.0593
18	Reactive	-0.0518	-0.1036	-0.0518
19	Active	0.0000	0.0000	0.0000
19	Reactive	0.0000	0.0000	0.0000
20	Active	0.1500	0.3000	0.1500
20	Reactive	0.1500	0.3000	0.1500
21	Active	-0.0907	-0.1814	-0.0907
21	Reactive	-0.0983	-0.1964	-0.0982
22	Active	0.0593	0.0000	-0.0593
22	Reactive	0.0518	0.0000	-0.0518
23	Active	0.1500	0.0000	-0.1500
23	Reactive	0.1500	0.0000	-0.1500
24	Active	-0.0907	0.0000	0.0907
24	Reactive	-0.0983	0.0000	0.0983

25	Active	-0.1001	-0.2003	-0.1001
25	Reactive	-0.0836	-0.1672	-0.0836
26	Active	0.0000	0.0000	0.0000
26	Reactive	0.0000	0.0000	0.0000
27	Active	0.0094	0.0188	0.0094
27	Reactive	-0.0146	-0.0293	-0.0146
28	Active	0.1001	0.0000	-0.1001
28	Reactive	0.0836	0.0000	-0.0836
29	Active	0.0094	0.0000	-0.0094
29	Reactive	-0.0146	0.0000	0.0146
30	Active	-0.0778	-0.0778	0.0000
30	Reactive	-0.0832	-0.0832	-0.0001
31	Active	-0.0321	-0.0321	-0.0001
31	Reactive	-0.0997	-0.0997	0.0000
32	Active	0.1378	0.1378	0.0000
32	Reactive	0.1132	0.1132	0.0000
33	Active	-0.0285	-0.0285	0.0000
33	Reactive	0.0068	0.0067	0.0000
34	Active	-0.1101	-0.1101	0.0000
34	Reactive	-0.0433	-0.0433	0.0001
35	Active	0.0590	0.0590	0.0000
35	Reactive	0.0564	0.0564	0.0000
36	Active	0.0729	0.0728	-0.0001
36	Reactive	0.0530	0.0531	0.0000
37	Active	0.0213	0.0213	0.0000
37	Reactive	0.0031	0.0031	0.0000
38	Active	-0.0457	-0.0456	0.0001
38	Reactive	0.0165	0.0165	0.0000
39	Active	-0.1699	-0.1699	0.0000
39	Reactive	-0.0230	-0.0230	0.0000
40	Active	-0.0718	-0.0718	0.0000
40	Reactive	-0.0167	-0.0167	0.0000
41	Active	-0.1692	-0.1691	0.0001
41	Reactive	-0.0996	-0.0996	0.0000
42	Active	-0.1625	-0.3251	-0.1625
42	Reactive	-0.0267	-0.0534	-0.0267
43	Active	-0.1625	0.0000	0.1625
43	Reactive	-0.0267	0.0000	0.0267
44	Active	-0.0082	-0.0082	0.0000
44	Reactive	-0.0060	-0.0060	0.0000
45	Active	-0.0649	-0.0649	0.0000
45	Reactive	-0.0038	-0.0038	0.0000
46	Active	-0.0294	-0.0294	0.0000
46	Reactive	-0.0054	-0.0054	0.0000
47	Active	-0.0437	-0.0437	0.0000
47	Reactive	-0.0045	-0.0045	0.0000
48	Active	-0.0448	-0.0448	0.0000
48	Reactive	-0.0163	-0.0163	0.0000
49	Active	-0.0448	-0.0448	0.0000
49	Reactive	-0.0163	-0.0163	0.0000
50	Active	0.0824	0.0825	0.0000
50	Reactive	0.0223	0.0224	0.0000
51	Active	0.0204	0.0204	0.0000
51	Reactive	0.0054	0.0054	0.0000
52	Active	0.0507	0.0507	0.0000
52	Reactive	0.0115	0.0116	0.0001
53	Active	-0.0824	-0.0825	0.0000
53	Reactive	-0.0224	-0.0224	0.0000

54	Active	0.0838	0.0838	0.0000
54	Reactive	0.0270	0.0271	0.0001
55	Active	-0.0507	-0.0507	0.0000
55	Reactive	-0.0115	-0.0116	-0.0001
56	Active	0.0185	0.0185	0.0000
56	Reactive	0.0031	0.0031	0.0000
57	Active	-0.0305	-0.0305	0.0000
57	Reactive	-0.0076	-0.0076	0.0000
58	Active	0.0120	0.0120	0.0000
58	Reactive	0.0045	0.0045	0.0000
59	Active	0.0000	0.0000	0.0000
59	Reactive	0.0000	0.0000	0.0000
60	Active	-0.0185	-0.0185	0.0000
60	Reactive	-0.0031	-0.0031	0.0000
61	Active	0.0305	0.0305	0.0000
61	Reactive	0.0076	0.0076	0.0000
62	Active	-0.0120	-0.0120	0.0000
62	Reactive	-0.0045	-0.0045	0.0000
63	Active	-0.0346	-0.0346	0.0000
63	Reactive	-0.0068	-0.0068	0.0000
64	Active	-0.0346	-0.0346	0.0000
64	Reactive	-0.0068	-0.0068	0.0000
65	Active	0.0530	0.0530	0.0000
65	Reactive	0.0095	0.0095	0.0000
66	Active	0.0530	0.0530	0.0000
66	Reactive	0.0095	0.0095	0.0000
67	Active	-0.0184	-0.0184	0.0000
67	Reactive	-0.0027	-0.0027	0.0000
68	Active	-0.0184	-0.0184	0.0000
68	Reactive	-0.0027	-0.0027	0.0000

power flows, the power flows in the links at the other substation can be calculated from adjacent generator, load and line power measurements and the appropriate bus-section power flow sum check equations. Thus as can be seen in table 6.20 the estimates for links 18 to 29 in substation 8 and links 42 and 43 in substation 12 are all in error.

The values of the link power flow estimates in substations 8 and 12 are such that the majority of the estimates are set to zero, as defined by the dummy measurement equations, and the minimum number of links as possible have been assigned a power flow such that the sum of the power flows of all generators, loads and lines connected to the substation is zero. In other words the bus-section power flow sum check equation for each bus-section has been satisfied. This result is to be expected when the set of linear equations and their relative weighting factors are considered. Each link which has no power flow measurement is defined to have a power flow estimate of zero by a dummy measurement equation which has a low weighting thus allowing the equation to be readily violated. The link power flow estimate may also be defined by either one of the bus-section power flow sum check equations at each end of the link, however one equation may only define the power flow estimate of one element in the system. These equations have a high weight which forces the estimates to satisfy the equations. The power flow estimates of the genertors, loads and lines attached to the bus-section will be defined by other equations, thus in order to satisfy the sum check equation one link connected to the bus-section will have a power flow estimate equal but opposite to the sum of the power flow estimates of the generators, loads and lines connected to the bus-section. The remaining links at the bus-section will have a power flow estimate of zero as defined by the dummy measurement equation.

It is likely that for the foreseeable future the link power flow measurements will not be available. In this situation any substation which has more than n-1 links, where n equals the number of bus-sections, will be unobservable as far as the link power flow estimates are concerned. Some time was spent in an effort to develop a heuristic method for estimating the link power flows for the unobservable links. The general idea was to increase the weighting factors of the dummy measurement equations and set the input value of the equation to a suitable value based on the previous estimate but amended slightly in such a way so as to reduce the larger link power flow estimates and increase the smaller ones. These attempts however gave no siginficant improvement in the unobservable link flow estimates.

If reliable link flow estimates are desired then they could be obtained

using a method identical to the method used by the simulator to calculate the link power flow measurement values. A separate least squares problem for the link power flow estimates is set up at the end of the main estimation problem. The technique is explained fully in chapter 3, however a brief explanation follows. A set of equations are formed which equate the sum of the link power flow estimates at each bus-section to the total power flow injected into the bus-section from all the generators, loads and lines. The problem uses the estimates of the generator, load and line power flows which have already been evaluated. It would be advisable to omit any link power flow measurements, should any exist as a corrupt measurement value would only corrupt the estimates as there is no redundancy to enable the error to be corrected. The problem is solved using the least squares method described in chapter 5 with the initial value of the link power flow estimates set to zero. The solution obtained is such that the sum of the squares of the link power estimates is a minimum, subject to the constraint that the sum of the flow estimates at each bus-section is equal to the total injection. The total time required to solve both the active and reactive problems would be in the order of 5 seconds.

A full investigation of the benefits of including a least squares estimation of the link power flows has not been possible. It is likely that for the present it would not be necessary to include a separate estimation section for the link power flows. However it would still be advantageous to include the links in the overall estimation problem so that the program is able to operate at the bus-section level and retain its ability to reject corrupt switch status measurements which would otherwise lead to the nodal representation of the network being invalid.

Chapter 7
Network flow techniques

7.1 Introduction to network flow techniques

This chapter describes the adaptation of a network flow technique to solve the active and reactive power flow sub-estimation problems. Network flow problems are usually, but not necessarily, described in terms of a linear optimisation problem. The theory and implementation of the two techniques used to solve the linear set of equations formed in each of the 4 sub-estimation problems has been outlined in chapter 5 and appendix 4. The first of the two techniques namely the Revised Simplex method minimises an objective function evaluated from the modulus of a weighted error term associated with each equation. The second technique, namely the least squares method, minimises an objective function evaluated from the weighted square of an error term associated with each equation.

A mathematical algorithm also developed from the original Simplex method of Dantzig[32] minimises an objective function based on the flow within a network, hence the name network flow technique or alternatively Netflow technique. The problem is described in terms of a number of nodes and inter-connecting arcs or pipes. Each arc has an associated cost which may be either a linear or non-linear function of the flow in the arc. The solution to the problem minimises the sum of the costs of all the arcs while maintaining the flow injections at all the nodes. Obvious applications for this technique include urban traffic systems, railway systems, pipe network systems and communication systems.[70] Other authors have applied the technique to the solution of electrical network problems, examples of the application in this area include contingency analysis,[60] economic dispatch[83,88,119] and fuel scheduling together with active power generation rescheduling and load shedding.[23,24] However little has been published on the use of network flow techniques in state estimation in electrical power systems.

A detailed discussion of the history, application and solution techniques of network flow problems has been published by Kennington and Helgason[70]. The text also includes the Fortran code of a Netflow solution algorithm entitled NETFLO. The authors claimed fast solution times for the algorithm in comparison with other solution techniques including the Out-of-Kilter method. Initial comparisons of the NETFLO algorithm with an Out-of-Kilter algorithm already implemented on the machine supported this claim.

The sub-estimation problems for the active and rective power flows could easily be visualised in terms of the power flowing through the transmission

lines of the electrical power system being represented by the arcs of the network flow problem, the electrical nodes being represented by the nodes of the network flow problem. The fast solution times of the NETFLO algorithm suggested that the application of the technique to the active and reactive power-flow sub-estimation problems might result in an improvement of the overall solution times.

7.2 The theory of the solution of network flow problems

The reader is referred to the publication by Kennington and Helgason[70] for a detailed discussion on the theory of the solution of network flow problems or alternatively to the paper by Hobson, Fletcher and Stadlin[60] for a brief discussion. However a summary of the theory relevant to the implementation of the technique in the context of the 4 stage decomposed state estimation algorithm together with those features which pertain to the fast solution times will be presented in this section.

As has already been mentioned in the preceding section the network flow problem is defined in terms of an inter-connected system of arcs and nodes. A typical system is illustrated in figure 7.1. Both the nodes and the arcs are numbered in sequence, thus the network may be described by a node-arc incidence matrix as illustrated by figure 7.2. The nodal inter-connections of the arcs are represented by the columns of the matrix, it should be noted that each arc may have a flow in one direction only and the direction of the flow is indicated by the sign of the elements in the node-arc incidence matrix. A positive element implies that the arc originates at that node and hence the flow is directed away from the node, while a negative element indicates that the arc terminates on the node and consequently the direction of the flow is towards the node. The flows in the arcs are usually represented by the vector X. Each node may have an optional injection which may be positive or negative, i.e. a supply or a demand, the nodal injections are usually represented by the vector B. The nodal injections are not represented in the initial node arc incidence matrix. Each arc has an associated upper and lower flow limit which formulate a set of constraints which must be adhered to in obtaining the solution. The algorithm NETFLO has been implemented with the constraint that the cost associated with the flow in a given arc is a linear function of the flow in the arc, thus each arc has an associated cost term. The solution to the network flow problem thus minimises the sum over all the arcs of the product of the flow and the cost term, subject to the upper and lower constraints mentioned above. The network flow problem can thus be

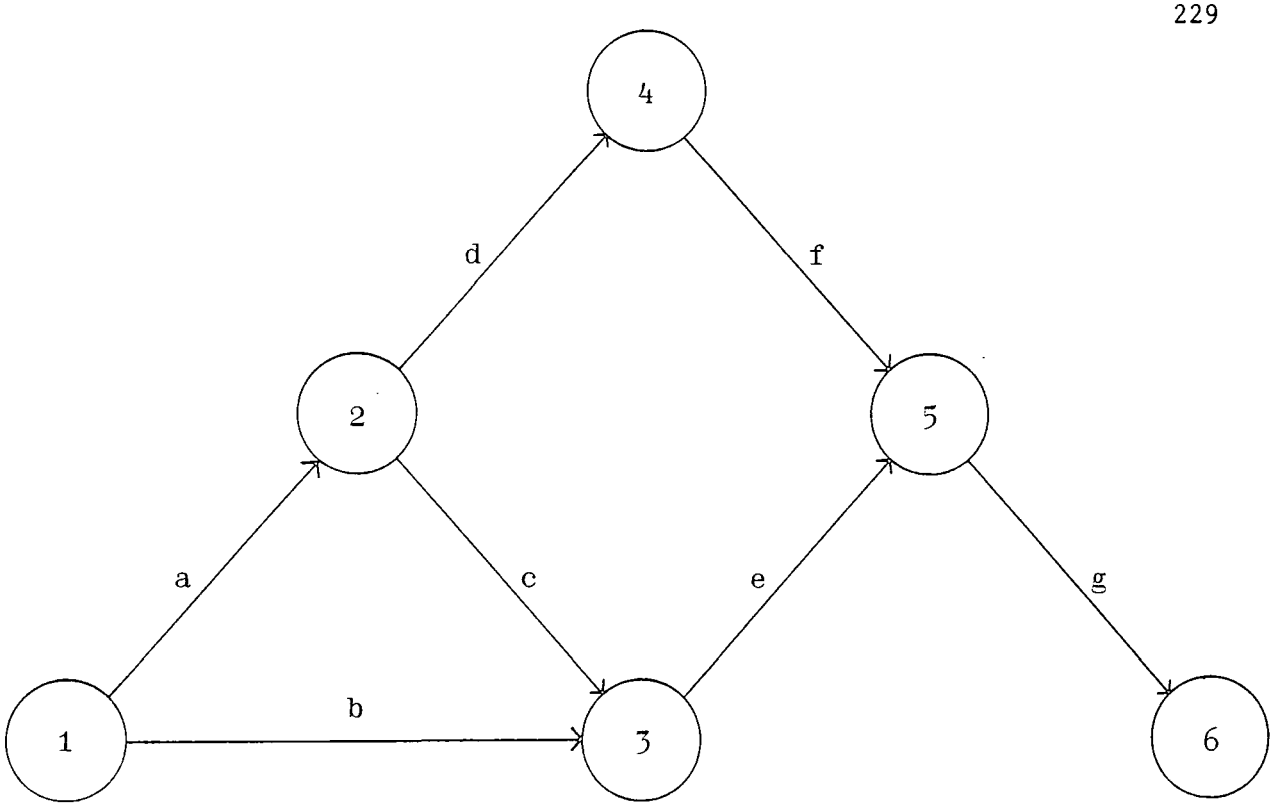


Fig. 7.1: Illustrative network

		branch						
		a	b	c	d	e	f	g
	1	+1	+1					
	2	-1		+1	+1			
node	3		-1	-1		+1		
	4				-1		+1	
	5					-1	-1	+1
	6							-1

Fig. 7.2: Node-arc incidence matrix

written in terms of a conventional linear programming problem. That is the algebraic sum of the flows on all the arcs which originate or terminate at a node must equal the nodal injection. The objective is to minimise the objective function which is evaluated from the flows in the arcs, subject to the constraints that the flows in all the arcs are within the upper and lower bounds specified for each arc. Mathematically the problem may thus be written as

$$\begin{aligned} &\text{Min} && CX \\ &\text{subject to} && AX = B \\ & && L \leqq X \leqq U \end{aligned} \tag{7.1}$$

where

C = vector of cost coefficients.
X = vector of arc flows.
A = node arc incidence matrix.
B = vector of nodal injections.
L = vector of lower arc flow limits.
U = vector of upper arc flow limits.

The solution of the network flow problem using linear programming can be made more efficient than a general linear programming algorithm by exploiting the unique structure of the node-arc incidence matrix A. These enhancements to general linear programming algorithms can be derived by the application of graph theory[34,43] to the node-arc incidence matrix. The formulation of the Revised Simplex process divides the variables into two groups, namely those variables which are non-basic and thus have a value of zero and those which are basic and are defined by the multiplication of the inverse of the basis matrix and the input vector. Likewise the formulation of the network flow problem requires that the arcs of the network be divided into two groups. The two sets of arcs form a tree and a co-tree respectively. The tree is a set of arcs which interconnect all the nodes of the network without forming any closed paths, the co-tree is the remainder of the arcs not required to form the tree. Figure 7.3 illustrates a set of arcs which form a tree for the network illustrated in figure 7.1. It can be shown that the rank of the sub-set of the node-arc incidence matrix formed from the columns representing the arcs of the tree in the original node-arc incidence matrix is one less than the number of nodes.[70] In order to form a matrix which is a full rank it is required that the tree be a rooted tree in which one node has only one arc

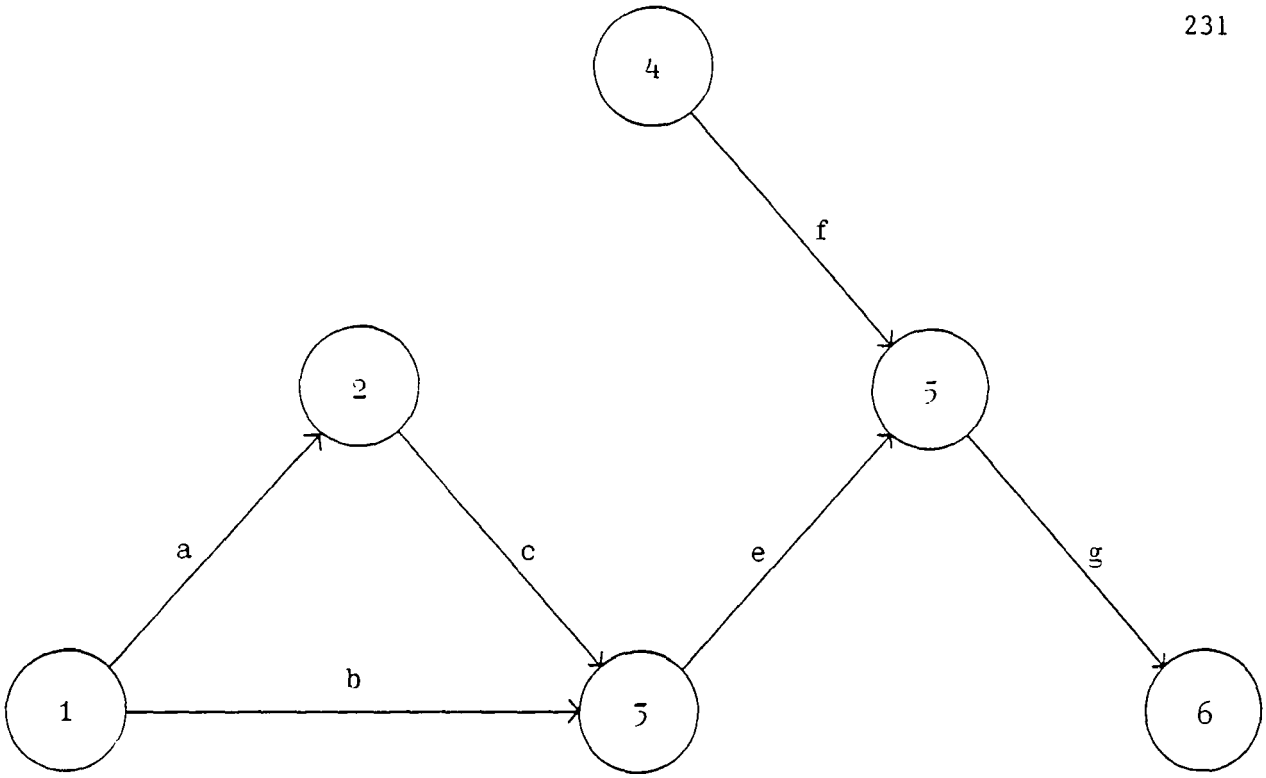


Fig. 7.3: A tree formed from fig. 7.1

		branch				
		a	b	e	f	g
	1	+1	+1	0	0	0
	2	-1	0	0	0	0
node	3	0	-1	+1	0	0
	4	0	0	0	+1	0
	5	0	0	-1	-1	+1

Fig. 7.4: Reduced node-arc incidence matrix

terminating on it, this arc and the corresponding node are known as the root arc and the root node respectively. The formation of a rooted tree may be achieved in one of two ways, there may already exist a suitable arc and node or alternatively an additional arc and node may be appended to the tree. It should be noted that the root node is to be used as a reference point and is not required in the node-arc incidence matrix, thus if the second alternative of forming a rooted tree is adopted then only the additional column representing the root arc need be appended to the node-arc incidence matrix. The result of this operation is a tree which has a square node-arc incidence matrix with only a +1 in the column representing the root arc. In figure 7.3 the root arc and the root node are arc g and node 6 respectively, figure 7.4 illustrates the reduced node-arc incidence matrix for the network in figure 7.3. The reduced node-arc incidence matrix may be inverted by the formation of the branch-path connection matrix. This matrix has elements at the intersection of rows and columns if the arc represented by the column of the reduced arc-node incidence matrix is part of a path from the node represented by the row of the matrix to the reference node. The element is assigned the value of +1 if the direction of the arc corresponds to the direction of the path and the value of -1 if it opposes the direction of the path. The remainder of the matrix is zero. It can be shown[15] that the transpose of the branch-path connection matrix is the inverse of the reduced are-node incidence matrix. The branch-path connection matrix is illustrated in figure 7-5. This method of evaluating the inverse of the reduced arc-node matrix thus eleminates the need to store the inverse of the matrix and also the time consuming process of updating the inverse matrix during each iteration.

The solution of a network flow problem is analogous to the solution method of the Simplex method. The solution of an over determined set of linear equations by the Simplex method involves the selection of a suitable non-basic variable to enter the basis which will lead to the reduction in the objective function. A suitable variable is then selected to leave the basis and an exchange takes place. The process of exchanging the variables requires the definition of the non-basic variable in terms of the basic variable. This may be achieved by multiplying the inverse of basis into the column vector representing the non-basic variable. (See chapter 5 and appendix 4 for further details). The solution of the network flow problem involves the exchange of an arc in the co-tree with one in the tree, subject to the minimisation of the objective function. As the non-basic variables in the linear programming problem are defined to be zero while the basic variables may have any value so the arcs of the co-tree must have a flow equal to their

		branch				
		a	b	e	f	g
	1	0	+1	+1	0	+1
	2	-1	+1	+1	0	+1
node	3	0	0	+1	0	+1
	4	0	0	0	+1	+1
	5	0	0	0	0	+1

Fig. 7.5: Branch-path connection matrix

upper or lower limit while the arcs of the tree may have any value within their upper and lower limits. The arc which is to enter the tree is defined in terms of the arcs already in the tree by multiplying the inverse of reduced arc-node incidence matrix into the column vector of the reduced arc-node incidence matrix which represents the arc entering the tree. This is equivalent to evaluating the sensitivity of the basic variables to the non-basic variable in a linear programming problem. Thus in figure 7.1 and 7.3 if the arc d is to be brought into the tree its definition in terms of the arcs already in the basis may be found by multiplying the transpose of the branch-path connection matrix, see figure 7.5 into the column headed arc d of the reduced arc-node incidence matrix, figure 7.4. This results in a vector which contains the elements [-1 +1 +1 -1 0]. Thus the arc d is equivalent to the path obtained using the tree arcs a,b,e and f (excluding the tree arc g)

a negative sign implies the direction is opposite to that of the original tree arc. In other words arc d is equivalent to the reverse of arc a followed by arcs b and e and then finally the reverse of arc f. The determination of which arc is to leave the tree is made by evaluating which arc has the smallest ratio of the permissible movement to its sensitivity. The selection of the co-tree arc to enter the tree is made by evaluating the relative objectives of the arcs. However while in the case of a linear programming problem the value of a non-basic variable may only increase from zero the value of the flow in a co-tree arc may increase or decrease depending on which limit the flow is presently set at. Thus the direction in which the flow will be changed also has to be considered once the relative objectives have been determined.

The physical interpretation of exchanging a tree arc with a co-tree are in terms of the effect on the flows within the network can be thought of in the following terms. When a co-tree arc is brought into the tree the flow in the co-tree arc is moved towards its opposite limit and the flows in the tree arcs which describe the co-tree arc are changed to reflect the change in the flow of the co-tree arc. When one of the tree arcs reaches a limit then that arc is the one which the co-tree arc will replace. The flows in the remaining tree arcs being set to their flow value at the point at which the outgoing arc reaches its limit. Should the flow in the co-tree arc reach its opposite limit before the flow of any of the tree arcs reaches a limit then the flows in the tree arcs are modified but no exchange takes place. This phenomena is known as a bounding iteration.

The basic procedure for solving a network flow problem is thus analogous to that for solving a linear programming problem, however it is possible to

take advantage of the unique structure of the node-arc incidence matrix to reduce the solution times. The method of evaluating the inverse inverse of the reduced node-arc incidence matrix has already been mentioned, the entire problem may also be solved using only integer storage arrays and integer arithmetic which will result in faster manipulation of the data. Numerous other modifications are possible for enhancing the performance of the method, including the evaluation of a feasible starting point, for further details the reader is referred to the references previously mentioned.[60,70]

7.3 Implementation of the network flow technique

The NETFLO program listed in the book by Kennington and Helgason was entered into a file and minor modifications made which involved the conversion of the initial input section from external read statements to a subroutine input. The major inputs to the subroutine were thus as follows: two vectors defining the sending and receiving nodes of each of the arcs, two vectors defining the upper and lower flow limits for each of the arcs, a vector defining the cost of the flow in each arc and finally a vector defining the injection at each of the nodes in the network flow problem. The major outputs of the subroutine were as follows: a vector containing the optimised arc flows, a vector containing the cost of the flow associated with each arc and the total cost of the solution. It should be noted that the entire subroutine uses integer arithmetic only, thus in order to retain an accuracy of approximately five decimal places on a power flow of 1.0 per unit the power flow measurements are all multiplied by 100,000 and then converted to an integer number. The reverse procedure was used to translate the optimal network flow solution to the power flow estimates within the power system.

It would be possible to define the active and reactive sub-estimation problems in terms of the network flow variables such that there is a one to one correspondence between the bus-sections of the power system and the nodes of the network flow problem. However this would require that the injection at every bus-section be accurately known and that there are no power flow losses along the transmission lines. The former requirement arises because the injection values at the nodes of the network flow problem are used in determining the optimal flows along the arcs. Thus any error in the value of the nodal injection would lead to errors in the values of the arc flows and hence errors in the final power flow estimates. It is unlikely that the injection at every bus-section is known and any measured values are bound to contain some errors of one degree of another. The latter requirement of no

transmission line losses would lead to a similar situation found in the original data validation program discussed in chapter 4. The imposition of the requirement that the transmission line losses were zero resulted in the estimates of the load and generator power flows being in error in order to satisfy the bus-section power flow sum check equations. The same problem is likely to arise for a network flow problem because both the injection values and the power flow sum check equation would be adhered to which may result in an arc flow being defined by two contradicting equations.

The active and reactive sub-estimation problems have thus been defined by a network flow problem in which the injection at the nodes has been set to zero. The power system injections, i.e. the generator and load power flows together with the transmission line losses have all been represented by arcs in the network flow problem. These arcs all terminate at or originate from an additional super node added to the network flow problem. The network flow problem is thus a source-less and sink-less network. As well as adding the super node an addition node has to be added in each line, this node is the imaginary point at which the line power flow losses, attributed to the impedence of the line, are withdrawn from the line. There is a path from the additional line node to the super node which completes the source-less and sink-less network flow problem.

The active and reactive sub-estimation problems are thus defined by a pair of bi-directional arcs (ie two arcs connected between the same nodes with opposite directions of flow) for every point in the power system network where a power flow estimate can be made, including the estimates of the power flow losses in the transmission lines. A pair of bi-directional arcs is required to allow for power flow estimates to be positive or negative, an arc may only have a flow greater than or equal to zero. The direction of the active power flows in the arcs representing the generators, loads and the line flow losses is always fixed, i.e. the direction of flow in a generator arc is always from the super node into the network while the direction of flow in the load and line loss arcs is always from the network towards the supernode. This is not the case for the reactive power flows, the flow may be in either direction in all cases. Thus the network flow problem may be reduced slightly for the active power flow sub-estimation problem although as can be seen in the results section, section 7.4, the advantages are minimal. Each arc in the network flow problem has no upper limit, (the value is set to a large number) a lower limit of zero and a cost of zero. Thus the reactive sub-estimation problem of the simple power system network illustrated in figure 7.6 would be represented by the network flow problem illustrated in figure 7.7.

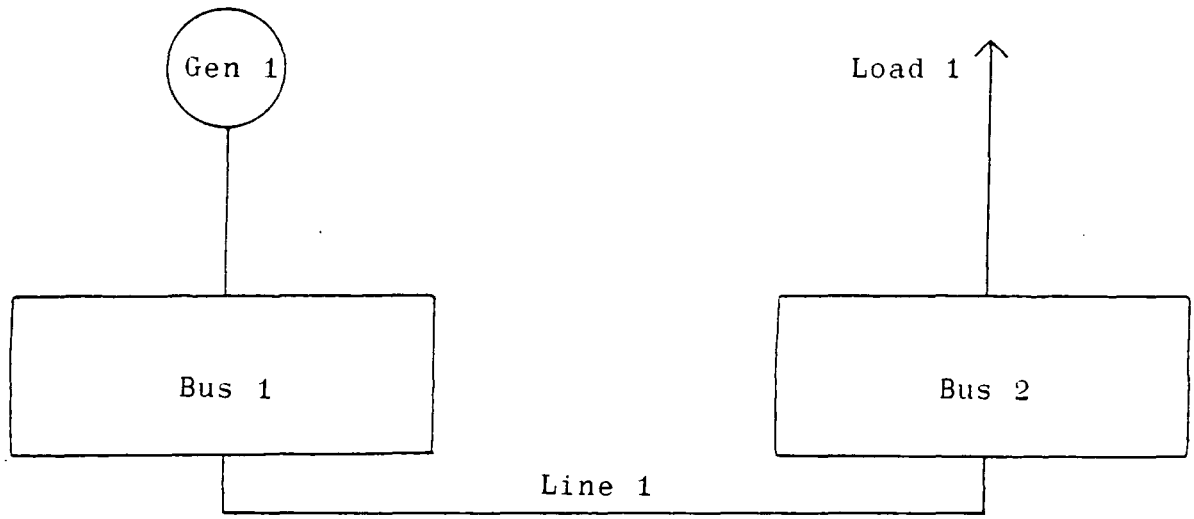


Fig. 7.6: A simple power system

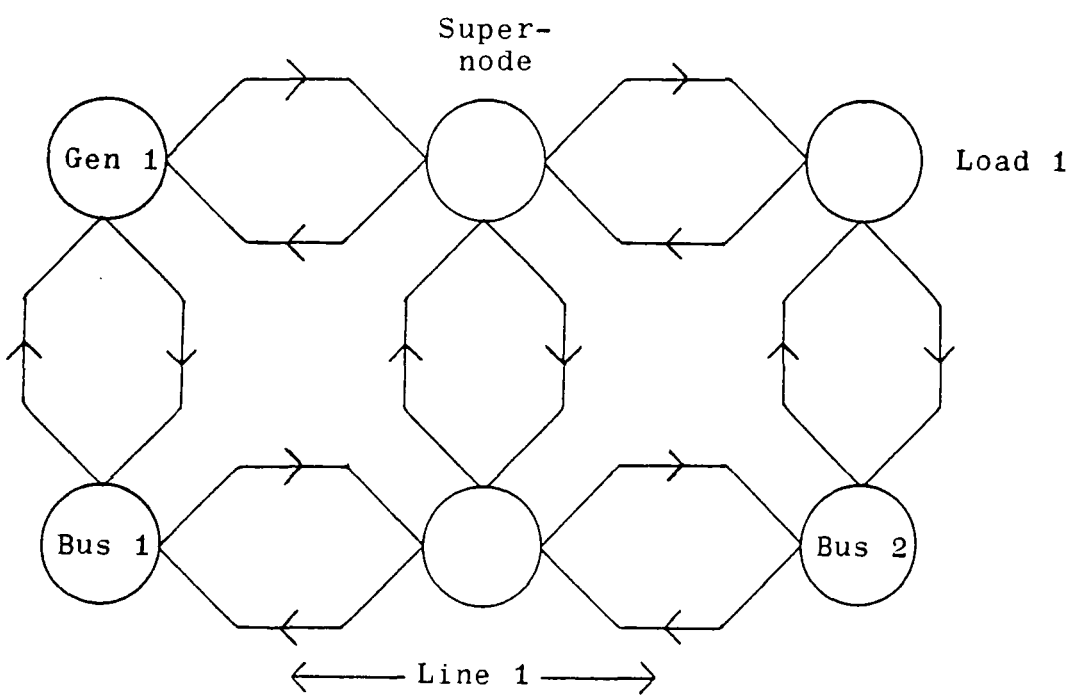


Fig. 7.7: Reactive power network flow representation of fig. 7.6

The network flow problem as it stands will have an optimal solution of zero flow in all the arcs. The power flow measurement values are needed to force a flow through the arcs of the network flow problem. To date two different methods have been used to implement the power flow measurement values and the values of the power flow losses in the transmission lines which are calculated from the previous estimates. The first method requires the addition of a pair of bi-directional arcs and a node for each power flow measurement and for each line. It should be noted that the additional arcs and node for each line are required to force the flow through the arcs representing the line losses while the measurement arcs and nodes force the flow through the arcs representing the power flow estimates.

Each of the of additional measurement arcs is assigned an identical cost which may be varied between each pair of arcs to reflect the accuracy of the measurement value. The upper flow limit of both arcs is set to infinity. The lower flow limit of the arc with the direction of the flow the same as the direction of flow indicated by the sign of the flow measurement is set to the value of the measurement (translated into network flow terms by multiplying by 100,000 and converting to an integer). The lower flow limit of the other arc in the pair is set to zero. The additional measurement arcs and node are inserted into the existing network flow problem adjacent to the arcs representing the estimate of the power flow for that point in the power system. The lower flow limit on the measurement arc which has been set equal to the measurement value forces a flow through that arc and the corresponding estimate arc. If however this flow is not in accordance with the flows in the rest of the network problem which may be defined by other measurement arcs then the resultant flow through the pair of measurement arcs may be altered to suit the requirements of the overall network. If the measurement value is too small then the resultant flow through both arcs may be increased by raising the flow above the lower limit. Alternatively the flow on the measurement arc with its direction in the opposite direction may be increased from zero if the measurement value is too big. The flow in the estimate arcs assumes the resultant flow of the two measurement arcs, however the overall cost of the solution is increased if the resultant flow is not equal to the measurement value. The network flow problem is thus solved by minimising the cost of the flows in the measurement arcs. The network flow representation of the transmission line illustrated in figure 7.8 in which there is a measurement of the power flow at the sending end is illustrated in figure 7.9. An open switch may be incorporated into the network flow problem by simply assigning a cost to the two arcs representing the estimate of the power flow at that point.

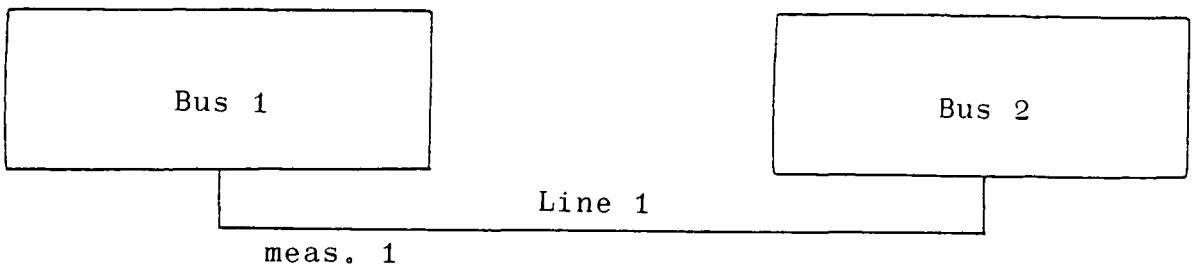


Fig. 7.8: A transmission line with a measurement at the sending end

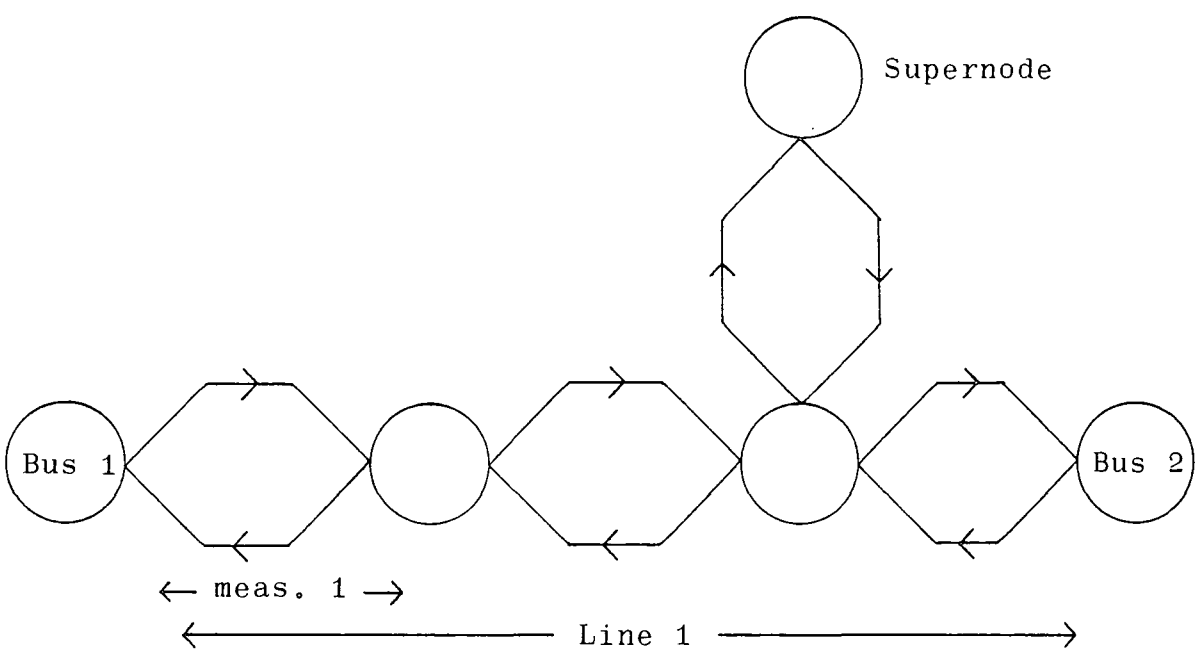


Fig. 7.9: Reactive power network flow representation of fig. 7.8

This has the effect of making it undesirable to have a flow through these arcs but it does not prevent the optimal solution from having a flow should the switch actually be closed.

The second method of implementing the active and reactive sub-estimation problems reduced the network flow problem size by omitting the measurement arcs altogether. The following restrictions were put on the power system network: each point may have only one measurement value, if two or more exist then the last value entered will be used, each point may have only one switch status measurement, if a power system element has more than one switch status measurement then they must be combined into one measurement. These restrictions have been applied merely to ease the coding of the program. It would be possible to add an additional pair of arcs and a node for each additional duplicated measurement, however this would require some fairly complex programming. The removal of the pair of measurement arcs and node stems from the fact that in the previous implementation the two pairs of arcs which represent an estimate of the power flow and its associated measurement both carry an identical resultant flow. Thus the duplication is serving no useful purpose and unnecessarily increases the size of the network flow problem. However reducing the size of the problem may reduce the ability of the method to detect and correct bad data, this point is discussed in greater detail in the following section which presents the results of the two techniques.

The representation of an open switch measurement in the network flow problem is not as straightforward as in the previous implementation. It would be possible to increase the cost on the arcs representing the estimate of the power flow at that point in the power system, as in the previous implementation. However if the estimate has an associated measurement then this method is no longer valid because to add the cost of the open switch to the cost of the measurement would bias the solution towards one in which the measurement value was accepted as being correct. If the measurement value was in error then the estimates of the power flows would be incorrect. Thus all open switch measurements have been implemented by adding an additional node together with a pair of arcs each with a lower limit of zero and no upper limit. This avoids the complications of having to determine whether or not that point in the power systems has an associated power flow measurement.

The overall size of the network flow problem and the reduction in the size of the network flow problem by eliminating the arcs and nodes associated with the measurements is outlined below.

	First implementation		Second implementation	
test network	No. of nodes	No. of arcs	No. of nodes	No. of arcs
5 subst.	46	126	17	34
30 subst.	286	786	115	222
57 subst.	332	566	136	287
118 subst.	834	2464	298	696

It should be noted no allowance has been made in the first implementation for the removal of the return arcs representing the generator load and line loss estimates in the active power sub-estimation problem.

Using the NETFLOW algorithm has led to a greater understanding of the solution of a network flow problem and possible enhancements to the implementation may be possible. The performance of the two methods of implementation is discussed in the following section, 7.4, and possible enhancements at the end of the chapter in section 7.5.

7.4 Results

7.4.1 Solution times

The solution times of both methods of implementing the network flow technique have been compared with alternative methods of state estimation. Comparisons between the solution times of the active and reactive sub-estimation problems have also been presented together with the solution times for the overall estimation process for several different operating conditions.

The results listed in table 7.1 compare the solution times for both the active and reactive sub-estimation problems of the original linear programming method with those of the first of the network flow implications. The programs were operating on the 30 substation test network with no noise or gross errors applied to the measurement values. The table presents the solution times for each main iteration as the program calculates the estimates from a flat start and then continuing with the status of line 7 changed to open. As can be seen from the results the reduction in the size of the network flow problem for the active power flow sub-estimation problem by the removal of the return arc for the generator, load and line loss estimates does not significantly reduce the solution time.

The effect that the network flow technique has on the overall solution

Table 7.1: Comparison between the solution times of the Revised Simplex method and the Network flow technique during the estimation of the states of the 30 substation test network

The times are in seconds and the number of iterations required to converge each sub-estimation stage is shown in brackets.

Solution times obtained from a flat start

Iteration	Sub-estimation type			
	P		Q	
number	Simplex	Netflow	Simplex	Netflow
1	14.67 (118)	3.62 (654)	15.05 (131)	3.19 (617)
2	1.58 (6)	3.82 (673)	2.86 (18)	3.62 (657)
3	4.00 (25)	3.75 (663)	2.33 (13)	3.65 (636)
4	2.90 (16)	3.67 (661)	3.63 (24)	3.64 (662)

Solution times obtained continuing from the above estimates with line 7 open

Iteration	Sub-estimation type			
	P		Q	
number	Simplex	Netflow	Simplex	Netflow
1	6.16 (43)	3.57 (611)	6.14 (46)	3.59 (633)
2	2.85 (16)	3.60 (588)	5.71 (43)	3.52 (609)
3	1.48 (5)	3.58 (549)	1.53 (6)	3.55 (631)

time for the estimation process can be seen in table 7.2 in which the solution times of the original linear programming implementation for numerous operating conditions are compared with those of the first network flow implementation. As can be seen from the table the advantages are minimal and no significant gain in the solution times have been achieved.

Tables 7.3 and 7.4 list the same set of results for the second implementation of the network flow technique. Table 7.3 indicates that the solution times of the network flow technique are now comparable if not faster than the equivalent Revised Simplex technique. This is reflected in the overall solution times listed in table 7.4 in which the network flow technique results are significantly faster than the linear programming technique in all but one of the examples. The reduction of the size of the network flow problem dramatically affects the solution times of each of the sub-estimation problems.

The solution times of the first implementation of the network flow technique listed in table 7.1 are fairly constant while those of the second implementation listed in table 7.3 vary considerably. The reason for the variation is not clear but it is likely to arise from the fact that in the second implementation of the network flow technique a greater proportion of the total number of arcs are defined instantly by the imposition of the lower flow limit, which represents the measurement and line loss values. In the first implementation the initial flow on the estimate arcs is zero at the start of every network flow solution, no mechanism exists for supplying a starting point. Thus the flow in every pair of arcs representing an estimate will need updating and a variable but small number of the measurement arcs will also need updating depending on the accuracy of the measurement values. In the second implementation a large majority of the arcs have their flow defined instantly by the lower flow limit and only a small but variable number of the arcs will need their flows updating to compensate for the measurement errors. Since the values of the line losses are treated as a measurement they significantly effect the solution times of the second implementation because in the early stages of the state estimation process the values are calculated from estimate which are not reliable and hence the values are likely to be in error while in the later stages the values become more accurate.

Tables 7.5 and 7.6 compare the solution times between the Revised Simplex method and the second implementation of the network flow technique on the 118 substation test network. These tables illustrate that the network flow algorithm shows a similar quadratic increase in solution time with increasing network size as the linear programming method. This is to be expected as the

Table 7.2: Comparison between the solution times of the 4 stage state estimation program using the Revised Simplex method and the Network flow technique on the 30 substation test network

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Simplex	Netflow
Start estimator	72.37 (5)	54.00 (4)
Force to run	12.00 (1)	11.75 (1)
Open line 7	30.71 (3)	30.76 (3)
Open link 6	37.39 (3)	31.52 (3)
Close all switches	41.16 (4)	31.74 (3)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Simplex	Netflow
Start estimator	74.08 (4)	81.50 (7)
Force to run	4.70 (1)	37.63 (4)
Open line 7	33.70 (4)	41.23 (4)
Open link 6	26.10 (3)	30.66 (3)
Close all switches	30.89 (4)	39.73 (4)
Measurement update	28.21 (3)	29.56 (3)

Table 7.3: Comparison between the solution times of the Revised Simplex method and the Network flow technique with the reduced Netflow problem size during the estimation of the states of the 30 substation test network

The times are in seconds and the number of iterations required to converge each sub-estimation stage is shown in brackets.

Solution times obtained from a flat start

Iteration number	Sub-estimation type P Simplex	P Netflow	Q Simplex	Q Netflow
1	14.67 (118)	2.67 (198)	15.05 (131)	2.67 (185)
2	1.58 (6)	2.60 (198)	2.86 (18)	2.60 (181)
3	4.00 (25)	2.01 (158)	2.33 (13)	2.05 (136)
4	2.90 (16)	2.05 (142)	3.63 (24)	2.18 (144)
5	2.48 (13)	1.89 (122)	2.97 (19)	2.04 (141)

Solution times obtained continuing from the above estimates with line 7 open

Iteration number	Sub-estimation type P Simplex	P Netflow	Q Simplex	Q Netflow
1	6.16 (43)	1.01 (209)	6.14 (46)	0.81 (171)
2	2.85 (16)	1.20 (226)	5.71 (43)	0.94 (190)
3	1.48 (5)	1.30 (228)	1.53 (6)	1.09 (196)

Table 7.4: Comparison between the solution times of the 4 stage state estimation program using the Revised Simplex method and the Network flow technique with the reduced Netflow problem size on the 30 substation test network

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Simplex	Netflow
Start estimator	72.37 (5)	48.57 (5)
Force to run	12.00 (1)	9.28 (1)
Open line 7	30.71 (3)	16.73 (3)
Open link 6	37.39 (3)	11.19 (3)
Close all switches	41.16 (4)	12.09 (3)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Simplex	Netflow
Start estimator	74.08 (4)	49.93 (5)
Force to run	4.70 (1)	6.99 (1)
Open line 7	33.70 (4)	12.89 (3)
Open link 6	26.10 (3)	11.81 (3)
Close all switches	30.89 (4)	18.41 (5)
Measurement update	28.21 (3)	12.26 (3)

Table 7.5: Comparison between the solution times of the Revised Simplex method and the Network flow technique with the reduced Netflow problem size during the estimation of the states of the 118 substation test network

The times are in seconds and the number of iterations required to converge each sub-estimation stage is shown in brackets.

Solution times obtained from a flat start

	Sub-estimation type			
Iteration	P		Q	
number	Simplex	Netflow	Simplex	Netflow
1	81.07 (281)	22.21 (560)	84.61 (297)	20.21 (532)
2	7.31 (18)	21.46 (448)	27.18 (85)	18.51 (432)
3	6.64 (15)	19.43 (495)	21.76 (67)	18.37 (516)
4	18.02 (53)	18.37(526)	6.81 (16)	18.00 (565)
5	11.48 (31)	17.48 (516)	5.40 (11)	18.15 (560)

Solution times obtained continuing from the above estimates with line 50 open

	Sub-estimation type			
Iteration	P		Q	
number	Simplex	Netflow	Simplex	Netflow
1	23.07 (70)	6.20 (508)	21.09 (68)	2.18 (440)
2	12.27 (33)	5.09 (549)	14.43 (41)	5.85 (570)
3	7.04 (16)	7.24 (541)	5.43 (11)	6.16 (541)

Table 7.6: Comparison between the solution times of the 4 stage state estimation program using the Revised Simplex method and the Network flow technique with the reduced Netflow problem size on the 118 substation test network

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Simplex	Netflow
Start estimator	416.08 (6)	248.46 (5)
Force to run	32.63 (1)	50.66 (1)
Open line 50	161.70 (4)	62.05 (4)
Open load 7	124.83 (3)	67.11 (5)
Close all switches	190.64 (4)	87.69 (8)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Simplex	Netflow
Start estimator	441.93 (6)	250.89 (4)
Force to run	7.27 (1)	82.61 (2)
Open line 50	109.29 (3)	113.69 (8)
Open load 7	101.63 (3)	49.34 (3)
Close all switches	105.18 (3)	66.87 (4)
Measurement update	94.47 (3)	44.81 (3)

method of solving the network flow problem is analgous to that of solving the linear equations.

Tables 7.7, 7.8 and 7.9 compare the overall solution times of the Newton Raphson least squares method with the second implementation of the network flow technique on the 30, 57 and 118 substation test networks respectively. The solution times of the network flow technique are worse than those of the least squares method on the 30 substation test network, but it must be remembered that the network flow technique is operating at the bus-section level and is hence solving a larger problem. The situation on the 57 substation test network is reversed and the network flow technique is generally faster than the least squares method and the difference in the solution times increase still further on the 118 substation test network.

The results of the solution times indicate that there are significant gains to be made in using the second implementation of the network flow technique, however a few problems have been found with the method. These are discussed in the following section.

7.4.2 Performance

As in the case of the results of the linear programming method presented in chapter 6 only the voltage magnitude and voltage phase angle estimates have been presented in tabular form.

The estimates produced by both of the network flow implementations for the 30 substation network with no noise or gross errors applied to the measurements have not been listed in a table because they are identical to those of the linear programming method which have been listed in table 6.5. The estimates produced by both implementations when the measurements have been subject to the addition of 0.2% systematic noise and 1.5% random noise components have been listed in tables 7.10 and 7.11 respectively. Comparison of these tables with table 6.6 which lists the estimate calculated by the linear programming method indicates the similar nature of the 2 methods. The magnitude of the error on each of the estimates is almost identical.

No further results of the first implementation of the network flow technique will be presented in tabular form since the results are always the same as those of table 6.5. In other words the method is able to correct analogue measurement errors and also detect and correct both single and multiple corrupt switch status measurements. It should be noted that the original linear programming method failed to correctly estimate all the reactive power flows around substation 2 when the status of the 4 line

Table 7.7: Comparison between the solution times of the Newton-Raphson least squares state estimator and the 4 stage linear programming state estimator using the reduced Netflow problem size on the 30 substation test network

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Netflow
Start estimator	7.34 (2)	48.57 (5)
Force to run	0.14 (0)	9.28 (1)
Open line 7	7.62 (2)	16.73 (3)
Open link 6	4.01 (1)	11.19 (3)
Close all switches	7.21 (2)	12.09 (3)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Netflow
Start estimator	8.84 (3)	49.93 (5)
Force to run	0.47 (1)	6.99 (1)
Open line 7	8.17 (3)	12.89 (3)
Open link 6	7.74 (3)	11.81 (3)
Close all switches	8.69 (3)	18.41 (5)
Measurement update	7.37 (2)	12.26 (3)

Table 7.8: Comparison between the solution times of the Newton-Raphson least squares state estimator and the 4 stage linear programming state estimator using the reduced Netflow problem size on the 57 substation test network

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Netflow
Start estimator	28.03 (2)	56.89 (5)
Force to run	0.53 (0)	11.55 (1)
Open line 1	23.54 (2)	16.71 (3)
Open load 7	25.47 (2)	45.04 (13)
Close all switches	31.07 (3)	20.49 (4)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Netflow
Start estimator	37.00 (3)	67.21 (7)
Force to run	1.14 (1)	7.43 (1)
Open line 1	30.66 (3)	10.85 (3)
Open load 7	27.50 (4+3)	12.66 (4)
Close all switches	34.93 (3)	17.28 (4)
Measurement update	23.63 (2)	14.45 (4)

Table 7.9: Comparison between the solution times of the Newton-Raphson least squares state estimator and the 4 stage linear programming state estimator using the reduced Netflow problem size on the 118 substation test network

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Netflow
Start estimator	202.94 (3)	248.46 (5)
Force to run	0.68 (0)	50.66 (1)
Open line 50	111.29 (2)	62.05 (4)
Open load 7	118.68 (2)	67.11 (5)
Close all switches	141.92 (3)	87.69 (8)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Netflow
start estimator	213.56 (4)	250.89 (4)
Force to run	4.70 (1)	82.61 (2)
Open line 50	151.09 (3)	113.69 (8)
Open load 7	169.41 (3)	49.34 (3)
Close all switches	145.71 (3)	66.87 (4)
Measurement update	145.47 (3)	44.81 (3)

Table 7.10: Estimates from the state estimation program using the network flow technique on the 30 substation test network with 0.2% systematic noise and 1.5% random noise

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0375	-0.0063
2	2	1.0301	1.0237	-0.0064
4	3	1.0105	1.0040	-0.0065
5	4	1.0025	0.9959	-0.0066
7	5	0.9851	0.9787	-0.0064
10	6	0.9951	0.9886	-0.0064
16	7	0.9829	0.9764	-0.0065
17	8	0.9894	0.9830	-0.0064
25	9	1.0071	1.0006	-0.0064
26	10	0.9928	0.9861	-0.0067
36	11	1.0486	1.0428	-0.0057
37	12	0.9889	0.9823	-0.0067
39	13	1.0011	0.9945	-0.0066
40	14	0.9752	0.9682	-0.0069
41	15	0.9721	0.9652	-0.0069
45	16	0.9828	0.9760	-0.0068
46	17	0.9841	0.9774	-0.0067
47	18	0.9664	0.9594	-0.0071
48	19	0.9664	0.9593	-0.0072
49	20	0.9722	0.9651	-0.0071
50	21	0.9790	0.9721	-0.0069
51	22	0.9793	0.9724	-0.0069
52	23	0.9646	0.9576	-0.0070
53	24	0.9637	0.9566	-0.0071
55	25	0.9602	0.9532	-0.0070
56	26	0.9415	0.9340	-0.0075
57	27	0.9673	0.9605	-0.0068
63	28	0.9899	0.9834	-0.0065
64	29	0.9462	0.9390	-0.0072
69	30	0.9340	0.9267	-0.0073

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0378	-0.0003
4	3	-0.0728	-0.0741	-0.0013
5	4	-0.0870	-0.0886	-0.0015
7	5	-0.1310	-0.1332	-0.0022
10	6	-0.1028	-0.1047	-0.0019
16	7	-0.1241	-0.1263	-0.0022
17	8	-0.1067	-0..1086	-0.0019
25	9	-0.1317	-0.1338	-0.0021
26	10	-0.1690	-0.1725	-0.0035
36	11	-0.0922	-0.0938	-0.0016
37	12	-0.1482	-0.1511	-0.0028
39	13	-0.1199	-0.1222	-0.0023
40	14	-0.1666	-0.1698	-0.0033
41	15	-0.1698	-0.1731	-0.0033
45	16	-0.1626	-0.1657	-0.0031
46	17	-0.1712	-0.1745	-0.0033
47	18	-0.1832	-0.1864	-0.0032
48	19	-0.1874	-0.1910	-0.0036
49	20	-0.1839	-0.1875	-0.0036
50	21	-0.1780	-0.1814	-0.0034
51	22	-0.1778	-0.1813	-0.0035
52	23	-0.1798	-0.1832	-0.0034
53	24	-0.1864	-0.1900	-0.0037
55	25	-0.1855	-0.1893	-0.0037
56	26	-0.1938	-0.1982	-0.0044
57	27	-0.1799	-0.1834	-0.0035
63	28	-0.1111	-0.1131	-0.0020
64	29	-0.2040	-0.2083	-0.0043
69	30	-0.2213	-0.2259	-0.0046

Table 7.11: Estimates from the state estimation program using the network flow technique with the reduced Netflow problem size on the 30 substation test network with 0.2% systematic noise and 1.5% random noise

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0374	-0.0064
2	2	1.0301	1.0237	-0.0065
4	3	1.0105	1.0039	-0.0066
5	4	1.0025	0.9959	-0.0067
7	5	0.9851	0.9784	-0.0067
10	6	0.9951	0.9884	-0.0067
16	7	0.9829	0.9761	-0.0068
17	8	0.9894	0.9827	-0.0067
25	9	1.0071	1.0006	-0.0065
26	10	0.9928	0.9861	-0.0067
36	11	1.0486	1.0430	-0.0056
37	12	0.9889	0.9823	-0.0067
39	13	1.0011	0.9945	-0.0066
40	14	0.9752	0.9682	-0.0069
41	15	0.9721	0.9651	-0.0070
45	16	0.9828	0.9761	-0.0067
46	17	0.9841	0.9775	-0.0066
47	18	0.9664	0.9593	-0.0071
48	19	0.9664	0.9593	-0.0071
49	20	0.9722	0.9651	-0.0071
50	21	0.9790	0.9721	-0.0068
51	22	0.9793	0.9725	-0.0068
52	23	0.9646	0.9576	-0.0071
53	24	0.9637	0.9565	-0.0072
55	25	0.9602	0.9532	-0.0071
56	26	0.9415	0.9339	-0.0075
57	27	0.9673	0.9605	-0.0068
63	28	0.9899	0.9831	-0.0068
64	29	0.9462	0.9391	-0.0071
69	30	0.9340	0.9268	-0.0072

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0379	-0.0004
4	3	-0.0728	-0.0745	-0.0017
5	4	-0.0870	-0.0890	-0.0020
7	5	-0.1310	-0.1341	-0.0031
10	6	-0.1028	-0.1052	-0.0024
16	7	-0.1241	-0.1269	-0.0028
17	8	-0.1067	-0.1091	-0.0024
25	9	-0.1317	-0.1343	-0.0026
26	10	-0.1690	-0.1729	-0.0040
36	11	-0.0922	-0.0964	-0.0041
37	12	-0.1482	-0.1516	-0.0033
39	13	-0.1199	-0.1228	-0.0029
40	14	-0.1666	-0.1703	-0.0038
41	15	-0.1698	-0.1736	-0.0038
45	16	-0.1626	-0.1662	-0.0036
46	17	-0.1712	-0.1749	-0.0037
47	18	-0.1832	-0.1872	-0.0040
48	19	-0.1874	-0.1917	-0.0044
49	20	-0.1839	-0.1882	-0.0042
50	21	-0.1780	-0.1819	-0.0039
51	22	-0.1778	-0.1818	-0.0039
52	23	-0.1798	-0.1837	-0.0039
53	24	-0.1864	-0.1906	-0.0042
55	25	-0.1855	-0.1898	-0.0043
56	26	-0.1938	-0.1987	-0.0049
57	27	-0.1799	-0.1839	-0.0040
63	28	-0.1111	-0.1136	-0.0026
64	29	-0.2040	-0.2083	-0.0043
69	30	-0.2213	-0.2261	-0.0047

switches at this substation were corrupted. This was not the case for the first implementation of the network flow technique, all the estimates were 100% correct.

The second implementation of the network flow technique showed the same ability to correct analogue measurement errors but was more prone to errors when subjected to corrupt switch status measurements. When the status of the sending end of line 5 was corrupted from closed to open the program was able to detect the switch status error. However both substations 5 and 7 had their voltage magnitude and voltage phase angle estimates in error by 0.0013 per unit and 0.0012 per unit respectively. Similarly both generator 2 and the sending end of line 5 had their active and reactive power flow estimates in error by 0.0124 per unit and 0.0095 per unit respectively. However the magnitude of the errors is small and would be of little concern to power system operator. In the case of the 4 line switch status errors at substation 2 the second implementation of the network flow technique was still able to detect the errors but a few more of the estimates were in error. The voltage magnitude and voltage phase angle estimates have been listed in table 7.12. As can be seen the error on the voltage magnitude estimate at substations 5 and 7 has doubled from the case of only one switch status error and there is a small error at substations 1 and 2. In the case of the voltage phase angle estimates a small error has been propagated throughout all of the estimates. The power flow estimates were similar to those of the linear programming method, i.e. the reactive power flow in line 1 was set to zero which results in a few of the surrounding lines having small errors on them. The network flow technique also had an error on the active power flow estimate at load 1 of -0.0151 per unit with corresponding errors in the active power flow estimates of the adjacent lines.

The reason for the failure of the second implementation in the presence of corrupt switch status measurements must be connected with the representation of the open switch measurements and the reduction in the total number of arcs by the removal of the separate estimate arcs. The reduction in the total number of arcs leads to the reduction in the ratio of the number of incorrect switch status measurement arcs to the total number of arcs. This may affect the solution point to some extent but further work is required to pinpoint the exact cause of the problem. Since the magnitude of the errors on the state estimates is small it may be that the problem could be solved by adjusting the costs of the arcs representing the switches. It is the author's opinion that a little further work would overcome this problem which would then result in a fast and sound method of implementing the 4 stage decomposed linear

Table 7.12: Estimates from the state estimation program using the network flow technique with the reduced Netflow problem size on the 30 substation test network with 4 line switch status errors

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0437	-0.0001
2	2	1.0301	1.0300	-0.0002
4	3	1.0105	1.0105	0.0000
5	4	1.0025	1.0025	0.0000
7	5	0.9851	0.9878	0.0027
10	6	0.9951	0.9951	0.0000
16	7	0.9829	0.9856	0.0027
17	8	0.9894	0.9894	0.0000
25	9	1.0071	1.0071	0.0000
26	10	0.9928	0.9928	0.0000
36	11	1.0486	1.0486	0.0000
37	12	0.9889	0.9889	0.0000
39	13	1.0011	1.0011	0.0000
40	14	0.9752	0.9752	0.0000
41	15	0.9721	0.9721	0.0000
45	16	0.9828	0.9828	0.0000
46	17	0.9841	0.9841	0.0000
47	18	0.9664	0.9664	0.0000
48	19	0.9664	0.9664	0.0000
49	20	0.9722	0.9722	0.0000
50	21	0.9790	0.9790	0.0000
51	22	0.9793	0.9793	0.0000
52	23	0.9646	0.9646	0.0000
53	24	0.9637	0.9637	0.0000
55	25	0.9602	0.9602	0.0000
56	26	0.9415	0.9415	0.0000
57	27	0.9673	0.9673	0.0000
63	28	0.9899	0.9899	0.0000
64	29	0.9462	0.9462	0.0000
69	30	0.9340	0.9340	0.0000

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0374	0.0001
4	3	-0.0728	-0.0720	0.0009
5	4	-0.0870	-0.0862	0.0009
7	5	-0.1310	-0.1301	0.0009
10	6	-0.1028	-0.1020	0.0009
16	7	-0.1241	-0.1232	0.0009
17	8	-0.1067	-0.1058	0.0009
25	9	-0.1317	-0.1308	0.0009
26	10	-0.1690	-0.1681	0.0009
36	11	-0.0922	-0.0914	0.0009
37	12	-0.1482	-0.1474	0.0009
39	13	-0.1199	-0.1191	0.0009
40	14	-0.1666	-0.1657	0.0009
41	15	-0.1698	-0.1689	0.0009
45	16	-0.1626	-0.1618	0.0009
46	17	-0.1712	-0.1704	0.0009
47	18	-0.1832	-0.1824	0.0009
48	19	-0.1874	-0.1865	0.0009
49	20	-0.1839	-0.1831	0.0009
50	21	-0.1780	-0.1771	0.0009
51	22	-0.1778	-0.1770	0.0009
52	23	-0.1798	-0.1790	0.0009
53	24	-0.1864	-0.1855	0.0009
55	25	-0.1855	-0.1847	0.0009
56	26	-0.1938	-0.1929	0.0009
57	27	-0.1799	-0.1791	0.0009
63	28	-0.1111	-0.1102	0.0009
64	29	-0.2040	-0.2031	0.0009
69	30	-0.2213	-0.2205	0.0009

programming state estimator.

7.5 Enhancements

The NETFLOW program published in reference 70 is very efficient at solving the network flow problem. However its implementation is not ideally suited to the solution of the active and reactive sub-estimation problems. The NETFLO program has been written "efficiently" in Fortran IV and is unfortunately difficult to read. Insufficient time has prevented any major modifications to the code but the following enhancements may significantly improve the method.

The original NETFLO program was written for a single run on a network read in at the start of the program. The program thus has a built in initialisation section which does not cater for accepting the results of a previous solution as a starting point for the present solution. The modification of this initialisation section is likely to reduce the solution times of the second and subsequent solutions of each of the sub-estimation problems by a significant amount. The implementation of a re-start procedure may require some additional storage arrays but in modern computers memory limitations are not usually of concern.

The second major modification which would yield a considerable reduction in the solution times is the removal of the constraint that the flows in the arcs must be uni-directional. This would mean that the pair of arcs currently used to represent each estimate of the power flows would be replaced by just one arc thus reducing the total number of arcs in the network flow problem. The implementation of this enhancement may not be possible in terms of the internal vector which stores the arc flows since the solution method may rely to a significant extent on the values of the vector being non-negative. An alternative method of implementing this enhancement would be to apply a linear transformation to the network flow problem. An arc flow of zero would thus represent a power system flow of -100.0 per unit, an arc flow of $1\ E^7$ a power system flow of zero and an arc flow of $2\ E^7$ a power system flow of 100.0 per unit. This would require that the method of evaluating the cost be changed so that the cost of the flow is evaluated from the magnitude of the difference between the arc flow and $1\ E^7$.

This latter idea could be taken one step further in that the arc flow which corresponds to a cost of zero be specified on entry to the network flow solution subroutine. This would enable the pair of arcs representing a measurement value or a line flow loss value to be replaced by one arc. The cost of the flow on the arc would be arranged such that if the flow agrees

with the measurement value then the cost is zero but if the arc flow deviates from the measurement value in either direction then a cost which is proportional to the magnitude of the deviation is added to the total cost. This would reduce the total number of arcs in either implementation of the network flow technique by a half. The reduction in solution time may well be greater than a half bearing in mind the quadratic relationship between the network flow problem size and the solution times. However, it must be pointed out that the number of nodes remains the same and there is an additional burden associated with calculating the cost of the flow associated with each arc, thus the expected reduction in the solution time is difficult to estimate accurately.

The combination of the two major enhancements discussed above would significantly improve the method which has already been demonstrated as being an attractive method of solving the active and reactive sub-estimation problems.

Chapter 8
Multi-area state estimation

8.1 Introduction to multi-area state estimation

This chapter discusses the utilisation of the 4 stage decomposed linear programming state estimator in a multi-processor computing environment. There are two main reasons why the state estimator may have to run in such an environment. Firstly some computer manufacturers increase the computational throughput of the machine by coupling two or more processors together using a high speed parallel bus. In this configuration the processors are said to be tightly coupled. The memory and other peripherals are also connected to the high speed bus enabling the processors to share access to them. Secondly a large power system may be geographically or otherwise divided into a number of areas each of which has its own control computer. Each computer will have its own memory and peripherals and may communicate with the other computers using media such as private telephone lines. The processors are said to be loosely coupled in this situation. A multi-processor computer, whether it is tightly or loosely coupled will improve the solution time of the state estimator in two ways. The division of the overall task amongst the processors should reduce the solution time in proportion with the inverse of the number of processors. Furthermore the non-linear relationshiup between the solution time and the problem size for the state estimation problem results in a quadratic reduction of the solution time for each of the sub-problems.

However there are additional overheads to be allowed for when operating in a multi-processor environment. The distribution of the state estimator amongst the processors usually requires a program to co-ordinate the activities of each sub-process and in the case of loosely coupled processors considerable delays can occur in transmitting data from one machine to another, especially if large volumes of data are transferred. Wallach, Hancschin and Bongers[140] discuss the expected reduction in the solution time and conclude that in a practical case where sparsity is exploited then the solution time for a multi-processor configuration is obtained by dividing the solution time of the single processor configuration by a value slightly greater than the number of processors.

The conventional state estimation programs were designed to run on a single processor in which each step of the method is performed sequentially. In order to make use of a multi-processor computer the state estimation process needs to be divided into steps which can be performed simultaneously on each of the processors. Generally this approach is implemented by

assigning a section of the network to a state estimation program running in each of the processors. Once the estimates for all the sections have been obtained a master task transfers data between the processors to rationalise the state estimates from each area. The master task may run on any of the processors or in some cases when the master task has a significant computational load it may run on a dedicated processor. Multi-area methods of state estimation are often referred to as hierarchical or two-tier state estimation.

Ideally the implementation of a hierarchical state estimator should minimise the amount of data to be transferred between the processors, especially if they are loosely coupled and use slow speed serial links. The computational load of the master task should be minimised as well since it is likely that the other processors will be idle during this time, thus in effect wasting processing time.

Schweppe, Rom and Wildes[112,113,114] in their pioneering paper have been acknowledged as the first authors to suggest the decomposition of the state estimation problem. A comprehensive survey of numerous hierarchical state estimation methods has been published by Van Cutsem and Ribbens-Parella[138] in which the authors present a brief description of the methods and comment on the performance of the method. Thus no attempt will be made here to present a survey of methods but some of the points relevant to the design of a hierarchical state estimator will be outlined.

The division of a power system into a number of areas can be achieved in two ways. The network may be sectioned by splitting it across a transmission line such that the entire network consists of a number of areas which are inter-connected by tie lines. The tie lines may indeed be real tie lines in the network or they may be transmission lines located at an appropriate point to divide the network. The following authors have investigated this form of sectionalisation: Kobayashi, Narita and Hamman[74], Mukai[95] and Van Cutsem, Horward and Ribbens-Parella.[137] It should be noted that the areas are usually non-overlapping although in some cases the algorithms can be modified to accommodate this situation. The alternative method of dividing the power system involves the sharing of a node between the adjacent areas such that the network consists of a number of areas which overlap at the common nodes. Clements, Denison and Ringlee[26] and Wallach, Handschin and Bongers[140] have reported the use of this method. However, it should be noted that the method of network division used by Wallach et al. is similar to the method of dividing the network across transmission lines, the difference being that the transmission line is included as a tie line in both areas and not treated as a

special section of the network by the master task. The method of Wallach et al. can thus be thought of as dividing the network into a number of areas overlapping on the tie lines. The choice of the method of dividing the network into areas is influenced by factors such as the type of multi-processor computer being used (i.e. loosely or tightly coupled) and thus the data transfer rate between the machines, the number of measurements which may need to be transmitted to the computers assigned to two adjacent areas, the number of estimated values which need to be sent to a master task, the ease of including boundary measurements in the estimation algorithm and the feasibility of bad data analysis at the boundaries.

The second major point to consider when designing a hierarchical state estimation is whether the estimates will be optimal. The term optimal in this context is usually taken to mean whether the state estimates calculated by the hierarchical method are the same as those calculated by a conventional centralised method. The centralised method is assumed to produce optimal estimates because data from the entire network is used in the calculation of all the estimates where as in some hierarchical methods the areas have no knowledge of the network data from the other areas. The sub-optimal hierarchical methods are often implemented in a two step method whereby each area calculates the state estimates for its area and a master task then adjusts only the boundary estimates using the estimates obtained from each area.[26,137,140] In order to obtain optimal estimates the master task must pass the updated boundary estimates back to the areas for further processing, an iterative scheme is required whereby the boundary estimates are passed back and forth between the area tasks and the master task until overall convergence is achieved.[74,95] The sub-optimal, two step hierarchical methods are often prone to suffer from the effects of bad data, especially if the bad data is located near the boundaries,[138] but have the advantage of faster solution times.

8.2 Design of the multi-area state estimator

The design of the multi-area version of the 4 stage decomposed linear programming state estimator was governed by the following design aims. The program had to be implemented with the minimum of programming effort due to time limitations. If the robust nature of the original method was to be retained then the solution must be obtained in an iterative scheme whereby each area has a knowledge of the adjacent areas. It should be noted that in this case it is not necessarily true that the multi-area method of

implementing the 4 stage decomposed linear programming state estimator obtains identical results to the original centralised method. The reason for this is discussed in section 3 of the chapter. The final design aim was that there should be no restrictions imposed upon the choice of the points at which the network was divided and that all available measurements should be included in the estimation process.

The design thus chosen for the initial investigation was as follows. The network would be divided into areas by splitting the network across transmission lines. However each area would have available the measurements (if they existed) of active and reactive power flows at both ends of the tie line and the voltage magnitude measurements (if they existed) at both terminating nodes of the tie line. Thus this method of dividing the network is similar to the method used by Wallach et al.[140] and can be considered as a system composed of areas overlapping over tie lines. In order to transfer information from one area to the other each area would have available to it a set of state estimates calculated by the master task for the active and reactive power flows at both ends of the tie line and also the voltage magnitude and phase angle estimates for both terminating nodes of the tie line. Each area would include these state estimates in the measurement input for the next half of the area/master iteration phase. The state estimates would be treated as reliable measurements and thus assigned a higher weight than normal measurements. The design and operation of the program to calculate the state estimates in each area is discussed in more detail below, followed by a detailed discussion of the master task.

The area task would remain idle until instructed by the master task that an entire state estimation run was required, whereupon the area task would read into local storage a snapshot of the measurement values. The area task would then proceed to calculate the state estimates for its area in the normal iterative way, treating the state estimate input values for the tie lines as very unreliable measurements (the values used being those at the end of the last state estimation run). Once each area task has calculated the state estimates for its area, the estimates for the active and reactive power flows together with the voltage magnitude and phase angle estimates for both ends of all the tie lines associated with that area are passed to the master task. The area task will remain idle until the master task returns an updated set of tie line estimates to it. The area task will treat those updated estimates as reliable measurements and assign a higher weight to them. The area task commences the usual iterative estimation method using as input the original measurement values read at the start of the estimation run and the updated tie

line estimates obtained from the master task. The choice of the weights assigned to the updated estimates is discussed later in this section as is the method of passing data between the master and area tasks. The iterative process of passing the state estimates back and forth between the master task and the area tasks is continued until the master task ascertains that each of the area tasks has converged. The area tasks have converged when there is no significant difference between the value of the updated tie line estimates supplied by the master task and the value of the tie line estimates calculated by the area task.

In order to calculate the phase angle estimates in each area with respect to the system reference node, each area, except the area containing the phase angle reference node, has been assigned a tie line termination node to use as its phase angle reference point. The following should be noted on the subject of phase angle reference points, firstly a tie line may only be used as the reference point for one area and that some logic is required in the master task to ensure that a path exists from the system reference node across the tie lines to all the areas. It should be noted that the tie line termination node used as a phase angle reference node is the node at the end of tie line which is embedded in the area. Initially the master task only alters the phase angle estimates for these selected reference nodes in order that value of the phase angle at the tie line reference node is obtained with respect to the adjacent area and consequently the system reference node. When convergence approaches the master task then updates all the tie line phase angle estimates to smooth the estimates across the network. The weights on all the tie line phase angle estimates are initially set low except for the chosen tie line reference node, however once convergence approaches all the tie line phase angle weights are adjusted in accordance with other tie line estimates as described later in this section.

The master task periodically checks to see if the operator has requested a state estimation run or the time interval since the last run has expired. If either of these conditions are satisfied the master task instructs the area tasks to commence an estimation run. The master task will then remain idle until each area task has transferred its tie line estimates. The master task then calculates the values of the updated tie line estimates from the estimates obtained from the two areas inter-connected by the tie line. This process is itself an estimation problem and is solved by treating each tie line in turn as a power system with two nodes and one transmission line. The measurements for this two node system are the eight estimates of the power flows and voltage levels obtained from the two areas. The master task thus

uses the same methodology as the area tasks to calculate the updated tie line estimates.

Originally the master task used the same linear programming code to solve the linear set of equations formed for each of the 4 stages of this 2 node power system state estimation. However when the problem is analysed in detail each set of linear equations has only five equations from which to calculate two estimates. The two estimates being the updated tie line estimates of the two ends of the tie line for whichever stage of the 4 stage estimation process was being considered at the time. The five equations consisted of two measurement equations equating the updated tie line estimates to the tie line estimates calculated by one area, a second pair of similar equations for the tie line estimates from the other area and an equation equating the differences between the updated tie line estimates to a value calculated from the most recent updated tie line estimates from the other 3 stages of the 4 stage estimation process. The linear programming solution of the five equations had a tendency to accept the estimates from one area and totally reject the estimates from the other area. This caused convergence problems in some cases where an area has multiple tie lines because the area is unable to accept the updated estimates from one tie line which were calculated from values supplied by the adjacent area. Since the master task is processing values which are in themselves state estimates and ought to have had any gross errors removed, it would thus seem appropriate to smooth the values. The five linear equations are therefore solved using a least squares method which results in the updated estimates reflecting the estimates supplied by each area.

The master task returns the updated tie line estimates to all the areas which then compare the values with those sent to the master task. Each area task will transmit a message to the master indicating whether it has converged. If all the area tasks have converged then the master task instructs the area tasks to output the results and enter a wait state, otherwise the area tasks are instructed to process the updated tie line estimates.

The choice of the weights to use in the area tasks for the updated tie line estimates has a significant effect on the performance of the entire process. The initial logic for the tie line phase angle estimates has been explained earlier in this section and it has been mentioned that on the first iteration the weights of the other estimates are set very low in order that these estimates do not influence the solution point of the area task during this iteration. As convergence approaches the weights have to be increased

above those of ordinary measurements to force the area task to accept the updated tie line estimates as being correct and adjust the local estimates accordingly. However the weights must not be increased too high because if an updated tie line estimat is in error then the area task will not be able to reject it, resulting in erroneous state estimates. The weights must not be increased too soon because if the process is not close to convergence then a large weight on the updated tie line estimates may have a detrimental effect on the rate of convergence. Conversely if after several iterations the area task does not agree with the updated tie line estimates and no change occurs from one iteration to the next then the weights ought to be increased to force the area task to accept the updated tie line estimate.

Numerous schemes of setting the weights for the updated tie line estimates were implemented. The first scheme tried was to set the weights of the updated tie line estimates to twice those of the normal measurements weights on the second and successive iterations. The most complex scheme implemented involved incrementing the weights over three iterations to twice the value of the measurement weights and including a check whereby if, after three iterations, there was still disagreement between the tie line estimate calculated by the area task and the updated value returned by the master task and no change in the estimates occured from one iteration to the next task, then the weight of that tie line estimate was increased still further. However once the estimates calculated by the area task had changed the weight was returned to its normal upper level. This is to ensure that the area task does not accept the updated tie line estimate by forming an isolated node. The isolated node can be formed by the linear programming method rejecting all the transmission line difference equations for all the lines connected to the tie line termination node. This situation would result in only the updated tie line estimate measurement equation being eligible to define the estimates at that point.

This scheme showed no better results than a less sophisticated one in which the weights were gradually increased over six iterations to a value approximately three times the normal measurement weight. It was found that the tie line estimates for the remote end of the tie line did not contribute to estimation process and could have been omitted from the problem. Furthermore the weights on the updated tie estimates for the remote end of the tie line could not be increased above twice the normal measurement value. This is because the area task only has one transmission line difference equation to augment the updated tie line estimates at the remote termination node and the linear programming method would accept the updated tie line

estimates in any event if the weights were too high.

The data transfer between the area tasks and the master task was performed using a task common block. Each tie line had a total of sixteen estimates transferred between the master task and two area tasks. Both the area tasks and the master task were aware of the location of the estimates in the task common block for the tie lines relevant to the task (all the tie lines in the case of the master task). Each area task also had a set of flags used to co-ordinate the sequence of events and to transfer information on the state of convergence. The implementation of the method limited the number of areas to 5 and the number of tie lines to 15. This meant the total size of the task common block was less than 1k bytes.

It was initially intended to run the programs on a multi-processor system which have consisted of three Perkin-Elmer 3220 computers and a Perkin-Elmer 3230 computer. The computers would have been inter-connected by 9.6k baud serial links and a user transparent program would have maintained copies of all the task common blocks in all of the computers. However the data transfer package was not completed in time and the programs were all run on the Perkin-Elmer 3230 in a time sharing environment. This enabled an estimation of typical execution times (excluding the data transfer times) and a detailed study of the performance of the programs in terms of the accuracy of state estimates to be made.

To facilitate the operation of the programs, a series of data files were prepared which assigned the substations of the test network to a specific area. Upon the loading of the area tasks, each area task would read a data file to ascertain which substations it was to process. The area tasks then searched the data base stored in the task common blocks to determine which power system elements and measurement values belonged to the area and which transmission lines formed the inter-area tie lines. Details of the test networks used and the results of the programs are presented in the next section.

8.3. Results

The distributed version of the four stage decomposed linear programming state estimator was tested on a single Perkin-Elmer 3230 mini computer using a time sharing operating system. The IEEE 30 substation test network and the IEEE 118 substation test network described in chapter 3 were used for testing the programs. Each of the networks was divided into a number of different configurations. The choice of which transmission lines to use as the

inter-area tie lines was fairly random. However the following points were considered when making the choice. The number of substations in each area was to be similar, substations which appeared to be geographically close to each other were placed in the same group, however a transformer could be used as a tie line in order to group substations at different voltage levels in one area, the number of tie lines terminating at a single substation was varied to produce a wider selection of operating conditions and in the case where the network was divided into four areas whether the fourth area was connected by a tie line to the area containing the phase angle reference point.

The IEEE 30 substation test network is really a little small to split into many areas, but it was useful for use in debugging the programs. Therefore the IEEE 30 substation test network was divided into four different configurations as illustrated in figure 8.1. The transmission lines with dashed lines across them were selected to be the tie lines. The letters at the ends of the dashed lines indicate that the transmission line was used as a tie line in that configuration. Configuration A divided the network into two areas of 14 and 16 substations respectively with one substation in the first half of the network being the termination point for four tie lines. Configuration B divided the system into three areas, the number of substations in each area was 9, 7 and 14. It should be noted that this division has been used previously[64] and was thus chosen for comparison purposes. Configurations C and D divided the network into four areas, each of which contained between seven and eight substations. The main difference being that configuration D has three tie lines terminating at substation 6 while configuration C has four.

The IEEE 118 substation test network was divided into a total of seven different configurations as illustrated in figure 8.2. Configuration A divided the network into two halves of 70 and 48 substations using only four tie lines. Configurations B and C both divide the network into two areas of 57 and 61 substations, using 8 and 7 tie lines respectively, the main difference between the two configurations apart from the location of the tie lines is that configuration B has two tie termination points with a pair of tie lines connected to it whereas configuration C has only one such point. Configurations D and E divide the network into three areas, configuration D being a more even division with 41, 38 and 39 substations in each area and requiring ten tie lines, whereas configuration C requires only seven tie lines but has 36, 34 and 48 substations in each area. Configuratiuon D has three points at which a pair of tie lines terminate. Finally configurations F and G divided the network into four areas, both of which required 15 tie lines and

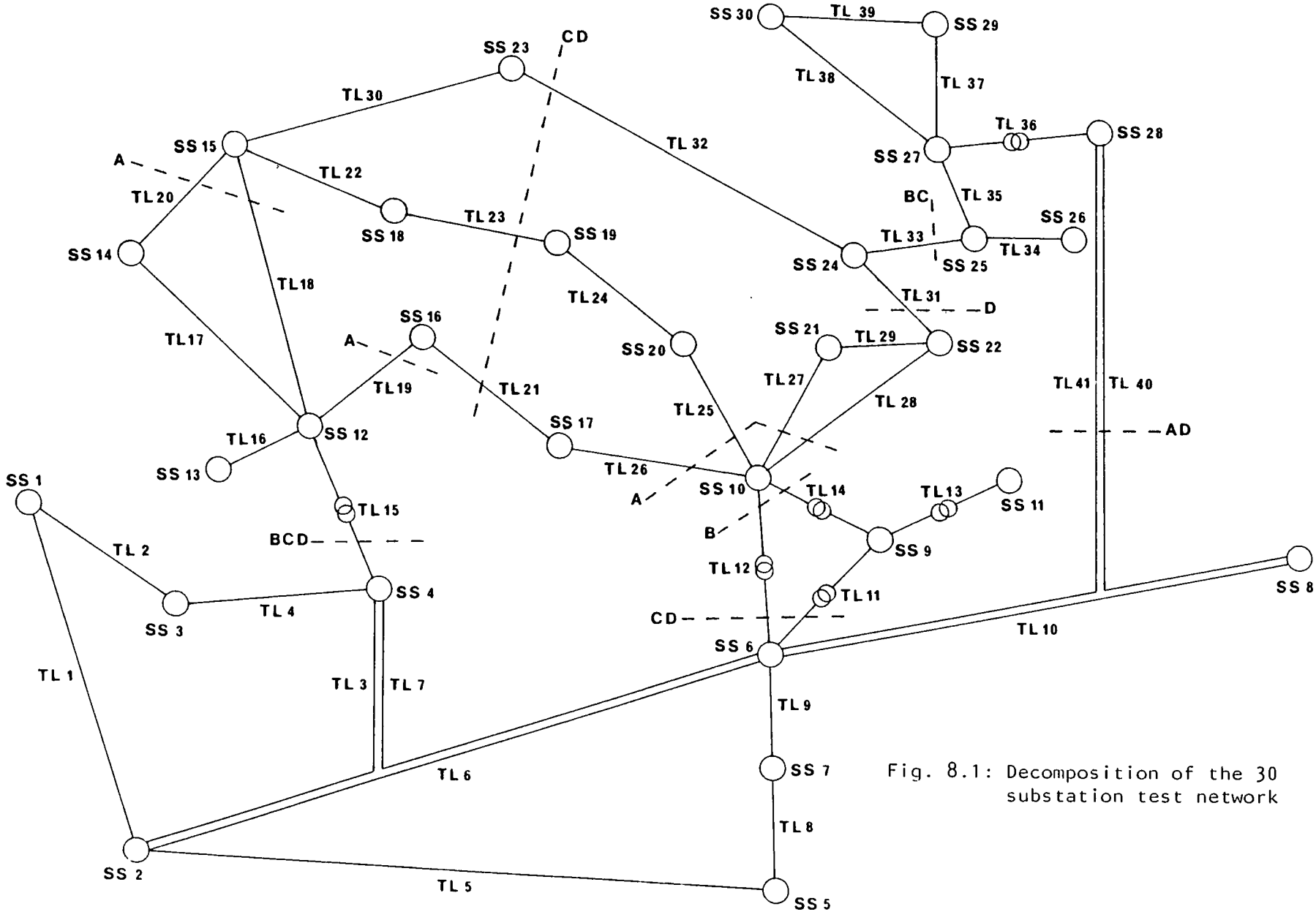


Fig. 8.1: Decomposition of the 30 substation test network

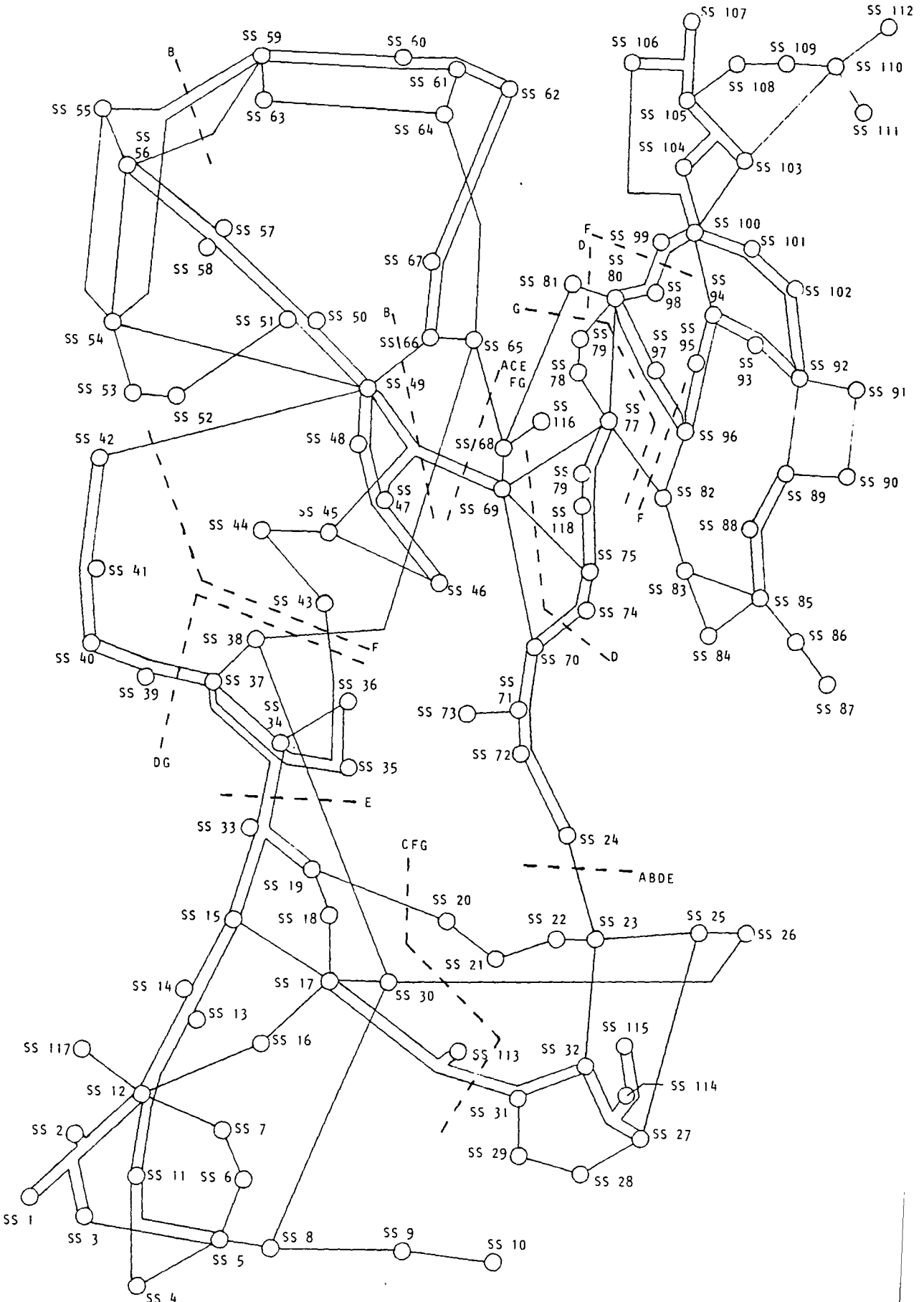


Fig. 8.2: Decomposition of the 118 substation test network

have no tie lines inter-connecting the first and fourth areas. Both the F and G configurations have a fairly even division of substations amongst the areas, the numbers being 32, 25, 32 and 29 for configuration F and 28, 29, 28 and 33 for configuration G. The former configuration had two points at which a pair of tie lines terminated, the latter had three such points.

The format for testing the distributed four stage decomposed linear programming state estimation method was identical to that used to evaluate the centralised method as described in chapter 6. Thus the severe measurement errors applied to the 30 substation test network are the same as those described at the start of section 6.4.3 and the sequence of events used to time the programs are those listed in table 6.4 for the 30 substation test network and in table 6.19 for the 118 substation test network. To summarise, the sequence of events for timing purposes are as follows:

1. Start the estimator.
2. Force the estimator to run by an operator request.
3. Open a transmission line (No Line number 7 for the 30 substation test network, line number 50 for the 118 substation test network).
4. Open link number 6 for the 30 substation test network or load number 7 for the 118 substation test network.
5. Close all the switches.
6. Update the measurement values if using noisy measurements.

The results presented in tabular form have been divided as follows. Tables 8.1 to 8.3 present the state estimate values for three different operating environments on the 30 substation test network. Tables 8.4 to 8.7 present the results of timing runs on the 30 substation test network. Tables 8.8 to 8.14 present the results of timing runs on the 118 substation test network. Only a limited number of state estimated results have been presented because the method was generally unreliable in its present form. One of the principal reasons for implementing a distributed version of the program was to reduce the overall solution times without sacrificing the reliability of the state estimates. The method fails to converge in many cases when operating on noisy measurements, thus a detailed study of the reliability of the method in the presence of gross measurement errors and switch status errors was postponed and subsequently not completed until the method was reliable when operating in the realistic environment of noisy measurements. A further discussion on the causes of the convergence problems is presented after a brief discussion of the state estimate results presented in tables 8.1 to 8.3.

A tabular set of state estimate results has not been produced for the case when the measurement values are subject to no noise or gross errors for either the 30 or 118 substation test networks because the programs converge with no errors at all on the state estimate results. Table 6.5 lists the estimates for the centralised method on the 30 substation test network when the measurements are noise and error free. Table 8.1 presents the state estimate results for the 30 substation test network when the measurement set has had the eight gross errors applied to it and the network has been divided into two areas (configuration A in figure 8.1). The eight gross errors are detailed at the start of section 6.4.3 and the results should be compared with those of table 6.8 which lists the results for the centralised method. It is interesting to note that line number 19 which is selected as a tie line in this case terminates at substation 12 where the active power flow measurement for the load is 30% too high and also at substation 16 where the program produces voltage estimates which are slightly in error. The voltage magnitude error of 0.0011 PU. is to be compared with a voltage magnitude error of 0.0032 PU. at substation 13 in the centralised case. The division of the network has shifted the state estimate error arising from the erroneous load measurement from substation 13 to a terminating node of a tie line. A similar shift in the location of the phase angle estimate error has also occurred. The division of the network into three and four areas did not reproduce this shift thus a preliminary conclusion to be drawn from this observation is that the division of the network is likely to affect the performance of the method in terms of whether the state estimates are optimal or not. A much more detailed study of the effects of bad data on the state estimates using many different sets of corrupt measurements would be required to ascertain the reliability of the method in the presence of gross measurements.

Tables 8.2 and 8.3 present the state estimate results for the 30 substation test network when the measurements were subject to the addition of 0.2% systematic and 1.5% random noise. The results in these tables are to be compared with those of the centralised method listed in table 6.6. It should be noted that the results in table 8.2 are for the case when the network was divided into two areas (configuration A in figure 8.1) while the results in table 8.3 are for the case when the network was divided into four areas (configuration C in figure 8.1). The method failed to converge in the latter case, that is after 20 iterations one or more of the area tasks failed to accept the updated tie line estimates on one or more of the tie lines as being valid. In this situation the process is halted and the state estimate values are written to memory as in the case when convergence is achieved.

Table 8.1: Estimates from the distributed state estimation program on the 30 substation test network split into 2 areas with 8 severely corrupted analogue measurements

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0438	0.0000
2	2	1.0301	1.0301	0.0000
4	3	1.0105	1.0105	0.0000
5	4	1.0025	1.0025	0.0000
7	5	0.9851	0.9851	0.0000
10	6	0.9951	0.9951	0.0000
16	7	0.9829	0.9829	0.0000
17	8	0.9894	0.9894	0.0000
25	9	1.0071	1.0071	0.0000
26	10	0.9928	0.9928	0.0000
36	11	1.0486	1.0486	0.0000
37	12	0.9889	0.9889	0.0000
39	13	1.0011	1.0011	0.0000
40	14	0.9752	0.9752	0.0000
41	15	0.9721	0.9721	0.0000
45	16	0.9828	0.9839	0.0011
46	17	0.9841	0.9847	0.0006
47	18	0.9664	0.9664	0.0000
48	19	0.9664	0.9664	0.0000
49	20	0.9722	0.9722	0.0000
50	21	0.9790	0.9790	0.0000
51	22	0.9793	0.9793	0.0000
52	23	0.9646	0.9646	0.0000
53	24	0.9637	0.9637	0.0000
55	25	0.9602	0.9602	0.0000
56	26	0.9415	0.9415	0.0000
57	27	0.9673	0.9673	0.0000
63	28	0.9899	0.9899	0.0000
64	29	0.9462	0.9462	0.0000
69	30	0.9340	0.9340	0.0000

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0375	0.0000
4	3	-0.0728	-0.0728	0.0000
5	4	-0.0870	-0.0870	0.0000
7	5	-0.1310	-0.1310	0.0000
10	6	-0.1028	-0.1028	0.0000
16	7	-0.1241	-0.1241	0.0000
17	8	-0.1067	-0.1067	0.0000
25	9	-0.1317	-0.1317	0.0000
26	10	-0.1690	-0.1690	0.0000
36	11	-0.0922	-0.0922	0.0000
37	12	-0.1482	-0.1482	0.0000
39	13	-0.1199	-0.1199	0.0000
40	14	-0.1666	-0.1666	0.0000
41	15	-0.1698	-0.1698	0.0000
45	16	-0.1626	-0.1632	-0.0005
46	17	-0.1712	-0.1717	-0.0005
47	18	-0.1832	-0.1832	0.0000
48	19	-0.1874	-0.1874	0.0000
49	20	-0.1839	-0.1839	0.0000
50	21	-0.1780	-0.1780	0.0000
51	22	-0.1778	-0.1778	0.0000
52	23	-0.1798	-0.1798	0.0000
53	24	-0.1864	-0.1864	0.0000
55	25	-0.1855	-0.1855	0.0000
56	26	-0.1938	-0.1938	0.0000
57	27	-0.1799	-0.1799	0.0000
63	28	-0.1111	-0.1111	0.0000
64	29	-0.2040	-0.2040	0.0000
69	30	-0.2213	-0.2213	0.0000

Table 8.2: Estimates from the distributed state estimation program on the 30 substation test network split into 2 areas with 0.2% systematic noise and 1.5% random noise

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0424	-0.0014
2	2	1.0301	1.0288	-0.0013
4	3	1.0105	1.0092	-0.0014
5	4	1.0025	1.0012	-0.0014
7	5	0.9851	0.9838	-0.0013
10	6	0.9951	0.9937	-0.0014
16	7	0.9829	0.9814	-0.0014
17	8	0.9894	0.9880	-0.0014
25	9	1.0071	1.0058	-0.0013
26	10	0.9928	0.9914	-0.0014
36	11	1.0486	1.0479	-0.0007
37	12	0.9889	0.9876	-0.0014
39	13	1.0011	0.9998	-0.0013
40	14	0.9752	0.9737	-0.0014
41	15	0.9721	0.9707	-0.0014
45	16	0.9828	0.9814	-0.0014
46	17	0.9841	0.9827	-0.0014
47	18	0.9664	0.9649	-0.0016
48	19	0.9664	0.9648	-0.0017
49	20	0.9722	0.9705	-0.0017
50	21	0.9790	0.9775	-0.0015
51	22	0.9793	0.9778	-0.0015
52	23	0.9646	0.9632	-0.0014
53	24	0.9637	0.9621	-0.0015
55	25	0.9602	0.9588	-0.0014
56	26	0.9415	0.9397	-0.0018
57	27	0.9673	0.9661	-0.0012
63	28	0.9899	0.9887	-0.0012
64	29	0.9462	0.9445	-0.0017
69	30	0.9340	0.9324	-0.0016

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0371	0.0004
4	3	-0.0728	-0.0734	-0.0006
5	4	-0.0870	-0.0877	-0.0007
7	5	-0.1310	-0.1325	-0.0014
10	6	-0.1028	-0.1042	-0.0014
16	7	-0.1241	-0.1256	-0.0015
17	8	-0.1067	-0.1081	-0.0014
25	9	-0.1317	-0.1329	-0.0012
26	10	-0.1690	-0.1718	-0.0028
36	11	-0.0922	-0.0941	-0.0019
37	12	-0.1482	-0.1504	-0.0022
39	13	-0.1199	-0.1221	-0.0022
40	14	-0.1666	-0.1690	-0.0024
41	15	-0.1698	-0.1722	-0.0024
45	16	-0.1626	-0.1649	-0.0023
46	17	-0.1712	-0.1736	-0.0023
47	18	-0.1832	-0.1855	-0.0023
48	19	-0.1874	-0.1900	-0.0026
49	20	-0.1839	-0.1865	-0.0026
50	21	-0.1780	-0.1805	-0.0025
51	22	-0.1778	-0.1804	-0.0025
52	23	-0.1798	-0.1822	-0.0024
53	24	-0.1864	-0.1890	-0.0026
55	25	-0.1855	-0.1882	-0.0027
56	26	-0.1938	-0.1970	-0.0032
57	27	-0.1799	-0.1824	-0.0025
63	28	-0.1111	-0.1126	-0.0015
64	29	-0.2040	-0.2064	-0.0024
69	30	-0.2213	-0.2236	-0.0022

Table 8.3: Estimates from the distributed state estimation program on the 30 substation test network split into 4 areas (lines cut at C) with 0.2% systematic noise and 1.5% random noise

Values are in P.U.
Error = Estimate - True value

Voltage magnitude estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	1.0438	1.0398	-0.0040
2	2	1.0301	1.0262	-0.0039
4	3	1.0105	1.0065	-0.0040
5	4	1.0025	0.9985	-0.0041
7	5	0.9851	0.9825	-0.0026
10	6	0.9951	0.9924	-0.0027
16	7	0.9829	0.9801	-0.0027
17	8	0.9894	0.9867	-0.0027
25	9	1.0071	1.0045	-0.0025
26	10	0.9928	0.9898	-0.0030
36	11	1.0486	1.0466	-0.0020
37	12	0.9889	0.9847	-0.0042
39	13	1.0011	0.9970	-0.0041
40	14	0.9752	0.9708	-0.0044
41	15	0.9721	0.9676	-0.0045
45	16	0.9828	0.9783	-0.0045
46	17	0.9841	0.9796	-0.0045
47	18	0.9664	0.9619	-0.0045
48	19	0.9664	0.9618	-0.0046
49	20	0.9722	0.9676	-0.0046
50	21	0.9790	0.9758	-0.0031
51	22	0.9793	0.9761	-0.0031
52	23	0.9646	0.9601	-0.0045
53	24	0.9637	0.9592	-0.0045
55	25	0.9602	0.9559	-0.0044
56	26	0.9415	0.9367	-0.0048
57	27	0.9673	0.9643	-0.0030
63	28	0.9899	0.9871	-0.0028
64	29	0.9462	0.9427	-0.0035
69	30	0.9340	0.9306	-0.0034

Voltage phase angle estimates

Bus No.	Node No.	True value	Estimate	Error
1	1	0.0000	0.0000	0.0000
2	2	-0.0375	-0.0359	0.0016
4	3	-0.0728	-0.0712	0.0016
5	4	-0.0870	-0.0856	0.0014
7	5	-0.1310	-0.1310	0.0000
10	6	-0.1028	-0.1027	0.0002
16	7	-0.1241	-0.1241	0.0000
17	8	-0.1067	-0.1065	0.0002
25	9	-0.1317	-0.1316	0.0001
26	10	-0.1690	-0.1687	0.0003
36	11	-0.0922	-0.0925	-0.0002
37	12	-0.1482	-0.1476	0.0006
39	13	-0.1199	-0.1191	0.0008
40	14	-0.1666	-0.1662	0.0003
41	15	-0.1698	-0.1694	0.0004
45	16	-0.1626	-0.1621	0.0005
46	17	-0.1712	-0.1708	0.0004
47	18	-0.1832	-0.1826	0.0006
48	19	-0.1874	-0.1872	0.0002
49	20	-0.1839	-0.1837	0.0002
50	21	-0.1780	-0.1775	0.0004
51	22	-0.1778	-0.1774	0.0004
52	23	-0.1798	-0.1796	0.0003
53	24	-0.1864	-0.1862	0.0002
55	25	-0.1855	-0.1854	0.0001
56	26	-0.1938	-0.1943	-0.0005
57	27	-0.1799	-0.1796	0.0003
63	28	-0.1111	-0.1110	0.0001
64	29	-0.2040	-0.2037	0.0003
69	30	-0.2213	-0.2209	0.0005

The results shown in tables 8.2 and 8.3 are encouraging in that the magnitudes of the errors on the estimates are less than those in table 6.6. It is likely that the improvement can be attributed to the least squares smoothing effect which results from the processing of the tie line estimates by the master task. The magnitudes of the errors on the voltage phase angle estimates in table 8.3 are small in comparison with those in tables 6.6 and 8.2 and do not exhibit the general trend of increasing in magnitude as the number of transmission lines increases between the substation containing the reference point and the estimation point. This arises because of the failure of the method to converge. As has been explained each area task selects a tie line termination node to be reference point for that area if the area does not contain the system reference point. The voltage phase angle estimates are evaluated with respect to this node and without regard to any of the other tie line termination nodes until convergence approaches and in this case convergence has not been achieved thus the phase angle estimates in each area have been evaluated with respect to a single node rather than with consideration to all the nodes as in the centralised method. However it should be noted that the least squares smoothing effect still occurs because the master task evaluates the voltage phase angle difference across the tie lines using the least squares method and then subtracts this difference from the estimated value supplied by the area task containing the system reference point. It would not be a good idea to adopt a method of evaluating the voltage phase angle estimates by selecting a single tie line termination node to be the be the reference point as opposed to the method of averaging across all the tie line termination nodes because any bad data adjacent to the selected reference point would lead to erroneous estimates.

The timing results presented in tables 8.4 to 8.7 have been obtained from using the 30 substation test network divided into the four different configurations illustrated in figure 8.1. As in the case of the centralised method the programs were run through a set of sequence of events using measurements which were both noise and gross error free and then using measurements subject to the addition of 0.2% systematic noise and 1.5% random noise. Each table has two halves, the upper half lists the results on the noise and gross error free measurements, the lower half lists the results obtained using the noisy measurements. The tasks were all run in a time sharing environment in a single processor and the CPU time used by the master task and each of the area tasks is listed together with the number of iterations the master task required to obtain overall convergence. The number of iterations is defined as the number of times updated tie line estimates are

Table 8.4: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 30 substation test network split into 2 areas

The tie lines are indicated by the letter A in figure 8.1. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	72.37 (5)	15.00 (1)	30.71 (3)	37.39 (3)	41.16 (4)
Master	0.38 (2)	0.12 (1)	0.47 (3)	0.09 (1)	0.48 (1)
Area 1	39.00	7.09	34.51	5.05	35.81
Area 2	28.92	10.44	18.71	3.05	24.26
Total distributed	68.30	17.65	53.69	8.19	60.55
Elapsed distributed	72	17	59	7	68

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	74.08 (4)	4.70 (1)	31.35 (3)	21.28 (2)	29.95 (3)	28.21 (3)
Master	0.89 (4)	0.12 (1)	0.81 (4)	0.76 (4)	0.78 (4)	0.68 (4)
Area 1	52.40	3.18	42.38	33.38	36.48	31.96
Area 2	35.11	1.88	23.94	19.31	22.40	22.15
Total distributed	88.40	5.18	67.13	53.45	59.66	54.79
Elapsed distributed	100	5	77	57	69	62

Table 8.5: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 30 substation test network split into 3 areas

The tie lines are indicated by the letter B in figure 8.1. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	72.37 (5)	15.00 (1)	30.71 (3)	37.39 (3)	41.16 (4)
Master	0.21 (2)	0.07 (1)	0.24 (1)	0.19 (1)	0.22 (2)
Area 1	14.53	2.86	11.07	1.85	11.58
Area 2	15.89	2.23	9.43	2.03	10.59
Area 3	21.87	2.40	11.57	1.67	15.18
Total distributed	52.50	7.56	32.31	5.74	37.57
Elapsed distributed	54	7	34	5	39

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	74.08 (4)	4.70 (1)	31.35 (3)	21.28 (2)	29.95 (3)	28.21 (3)
Master	0.88 (7)	0.09 (1)	2.19 (>20)	1.07 (9)	0.89 (7)	0.65 (5)
Area 1	23.84	1.30	43.22	21.57	22.64	14.27
Area 2	25.29	1.34	37.17	19.04	17.64	14.96
Area 3	30.19	1.33	39.15	20.27	20.25	15.47
Total distributed	80.20	4.06	122.73	61.95	61.42	45.35
Elapsed distributed	89	4	142	71	71	51

Table 8.6: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 30 substation test network split into 4 areas

The tie lines are indicated by the letter C in figure 8.1. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	72.37 (5)	15.00 (1)	30.71 (3)	37.39 (3)	41.16 (4)
Master	0.28 (2)	0.07 (1)	0.40 (3)	0.11 (1)	0.15 (2)
Area 1	12.47	2.52	12.60	2.82	11.18
Area 2	7.12	2.04	5.11	1.25	5.69
Area 3	12.92	2.38	9.61	1.22	8.77
Area 4	15.95	2.23	11.23	2.28	10.38
Total distributed	48.74	9.24	38.95	7.68	36.17
Elapsed distributed	49	9	41	7	37

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	74.08 (4)	4.70 (1)	31.35 (3)	21.28 (2)	29.95 (3)	28.21 (3)
Master	2.47 (>20)	1.35 (10)	1.76 (12)	1.06 (8)	1.21 (8)	1.41 (10)
Area 1	39.71	18.70	28.26	19.35	21.44	21.59
Area 2	21.45	10.03	14.93	11.30	10.02	11.60
Area 3	34.28	16.96	21.58	15.50	16.47	21.08
Area 4	45.34	18.01	24.25	17.03	20.60	18.63
Total distributed	143.25	64.05	90.78	64.24	69.54	74.31
Elapsed distributed	157	68	96	67	72	92

Table 8.7: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 30 substation test network split into 4 areas

The tie lines are indicated by the letter D in figure 8.1. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	72.37 (5)	15.00 (1)	30.71 (3)	37.39 (3)	41.16 (4)
Master	0.41 (3)	0.07 (1)	0.40 (3)	0.10 (1)	0.35 (3)
Area 1	23.84	2.55	17.79	2.67	18.12
Area 2	10.85	1.14	6.46	0.94	7.36
Area 3	9.15	1.23	5.61	0.74	6.38
Area 4	16.62	1.79	11.32	1.77	11.58
Total distributed	60.86	6.78	31.58	6.22	43.79
Elapsed distributed	63	7	43	6	46

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	74.08 (4)	4.70 (1)	31.35 (3)	21.28 (2)	29.95 (3)	28.21 (3)
Master	1.16 (7)	0.10 (1)	2.76 (>20)	2.64 (>20)	1.05 (6)	2.45 (>20)
Area 1	32.75	1.81	54.83	45.00	27.80	43.81
Area 2	17.40	0.83	27.43	22.56	12.52	22.94
Area 3	12.41	0.60	21.57	18.78	10.11	18.11
Area 4	20.22	1.07	30.74	27.78	13.79	27.43
Total distributed	83.94	4.41	137.33	116.76	65.27	114.74
Elapsed distributed	89	4	149	127	68	124

returned to the area tasks from the master task. It should be noted that in the cases when the number of iterations is greater than 20 the method is deemed to have failed to converge. The CPU time and number of iterations used by the centralised method is included at the top of each half of the table for comparison purposes. The total CPU time and the elapsed time for the distributed method are listed at the bottom of each half of the table. The total CPU time is the sum of all the CPU times of the master task and the area tasks. The elapsed time is equivalent to the time obtained by using a stopwatch to time the state estimation process from start to finish and is thus greater than the total CPU time in the cases when one or more area tasks are idling while waiting for the other area tasks to converge. It should be noted that the elapsed times were obtained by calls to the operating system time routine from the master task at the start and finish of an estimation run. The time routine returns the time with an accuracy of plus or minus one second. Thus in some cases when the solution was obtained very quickly the difference between the start and finish times, the elapsed time, is smaller than the total CPU time. Tables 8.8 to 8.14 present the timing results for the seven different configurations of the 118 substation test network as illustrated in figure 8.2.

The discussion on the timing results will be divided into two halves, a discussion on the results when the measurements have had no noise applied to them and a discussion on the results when noise is applied.

The method has no problems in converging when no noise is applied to the measurements. When operating on the 30 substation test network the master task usually converges in one or two iterations. The total CPU used by the distributed method is similar to the centralised method in cases when a line switch either opens or closes, as illustrated by event numbers 3 and 5 in tables 8.4 to 8.7. However the tripping of a link switch results in a fast solution time because the areas which do not contain the link are already at the solution point and only a small adjustments are needed to the active and reactive power flow estimates in the area containing the link. This is illustrated by the times for event number 4 in the tables 8.4 to 8.7. The total CPU time for the distributed method in first state estimation run, event 1, was always less than the CPU time for the centralised method and in several cases the elapsed time of the distributed method is less than the CPU time of the centralised method.

It would be expected that the ratio of the number of substations in each area would affect the ratio of the elapsed time to the total CPU time. If the division of the network resulted in one area containing substantially more

Table 8.8: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 118 substation test network split into 2 areas

The tie lines are indicated by the letter A in figure 8.2. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	416.08 (6)	32.63 (1)	161.70 (4)	124.83 (3)	190.64 (4)
Master	0.49 (2)	0.07 (1)	0.17 (1)	0.38 (3)	0.36 (3)
Area 1	178.94	18.60	69.33	77.07	99.70
Area 2	99.97	6.78	32.59	51.68	55.39
Total distributed	279.40	25.45	102.09	129.13	155.45
Elapsed distributed	296	25	103	148	160

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	441.93 (6)	7.27 (1)	109.29 (3)	101.63 (3)	105.18 (3)	94.47 (3)
Master	0.91 (6)	0.08 (1)	0.82 (7)	0.65 (5)	0.61 (5)	2.00 (>20)
Area 1	185.07	4.07	142.16	90.16	88.55	156.97
Area 2	109.60	3.12	98.25	44.31	43.73	98.82
Total distributed	295.58	7.27	241.23	135.12	132.89	257.79
Elapsed distributed	304	7	167	140	137	272

Table 8.9: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 118 substation test network split into 2 areas

The tie lines are indicated by the letter B in figure 8.2. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	416.08 (6)	32.63 (1)	161.70 (4)	124.83 (3)	190.64 (4)
Master	0.62 (2)	0.11 (1)	0.29 (1)	0.59 (3)	0.54 (3)
Area 1	153.94	13.87	51.50	63.77	109.12
Area 2	139.95	17.09	52.69	86.30	99.59
Total distributed	294.51	31.07	104.48	150.66	209.25
Elapsed distributed	302	32	106	158	215

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	441.93 (6)	7.27 (1)	109.29 (3)	101.63 (3)	105.18 (3)	94.47 (3)
Master	3.19 (>20)	2.18 (>20)	1.28 (7)	1.65 (9)	2.98 (19)	1.03 (5)
Area 1	205.27	81.95	94.53	109.26	140.37	84.92
Area 2	216.99	87.89	82.69	110.49	124.92	78.70
Total distributed	425.45	172.02	178.50	221.40	268.27	164.65
Elapsed distributed	443	185	185	209	283	169

Table 8.10: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 118 substation test network split into 2 areas

The tie lines are indicated by the letter C in figure 8.2. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	416.08 (6)	32.63 (1)	161.70 (4)	124.83 (3)	190.64 (4)
Master	0.62 (2)	0.10 (1)	0.53 (3)	0.83 (5)	0.58 (3)
Area 1	143.26	17.27	86.68	130.72	108.05
Area 2	147.78	12.91	72.23	109.86	97.92
Total distributed	291.66	30.28	159.44	241.41	206.55
Elapsed distributed	298	30	164	247	214

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	441.93 (6)	7.27 (1)	109.29 (3)	101.63 (3)	105.18 (3)	94.47 (3)
Master	1.43 (7)	0.09 (1)	1.21 (7)	0.93 (5)	1.05 (6)	2.69 (>20)
Area 1	161.93	3.84	92.09	85.89	96.16	143.26
Area 2	166.33	4.39	82.16	69.36	83.75	127.56
Total distributed	329.69	8.32	175.46	156.19	180.96	273.51
Elapsed distributed	342	9	184	164	190	292

Table 8.11: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 118 substation test network split into 3 areas

The tie lines are indicated by the letter D in figure 8.2. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	416.08 (6)	32.63 (1)	161.70 (4)	124.83 (3)	190.64 (4)
Master	0.89 (6)	0.11 (1)	0.51 (3)	0.61 (4)	0.83 (5)
Area 1	139.79	11.45	47.49	44.19	62.87
Area 2	141.18	8.66	64.70	57.60	84.04
Area 3	101.64	5.50	35.92	42.54	54.56
Total distributed	383.50	25.72	148.62	144.94	202.30
Elapsed distributed	391	25	151	148	239

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	441.93 (6)	7.27 (1)	109.29 (3)	101.63 (3)	105.18 (3)	94.47 (3)
Master	1.75 (10)	0.12 (1)	1.63 (9)	1.22 (6)	2.81 (>20)	1.45 (8)
Area 1	90.09	2.71	58.75	48.66	91.49	48.71
Area 2	103.51	3.03	65.56	59.61	101.80	69.14
Area 3	90.29	2.61	52.55	54.41	80.21	51.27
Total distributed	285.64	8.47	178.49	163.90	276.31	170.57
Elapsed distributed	294	8	185	168	289	176

Table 8.12: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 118 substation test network split into 3 areas

The tie lines are indicated by the letter E in figure 8.2. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	416.08 (6)	32.63 (1)	161.70 (4)	124.83 (3)	190.64 (4)
Master	0.38 (2)	0.07 (1)	0.41 (3)	0.67 (5)	0.63 (5)
Area 1	57.78	4.36	35.88	48.71	47.87
Area 2	67.58	5.80	44.64	58.68	69.04
Area 3	99.77	6.44	58.39	71.08	72.22
Total distributed	225.51	16.67	139.32	179.15	189.76
Elapsed distributed	391	25	151	148	239

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	441.93 (6)	7.27 (1)	109.29 (3)	101.63 (3)	105.18 (3)	94.47 (3)
Master	1.23 (7)	0.11 (1)	1.29 (9)	0.92 (6)	1.08 (7)	2.54 (>20)
Area 1	58.47	2.27	48.93	43.53	39.38	61.32
Area 2	69.07	2.33	62.00	50.05	51.26	78.17
Area 3	109.99	3.10	61.24	48.30	62.41	95.83
Total distributed	238.76	7.81	173.46	142.80	154.13	237.86
Elapsed distributed	245	8	179	146	158	246

Table 8.13: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 118 substation test network split into 4 areas

The tie lines are indicated by the letter F in figure 8.2. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	416.08 (6)	32.63 (1)	161.70 (4)	124.83 (3)	190.64 (4)
Master	0.67 (3)	0.05 (1)	0.69 (3)	1.20 (6)	0.73 (4)
Area 1	64.38	5.14	35.30	64.15	54.56
Area 2	52.50	6.44	30.81	39.80	50.49
Area 3	65.94	4.63	42.66	52.29	48.01
Area 4	50.88	2.99	20.42	30.69	34.75
Total distributed	234.37	19.25	129.88	188.13	188.54
Elapsed distributed	237	19	131	193	192

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	441.93 (6)	7.27 (1)	109.29 (3)	101.63 (3)	105.18 (3)	94.47 (3)
Master	4.13 (>20)	2.90 (>20)	4.04 (>20)	2.34 (9)	3.72 (>20)	4.31 (>20)
Area 1	92.18	59.63	78.35	54.53	89.03	73.26
Area 2	71.59	50.23	61.26	46.91	64.27	64.76
Area 3	110.45	59.74	81.37	56.35	87.89	83.77
Area 4	77.06	38.01	43.35	32.97	55.25	46.50
Total distributed	355.41	210.51	268.37	193.10	300.16	272.60
Elapsed distributed	368	221	291	198	313	284

Table 8.14: Comparison between the solution times of the centralised and distributed versions of the 4 stage linear programming state estimation programs on the 118 substation test network split into 4 areas

The tie lines are indicated by the letter G in figure 8.2. The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

	Event number				
Program	1	2	3	4	5
Centralised	416.08 (6)	32.63 (1)	161.70 (4)	124.83 (3)	190.64 (4)
Master	0.69 (3)	0.11 (1)	0.96 (5)	1.19 (6)	1.12 (6)
Area 1	56.59	4.66	38.93	65.63	54.21
Area 2	60.05	6.98	55.72	53.16	65.27
Area 3	55.65	4.64	48.07	47.34	52.48
Area 4	67.96	4.64	33.00	34.96	46.87
Total distributed	240.94	21.03	176.68	202.28	219.95
Elapsed distributed	245	21	180	207	224

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

	Event number					
Program	1	2	3	4	5	6
Centralised	441.93 (6)	7.27 (1)	109.29 (3)	101.63 (3)	105.18 (3)	94.47 (3)
Master	2.38 (9)	0.16 (1)	2.07 (8)	2.25 (9)	2.60 (10)	4.63 (>20)
Area 1	61.74	2.17	41.07	50.52	56.16	67.11
Area 2	63.91	2.14	46.20	54.22	48.34	71.02
Area 3	69.09	2.46	41.15	42.66	55.79	71.91
Area 4	64.86	2.08	35.47	41.92	41.99	62.41
Total distributed	261.98	9.01	165.96	191.57	204.88	277.08
Elapsed distributed	268	9	171	196	211	287

substations than the other areas, then the area tasks for the smaller area tasks would idle while remaining area task completes its re-estimation based on the updated tie line estimates. The non-linear relationship between the problem size and the solution time means that the large area task would finish the re-estimation process disproportionately more slowly than the smaller area tasks. In a situation where the tasks are running on separate processors the effect on the elapsed time would be more noticeable but in the time sharing environment of a single processor the effect is small. The results of table 8.8 best illustrate the effect. The results presented in table 8.8 are for the 118 substation test network split into two areas of 70 and 48 substations respectively. The difference between the total CPU time and the elapsed time is considerably greater than in the results presented in tables 8.9 and 8.10 where the network has been divided into two areas of 57 and 61 substations respectively.

The timing results for the 118 substation test network with the measurements subject to no noise generally follow the trends outlined by the results from the 30 substation test network. A larger number of iterations (typically four to six) of the master task were often required. A more detailed investigation would be required to determine the reason for the increase. However the following points are likely to have a singificant influence: The measurement redundancy is lower on the 118 substation test network than on the 30 substation network. This may result in the tie line estimates calculated on the first pass being further away from the solution point, thus requiring additional iterations of the master task to reduce the discrepancies to acceptable limits: The electrical distances, that is the number of transmission lines, between a pair of termination nodes for the adjacent areas is likely to be greater in the 118 substation test network. Thus any small errors in the estimates arising from the transmission line difference equations may accumulate across all the transmission lines between the termination nodes resulting in the estimates for the termination nodes from the two areas differing by a large amount. This large difference will again require more iterations of the master task to reduce it to acceptable limits.

The timing results for the distributed method operating on both the 30 substation test network and the 118 substation test network using measurements subject to the addition of noise illustrates the unreliability of the method in its present form. The total CPU times for the distributed method were far higher than the corresponding times for the centralised method. The additional CPU requirements arise from the additional master task iterations

required to reduce the differences in the tie line estimates to acceptable limits. Indeed in several cases on both test networks the discrepancies were not resolved and after 20 iterations of the master task that state estimation run was abandoned. In the cases where convergence was not achieved, a stalemate situation has developed whereby one or both of the area tasks does not accept the updated tie line estimates from the master task and they repeatedly return to the master task the same value as before. The master task is receiving the same inputs as before and hence re-calculates the same updated tie line estimates as before.

The slow convergence experience in many cases often can be attributed to the least squares solution method utilised in the master task. This situation arises when one area task rejects the updated tie line estimates from the master task and repeatedly returns the same value. The other area however accepts the updated tie line estimates, adjusts all its local estimates accordingly and returns to the master task the same tie line estimate values as it received. The master task then processes the new values and returns a different value which is somewhere between the values supplied by both the areas. Since however one area repeatedly returns the same estimate values to the master task the updated tie line estimates gradually move towards those values over a number of master task iterations.

It was usually one or both of the voltage sub-estimation problems which had convergence problems and usually a problem existed at only one or two termination nodes. The difficulty of converging the voltage sub-estimation problems was aggravated in the situation where a termination node had two or more tie lines attached to it. (The other end of the tie lines terminate at substations which may or may not be in different areas, however this generally had no effect on the problem arising from this situation). In all cases the master task treats each tie line in isolation from the rest of the system, thus for the termination node with two or more tie lines attached to it, the master task would calculate two or more sets of updated tie line estimates from two or more different sets of data. As the updated voltage tie line estimates actually apply to the same point in the network, any difference in the updated voltage estimate values causes additional problems for the estimation process. The problem does not arise for the power flow sub-estimation problems because although the tie lines terminate at one point the updated power flow estimates are for specific tie lines and any errors are easily corrected by the power flow sum check equation at that point.

A considerable amount of time was spent trying to resolve the poor convergence problems in the presence of noise. The effort was directed

towards changing the way the weights on the updated tie line estimates in the area tasks were altered. This however made little effect on the reliability of the method and it is suggested that the following changes may lead to improved results: Additional intelligence could be built into either the area tasks or the master task to form one set of updated voltage tie line estimates for those termination nodes with two or more tie lines attached to them; The tolerance on the convergence of the tie line estimates may be reduceable without sacrificing the accuracy of the state estimates. The tolerance used was the same as that used in all previous cases, the state estimate values must not change by more than 0.0001 PU.. This is a fairly stringent tolerance in terms of a percentage accuracy; A more radical change would be to split the network at a substation instead of across a tie line. In this case each area would consider the substation to be part of its local network and would receive updated estimates for the bus voltages and the power flows of all the elements at the substation. This method of dividing the network would remove the voltage problems caused by two or more tie lines terminating at a single node and reduce the computational burden of the master task to that of merely averaging sets of values.

Should the method be made to converge in at most three master task iterations then, judging from the timing results for the results listed in tables 8.4 to 8.14 when no noise was applied to the measurements, it would appear that the elapsed time for a truly distributed implementation of the method might approach a time derived by dividing the equivalent time for the centralised method by the numbr of processors. It would however be unlikely to be less than this time as might be expected from a combination of a linear increase in CPU power and a non-linear decrease in solution time for the smaller problems. The distribution of the state estimation problem amoung several processors is gradually becoming an acceptable method of reducing the solution time of the state estimation problem. The method of distributing the 4 stage decomposed state estimation algorithm over two or more processors presented in this chapter is not generally reliable in its present form but it does indicate that there are advantages to be gained if the iterative process of transfering estimates between the area tasks and the master task can be kept to a minimum.

Chapter 9
Conclusions

The thesis has studied the subject of bad data detection and state estimation in electric power systems. The original algorithms for state estimation were developed in the early seventies and much work has been published on the topic. The trend in modern control centres is to analyse larger and larger networks with a greater accuracy. The reason for this is two fold, firstly the power system companies wish to model the network of neighbouring companies in order to be aware of any possible contingencies which may affect their network and to be able to monitor and control the trading of power between companies. The second reason for needing the ability to accurately model large networks is often a result of local statutory laws requiring the company to provide safer and more reliable supply. This requirement also necessitates the company being able to analyse the network of adjacent companies. The trend to analyse larger networks results in the need to develop more sophisticated algorithms with faster solution times which take advantage of modern computer architecture. The on-line implementation of the state estimation algorithm and its inherent need for fast solution times and accurate results provides the main driving force behind the research into new algorithms.

The thesis has detailed a novel state estimation algorithm which shows good performance in the presence of both gross measurement errors and incorrect switch status measurements. Many state estimation algorithms are capable of processing gross measurement errors with varying degrees of success. However the problem of incorrect switch status measurements has not previously been addressed by the state estimation algorithm and it is generally assumed that the state estimation program will be supplied with correct switch status measurements which may or may not have been validated by a separate program. The state estimation algorithm presented in the thesis is also capable of modelling individual busbars and bus-couplers within a substation, a feature which conventional state estimation algorithms are unable to do. The reason for this is that a bus-coupler has zero impedance which cannot be accommodated in the traditional mathematical model of the system. The ability to include the bus-couplers in the new model stems from the unique formulation of the state estimation problem.

The performance equations of an electric power system are non-linear and the solution of the state estimation problem formulated by combining the performance equations and the telemetered measurements thus requires the

solution of a set of non-linear equations. The conventional approach is to linearise the equations by taking first order partial differentials to form a set of linear equations (Newton Raphson method) which may then be solved by a conventional solution method such as a least squares method. In some cases the state estimation problem may be divided into two sets of linear equations. The linearisation of the non-linear equations is repeated in an iterative fashion using successive approximations to the solution point until convergence is achieved.

The state estimation algorithm presented in the thesis devides the problem into four sets of linear equations, the first to estimate the active power flows in the network, the second the reactive power flow estimates, the third the voltage magnitude estimates and the last the voltage phase estimates. The four sets of linear equations are solved in a cyclic iterative fashion. Each linear equation defines either an estimate, the difference between the estimates or the sum of two or more estimates in terms of either a telemetered measurement or a value calculated from the network performance equations. An error term is added to each equation to account for any measurement errors. The values calculated from the performance equations require the use of estimated values, hence the need for the iterative scheme. It should be noted that the performance equations are exact and no approximations have been made in the linearisation of themathematical model as in the case of conventional state estimation methods. Thus unlike some conventional state estimation algorithms which run into convergence problems if some of the transmission lines have a high ratio of reactance to resistance the proposed algorithm should not experience any convergence problems. Although this feature has not been explicitly verified in tests, the networks used to evaluate the performance of the algorithm have several lines with high ratios of reactance to resistance and no convergence problems were observed.

It is the unique formulation of the proposed state estimation algorithm which enables it to tolerate incorrect switch status measurements. In a conventional state estimation algorithm, if a transmission line is open circuit then the mathematical model of the network has to be re-formulated, which usually involves updating the bus admittance matrix. Should the transmission line actually be closed circuit then the model of the network is invalid and the state estimation algorithm is unable to correct the model. In the case of the proposed algorithm the transmission line would remain in the model and additional equations which define the power flow through the line to be zero would be appended to those already in existence. The weighting factors applied to the equations defining the difference in voltage levels

across the line would be decreased to reduce the influence of these equations on the final solution point. If however the transmission line is a closed circuit the algorithm is likely to be able to reject the invalid equations arising from the incorrect switch status measurement and estimate the correct voltage and power flow values from the other redundant equations. A possible enhancement to the ability of the algorithm to detect and correct switch status measurement errors would be to estimate the switch status measurements as the algorithm proceeds and to dynamically amend those equuations derived from the switch status measurements. Although this will increase the computational burden of the algorithm, it may not be as difficult to implement as first thought as the effect of an equation on the solution point can be reduced by decreasing its weighting factor.

The unique formulation of the state estimation problem also allows the busbars and bus-couplers of a substation to be incorporated into the model. The zero impedance bus-couplers are incorporated into the mathematical model of the system by adding equations to the four sub-estimation problems which transfer active and reactive power from one busbar to another with no drop in the voltage levels, thus for example a value of zero would be used in the equations for the difference between the voltage levels of two busbars connected together by a bus-coupler. Including the bus-couplers in the model of the system increases the size of the problem and thus increases the solution times of the algorithm. It does however provide a method state estimation which is totally independent of the validity of all switch status measurements and a decision would have to be made whether to trade faster solution times against less reliable results.

The bus-couplers do not usually have power flow measurements available and in the case where the number of bus-couplers is equal to or greater than the number of busbars then additional 'dummy' measurement equations defining the power flow through the bus-couplers are required. It is possible to append a least squares estimation algorithm to the main algorithms to estimate the power flows through the bus-couplers. The method of solving the under-determined set of equations was used in the simulator program which was also able to calculate the power flows through the bus-couplers. The estimates of the power flows in the bus-couplers could be of use to both a power system operator and a planning engineer. In an on-line environment the state estimator could supply the operator with information on the loadings of the bus-couplers. This information may be especially useful in emergency conditions when equipment maybe operating near to its limit. Alternatively the estimated power flows through the bus-couplers could be used as input to a

modified contingency analysis program which could advise on alternative substation configurations to minimise any overloads. A planning engineer could make use of an off-line version of the program to study the overall effects that a change in the design of a substation might make on the system.

The initial linear programming method adopted to solve the four sets of linear equations was based on the Revised Simplex method which minimises the sum of all the weighted error terms in any one set of linear equations. It is the use of linear programming which enables the algorithm to efficiently reject gross measurement errors. The linear programming algorithm arrives at a solution which has effectively selected the least noisy set of linearly independent equations required to define every variable (estimate) in the set of equations. Those equations which have not been selected do not influence the solution values at all, thus a measurement which is grossly in error is likely to be completely rejected. The results of the testing of the proposed algorithm illustrate its ability to detect gross measurement errors. Other authors have reported good results when using linear programming methods to solve the linear equations of the Newton Raphson method, however their algorithms still rely on an accurate topological model of the network and as a result cannot tolerate invalid switch status measurements.

The linear programming solution point is to be contrasted against the least squares solution point in which every equation influences the solution values, thus gross measurement errors tend to distort the estimates. This point is illustrated in chapter 6 when a least squares method was used to solve the linear equations formed in the proposed state estimation algorithm. Strictly speaking the linear programming method of solving a linear set of equations is not statistically optimal because the solution point does not reflect all of the valid measurements. However the author considers that the state estimates obtained from a set of measurements with a reasonable level of redundancy are sufficiently accurate for practical applications. A possible extension to the state estimation algorithm proposed in the thesis would be to delete all the linear equations which include measurement values the proposed method has identified as being grossly in error and process the remaining equations with a least squares method. The combination of the two solution methods would require some tuning to determine the optimum time to halt the linear programming method and commence the least squares method. The result should be an algorithm which features the advantages of both methods but has the slight disadvantage of a more complex and larger program.

During the development of the algorithm the on-line requirement for fast solution times was borne in mind. The original linear programming method used

to solve the four sets of equations resulted in solution times which were marginally better than those of a conventional full Newton Raphson based state estimator when operating on the larger test networks. It is doubtful whether the solution times would be as fast as those from a fast decoupled Newton Raphson state estimator if one had been available for comparison purposes. Hence considerable effort was made to improve the solution times and two alternative methods of formulating and solving the state estimation problem have been discussed. The first method adapted a network flow algorithm for solving the active and reactive sub-estimation problems while the second considered the distribution of the entire state estimation algorithm over a number of processors.

Although the network flow method showed promising results it is the author's opinion that none of the methods of solving the sets of linear equations investigated, namely the Revised Simplex linear programming method, the conjugate gradient least squares method and the network flow method are ideally suited to the solution of the four sets of linear equations required to solve the state estimation problem. The coefficient matrix of each of the four sets of linear equations has only plus one or minus one as the non zero elements and a linear programming method designed to exploit both this feature and the sparse nature of the matrix ought to return fast solution times. The implementation of the method could use integer variables and thus exploit the speed at which most computers are able to process integer arithmetic.

The subject of distributed state estimation has received much attention recently as a possible method of not only improving the solution times of state estimators but also other power system analysis programs. The cost of multi processor computers has to be weighed against the faster solution times, but if the relative cost of computer hardware continues to fall then it would seem to be a viable approach. Distribution of the state estimation program among multiple processors is a complex task and the simplistic approach adopted in the thesis was prone to convergence problems. It did however illustrate that considerable savings in solution time could be achieved and that even a distributed program running on a single processor could return faster solution times than the centralised method in ideal situations. The development of a distributed algorithm is a parallel development which could run alongside the development of the centralised algorithm. The savings in solution time indicate that provided the distributed algorithm described in the thesis can be made stable without further increasing the computational burden of the program then it is worth pursuing. Suggested approaches include reducing the convergence tolerance between the areas, provided this is not

significantly detrimental to the reliability of the estimates and partitioning the network at nodes as opposed to across transmission lines.

During the development of the state estimation algorithm proposed in the thesis it was recognised that the method of selecting the initial set of linearly independent equations to form the linear programming basis for each of the sub-estimation problems was detrimental to the solution times. The method of implementation required that every estimate had a measurement equation to define the initial value, for those estimates which did not then an artificial or dummy measurement equation was used. However once the program was running all these dummy equations were rejected (unless some were required to define power flows in the bus-couplers) and processing time was being wasted as the linear programming method repeatedly scanned all the equations, including the dummy measurement equations, to find equations eligible for a basis swap.

A brief investigation of the effect on the solution times of the removal of the dummy measurement equations was made by deleting all the unnecessary dummy measurement equations after the first iteration. If the results had been satisfactory a fast and efficient network observability algorithm would be used to form the initial linear programming basis. The effects of the solution times are listed in tabular form in appendix 6, these results are to be compared with those of tables 5.2 and 6.4. The results show reductions in the solution times of over 25% in some cases. However the algorithm was less able to reject gross voltage measurement errors. The cause of this was not fully investigated but it is likely to be as a result of the absence of the dummy measurement equations which initially equated the voltage magnitude estimates to 1.0 PU. In the absence of the dummy measurement equations the initial voltage magnitude estimates would be initialised to an arbitrary value calculated from a line difference equation. Thus this method of improving the solution times was abandoned in favour of finding a more efficient linear programming method.

With the ever expanding size and complexity of both electric power systems and their control centres the need for a state estimation algorithm which is tolerant of both gross measurement errors and incorrect switch status measurements will increase. The algorithm presented in the thesis is an alternative to the conventional methods of bad data detection and state estimation with the advantageous features not found in the conventional methods of being able to tolerate invalid switch status measurements and analyse information at the substation level.

Appendix One

Appendix one lists the network parameters not shown on the network diagrams in chapter 3. Namely the generator and line parameters for the 5,the 30, the 57 and the 118 substation test networks together with the measurement points for the 57 and the 118 substation test networks.

The following abbreviations are used for the column headers. Note all the parameters (unless otherwise specified) are in per unit (P.U.) on a 100 MW. base.

GBUS	Generator bus-section number (no unit).
GPLLMT	Lower limit on generator active power output.
GPHLMT	Upper limit on generator active power output.
GRLI	Generator rate of change limit for increasing active power output (mega watt/second).
GRLD	Generator rate of change limit for decreasing active power output (mega watt/second).
GQLLMT	Lower limit on generator reactive power output.
GQHLMT	Upper limit on generator reactive power output.
RA	Generator armature resistance.
XA	Generator transient reactance.
H	Generator rotational inertia constant.
TC	Generator steam time constant (seconds)
GAIN	Voltage governor gain constant of generator.
GENP	Constant production cost term of generator (unit price/mega watt hour).
GENPC	Linear production cost term of generator (unit price/mega watt hour).
GENPQ	Quadratic production cost term of generator (unit price/mega watt hour).
GENSC	Cold start cost term of generator (unit price).
LNSBUS	Sending bus-section number of line (no unit).

LNRBUS	Receiving bus-section number of line (no unit).
LNRES	Series resistance of line.
LNREAC	Series reactance of line.
LNCRG	Line charging susceptace of line.
LNLMT	Active power flow limit of line.
LDBUS	Load bus-section number (no unit).

Generator parameters for the 5 substation test network.

No.	GBUS	GPLLMT	GPHLMT	GRLI	GRLD	GQLLMT	GQHLMT
1	1	0.0	1.5	0.0016	0.0016	0.8	-0.8
2	8	0.0	1.0	0.0016	0.0016	0.5	-0.5

No.	RA	XA	H	TC	GAIN	GENP	GENPC	GENPQ	GENSC
1	0.0	0.25	50.0	0.3	1.0	29.0	150.0	100.0	113.0
2	0.0	1.50	1.00	0.3	1.0	25.0	200.0	150.0	105.0

Line parameters for the 5 substation test network.

No.	LNSBUS	LNRBUS	LNRES	LNREAC	LNCRG	LNLMT
1	1	5	0.02	0.06	0.030	2.0
2	1	2	0.08	0.24	0.025	2.0
3	2	5	0.06	0.18	0.020	2.0
4	4	6	0.06	0.18	0.020	2.0
5	8	9	0.04	0.12	0.015	2.0
6	3	4	0.01	0.03	0.010	2.0
7	4	9	0.08	0.24	0.025	2.0

Generator parameters for the 30 substation test network.

No.	GBUS	GPLLMT	GPHLMT	GRLI	GRLD	GQLLMT	GQHLMT
1	1	0.50	2.00	0.0016	0.0016	1.00	-1.00
2	3	0.50	1.50	0.0016	0.0016	1.00	-1.00
3	8	0.20	0.70	0.0020	0.0020	0.50	-0.50
4	22	0.10	0.50	0.0020	0.0020	0.25	-0.25
5	36	0.10	0.50	0.0020	0.0020	0.25	-0.25
6	39	0.10	0.50	0.0020	0.0020	0.25	-0.25

No.	RA	XA	H	TC	GAIN	GENP	GENPC	GENPQ	GENSC
1	0.0	0.25	50.0	0.3	1.0	29.0	190.0	100.0	113.0
2	0.0	0.25	50.0	0.3	1.0	29.0	200.0	150.0	113.0
3	0.0	0.50	20.0	0.3	1.0	25.0	210.0	170.0	101.0
4	0.0	0.50	20.0	0.3	1.0	15.0	210.0	170.0	85.00
5	0.0	0.50	20.0	0.3	1.0	15.0	210.0	170.0	85.00
6	0.0	0.50	20.0	0.3	1.0	15.0	210.0	170.0	85.00

Line parameters for the 30 substation test network.

No.	LNSBUS	LNRBUS	LNRES	LNREAC	LNCRG	LNLMT
1	1	3	0.0192	0.0575	0.0264	1.30
2	1	4	0.0452	0.1852	0.0204	1.30
3	2	6	0.0570	0.1737	0.0184	0.65
4	4	5	0.0132	0.0379	0.0042	1.30
5	3	9	0.0472	0.1983	0.0209	1.30
6	2	12	0.0581	0.1763	0.0187	0.65
7	6	14	0.0119	0.0414	0.0045	0.90
8	7	16	0.0460	0.1160	0.0102	0.70
9	12	16	0.0267	0.0820	0.0085	1.30
10	13	18	0.0120	0.0420	0.0045	0.32
11	11	25	0.0000	0.2080	0.0000	0.65
12	10	34	0.0000	0.5560	0.0000	0.32
13	25	36	0.0000	0.2080	0.0000	0.65
14	25	35	0.0000	0.1100	0.0000	0.65
15	5	38	0.0000	0.2560	0.0000	0.65
16	38	39	0.0000	0.1400	0.0000	0.65
17	37	40	0.1231	0.2559	0.0000	0.32
18	37	44	0.0662	0.1304	0.0000	0.32
19	37	45	0.0945	0.1987	0.0000	0.32
20	40	42	0.2210	0.1997	0.0000	0.16
21	45	46	0.0824	0.1932	0.0000	0.16
22	43	47	0.1070	0.2185	0.0000	0.16
23	47	48	0.0639	0.1292	0.0000	0.16
24	48	49	0.0340	0.0680	0.0000	0.32
25	33	49	0.0936	0.2090	0.0000	0.32
26	26	46	0.0324	0.0845	0.0000	0.32
27	29	50	0.0348	0.0749	0.0000	0.32
28	30	51	0.0727	0.1499	0.0000	0.32
29	50	51	0.0116	0.0236	0.0000	0.32
30	41	52	0.1000	0.2020	0.0000	0.16
31	51	54	0.1150	0.1790	0.0000	0.16
32	52	54	0.1320	0.2700	0.0000	0.16
33	53	55	0.1885	0.3292	0.0000	0.16
34	55	56	0.2544	0.3800	0.0000	0.16
35	55	62	0.1093	0.2087	0.0000	0.16
36	63	61	0.0000	0.3960	0.0000	0.65
37	59	66	0.2198	0.4153	0.0000	0.16
38	60	71	0.3202	0.6027	0.0000	0.16
39	65	73	0.2399	0.4533	0.0000	0.16
40	24	63	0.0636	0.2000	0.0214	0.32
41	15	63	0.0169	0.0599	0.0065	0.32

Generator parameters for the 57 substation test network.

No.	GBUS	GPLLMT	GPHLMT	GRLI	GRLD	GQLLMT	GQHLMT
1	1	1.00	5.00	0.0015	0.0015	2.00	-2.00
2	2	0.50	1.50	0.0020	0.0020	1.00	-1.00
3	3	0.50	3.00	0.0016	0.0016	1.50	-1.50
4	6	0.20	1.00	0.0025	0.0025	0.75	-0.75
5	8	0.50	4.00	0.0015	0.0015	1.00	-1.00
6	9	0.15	2.00	0.0020	0.0020	1.00	-1.00
7	12	1.00	6.00	0.0014	0.0014	2.50	-2.50

No.	RA	XA	H	TC	GAIN	GENP	GENPC	GENPQ	GENSC
1	0.0	0.20	75.0	0.3	1.0	44.0	175.0	95.00	120.0
2	0.0	0.25	50.0	0.3	1.0	29.0	200.0	150.0	113.0
3	0.0	0.20	60.0	0.3	1.0	42.0	180.0	100.0	115.0
4	0.0	0.50	20.0	0.3	1.0	27.0	210.0	170.0	101.0
5	0.0	0.30	65.0	0.3	1.0	43.0	175.0	95.00	118.0
6	0.0	0.25	50.0	0.3	1.0	29.0	190.0	100.0	113.0
7	0.0	0.20	80.0	0.3	1.0	45.0	170.0	90.00	125.0

Load points for the 57 substation test network.

Load No.	LDBUS	Load No.	LDBUS	Load No.	LDBUS
1	1	2	2	3	3
4	5	5	6	6	8
7	9	8	10	9	12
10	13	11	14	12	15
13	16	14	17	15	18
16	19	17	20	18	23
19	25	20	27	21	28
22	29	23	30	24	31
25	32	26	33	27	35
28	38	29	41	30	42
31	43	32	44	33	47
34	49	35	50	36	51
37	52	38	53	39	54
40	55	41	56	42	57
43	48	44	18	45	25
46	53				

Line parameters for the 57 substation test network.

No.	LNSBUS	LNRBUS	LNRES	LNREAC	LNCRG	LNLMT
1	1	2	0.0083	0.0280	0.0645	2.00
2	2	3	0.0298	0.0850	0.0409	2.00
3	3	4	0.0112	0.0366	0.0190	1.00
4	4	5	0.0625	0.1320	0.0129	0.32
5	4	6	0.0430	0.1480	0.0174	0.32
6	6	7	0.0200	0.1020	0.0138	0.40
7	6	8	0.0339	0.1730	0.0235	0.90
8	8	9	0.0099	0.0505	0.0274	4.00
9	9	10	0.0369	0.1679	0.0220	0.40
10	9	11	0.0258	0.0848	0.0109	0.32
11	9	12	0.0648	0.2950	0.0386	0.16

12	9	13	0.0481	0.1580	0.0203	0.16
13	13	14	0.0132	0.0434	0.0055	0.20
14	13	15	0.2690	0.0869	0.0115	1.00
15	1	15	0.0178	0.0910	0.0494	3.00
16	1	16	0.0454	0.2060	0.0273	1.50
17	1	17	0.0238	0.1080	0.0143	2.00
18	3	15	0.0162	0.0530	0.0272	0.70
19	4	18	0.0000	0.2423	0.0000	0.65
20	5	6	0.0302	0.0641	0.0062	0.16
21	7	8	0.0139	0.0712	0.0097	1.50
22	10	12	0.0277	0.1262	0.0164	0.40
23	11	13	0.0223	0.0732	0.0094	0.20
24	12	13	0.0178	0.0580	0.0302	0.16
25	12	16	0.0180	0.0813	0.0108	0.55
26	12	17	0.0397	0.1790	0.0238	1.00
27	14	15	0.0171	0.0547	0.0074	1.30
28	18	19	0.4610	0.6850	0.0000	0.16
29	19	20	0.2830	0.4340	0.0000	0.16
30	20	21	0.0000	0.7767	0.0000	0.16
31	21	22	0.0736	0.1170	0.0000	0.16
32	22	23	0.0099	0.0152	0.0000	0.32
33	23	24	0.1660	0.2560	0.0042	0.16
34	24	25	0.0000	0.6028	0.0000	0.32
35	24	26	0.0000	0.0473	0.0000	0.16
36	26	27	0.1650	0.2540	0.0000	0.16
37	27	28	0.0618	0.0954	0.0000	0.32
38	28	29	0.0418	0.0587	0.0000	0.32
39	7	29	0.0000	0.0648	0.0000	1.00
40	25	30	0.1350	0.2020	0.0000	0.16
41	30	31	0.3260	0.4970	0.0000	0.16
42	31	32	0.5070	0.7550	0.0000	0.16
43	32	33	0.0392	0.0360	0.0000	0.16
44	32	34	0.0000	0.9530	0.0000	0.16
45	34	35	0.0520	0.0780	0.0016	0.16
46	35	36	0.0430	0.0537	0.0008	0.32
47	36	37	0.0290	0.0366	0.0000	0.32
48	37	38	0.0651	0.1009	0.0010	0.40
49	37	39	0.0239	0.0379	0.0000	0.16
50	36	40	0.0300	0.0466	0.0000	0.16
51	22	38	0.0192	0.0295	0.0000	0.32
52	11	41	0.0000	0.7490	0.0000	0.16
53	41	42	0.2070	0.3520	0.0000	0.16
54	41	43	0.0000	0.4120	0.0000	0.32
55	38	44	0.0289	0.0585	0.0010	0.60
56	15	45	0.0000	0.1042	0.0000	0.90
57	14	46	0.0000	0.0735	0.0000	0.90
58	46	47	0.0230	0.0680	0.0016	0.90
59	47	48	0.0182	0.0233	0.0000	0.32
60	48	49	0.0834	0.1290	0.0024	0.16
61	49	50	0.0801	0.1280	0.0000	0.16
62	50	51	0.1386	0.2200	0.0000	0.32
63	10	51	0.0000	0.0712	0.0000	0.65
64	13	49	0.0000	0.1910	0.0000	0.65
65	29	52	0.1442	0.1870	0.0000	0.40
66	52	53	0.0762	0.0984	0.0000	0.32
67	53	54	0.1878	0.2320	0.0000	0.16
68	54	55	0.1732	0.2265	0.0000	0.16
69	11	43	0.0000	0.1530	0.0000	0.32

70	44	45	0.0624	0.1242	0.0020	0.90
71	40	56	0.0000	1.1950	0.0000	0.16
72	41	56	0.5530	0.5490	0.0000	0.16
73	42	56	0.2125	0.3540	0.0000	0.16
74	39	57	0.0000	1.3550	0.0000	0.16
75	56	57	0.1740	0.2600	0.0000	0.16
76	38	49	0.1150	0.1770	0.0030	0.16
77	38	48	0.0312	0.0482	0.0000	0.32
78	9	55	0.0000	0.1205	0.0000	0.32

Measurement points for the 57 substation test network.

Frequency measurements.

Meas No.	Bus No.	Meas No.	Bus No.	Meas No.	Bus No.
1	1	2	2	3	3
4	6	5	8	6	9
7	12	8	18	9	25
10	53				

Voltage measurements.

Meas No.	Bus No.	Meas No.	Bus No.	Meas No.	Bus No.
1	1	2	2	3	3
4	4	5	6	6	7
7	8	8	9	9	10
10	11	11	12	12	13
13	14	14	15	15	18
16	20	17	21	18	24
19	25	20	26	21	29
22	32	23	39	24	40
25	41	26	43	27	44
28	45	29	46	30	49
31	51	32	53	33	55
34	56	35	57		

Generator power flow measurements.

Meas No.	Gen No.	Meas No.	Gen No.	Meas No.	Gen No.
1	1	2	2	3	3
4	4	5	5	6	6
7	7				

Load power flow measurements.

Meas No.	Load No.	Meas No.	Load No.	Meas No.	Load No.
1	1	2	2	3	3
4	5	5	6	6	7
7	8	8	9	9	10
10	11	11	12	12	15
13	17	14	18	15	19
16	20	17	21	18	22
19	23	20	24	21	25
22	26	23	27	24	28
25	29	26	30	27	31
28	32	29	33	30	34
31	35	32	36	33	37
34	38	35	39	36	40
37	41	38	42	39	43

Line power flow measurements.
(S) => measurement at the sending end of the line
(R) => measurement at the receiving end of the line

Meas No.	Line No.	Meas No.	Line No.	Meas No.	Line No.
1	1 (S)	2	1 (R)	3	2 (S)
4	2 (R)	5	3 (S)	6	3 (R)
7	4 (S)	8	5 (S)	9	5 (R)
10	6 (S)	11	7 (S)	12	7 (R)
13	8 (S)	14	8 (R)	15	9 (S)
16	10 (S)	17	11 (S)	18	11 (R)
19	12 (S)	20	12 (R)	21	13 (S)
22	14 (S)	23	15 (S)	24	15 (R)
25	16 (S)	26	17 (S)	27	18 (S)
28	18 (R)	29	19 (R)	30	20 (R)
31	22 (R)	32	23 (S)	33	24 (S)
34	24 (R)	35	25 (S)	36	26 (S)
37	27 (S)	38	28 (S)	39	29 (R)
40	33 (R)	41	34 (R)	42	36 (S)
43	38 (R)	44	40 (S)	45	42 (R)
46	43 (S)	47	45 (S)	48	48 (R)
49	49 (R)	50	50 (R)	51	51 (R)
52	52 (R)	53	53 (R)	54	54 (R)
55	57 (R)	56	58 (R)	57	59 (S)
58	60 (R)	59	61 (S)	60	62 (R)
61	64 (R)	62	65 (S)	63	66 (R)
64	67 (S)	65	68 (R)	66	69 (S)
67	70 (R)	68	72 (R)	69	76 (S)
70	76 (R)	71	77 (R)	72	78 (S)

Generator parameters for the 118 substation test network.

No.	GBUS	GPLLMT	GPHLMT	GRLI	GRLD	GQLLMT	GQHLMT
1	1	0.20	0.50	0.0020	0.0020	0.30	-0.30
2	4	0.10	0.40	0.0025	0.0025	0.90	-0.90
3	6	0.20	0.50	0.0015	0.0015	0.60	-0.60
4	8	0.20	0.50	0.0020	0.0020	1.50	-1.50
5	10	1.00	6.60	0.0010	0.0010	3.00	-3.00
6	12	0.20	1.00	0.0020	0.0020	0.50	-0.50
7	15	0.10	0.30	0.0025	0.0025	0.50	-0.50
8	18	0.10	0.30	0.0025	0.0025	0.50	-0.50
9	19	0.10	0.30	0.0025	0.0025	0.50	-0.50
10	24	0.20	0.50	0.0020	0.0020	0.90	-0.90
11	25	0.75	4.00	0.0014	0.0014	2.00	-2.00
12	26	1.00	5.00	0.0014	0.0014	3.00	-3.00
13	27	0.10	0.30	0.0025	0.0025	0.50	-0.50
14	31	0.20	0.40	0.0020	0.0020	0.50	-0.50
15	32	0.10	0.30	0.0025	0.0025	0.50	-0.50
16	34	0.10	0.40	0.0025	0.0025	0.90	-0.90
17	36	0.10	0.30	0.0025	0.0025	0.90	-0.90
18	40	0.20	0.50	0.0015	0.0015	0.90	-0.90
19	42	0.20	0.50	0.0015	0.0015	0.50	-0.50
20	46	0.10	0.40	0.0025	0.0025	0.50	-0.50
21	49	0.50	3.50	0.0015	0.0015	2.00	-2.00
22	54	0.20	0.70	0.0020	0.0020	0.35	-0.35
23	55	0.10	0.40	0.0025	0.0025	0.50	-0.50
24	56	0.10	0.40	0.0025	0.0025	0.50	-0.50
25	59	0.50	2.00	0.0016	0.0016	1.00	-1.00
26	61	0.50	2.00	0.0016	0.0016	1.00	-1.00
27	62	0.20	0.50	0.0018	0.0018	0.90	-0.90
28	65	1.50	5.00	0.0012	0.0012	3.00	-3.00
29	66	1.50	5.00	0.0012	0.0012	3.00	-3.00
30	69	1.50	6.60	0.0012	0.0012	3.00	-3.00
31	70	0.20	0.50	0.0020	0.0020	0.90	-0.90
32	72	0.10	0.30	0.0025	0.0025	0.50	-0.50
33	73	0.10	0.30	0.0025	0.0025	0.50	-0.50
34	74	0.20	0.50	0.0018	0.0018	0.50	-0.50
35	76	0.20	0.50	0.0020	0.0020	0.90	-0.90
36	77	0.10	0.30	0.0025	0.0025	0.90	-0.90
37	80	1.00	6.00	0.0015	0.0015	3.00	-3.00
38	85	0.10	0.30	0.0025	0.0025	0.50	-0.50
39	87	0.10	0.50	0.0018	0.0018	0.90	-0.90
40	89	1.50	6.60	0.0012	0.0012	3.30	-3.30
41	90	0.20	1.00	0.0020	0.0020	0.50	-0.50
42	91	0.10	0.50	0.0025	0.0025	0.35	-0.35
43	92	0.10	0.30	0.0025	0.0025	0.50	-0.50
44	99	0.10	0.30	0.0025	0.0025	0.50	-0.50
45	100	0.50	3.00	0.0015	0.0015	2.00	-2.00
46	103	0.20	0.70	0.0020	0.0020	0.35	-0.35
47	104	0.10	0.30	0.0025	0.0025	0.50	-0.50
48	105	0.10	0.30	0.0025	0.0025	0.50	-0.50
49	107	0.10	0.30	0.0025	0.0025	0.50	-0.50
50	110	0.10	0.30	0.0025	0.0025	0.50	-0.50
51	111	0.10	0.50	0.0025	0.0025	0.90	-0.90
52	112	0.10	0.50	0.0025	0.0025	0.90	-0.90
53	113	0.10	0.30	0.0025	0.0025	0.50	-0.50
54	116	0.50	2.50	0.0016	0.0016	1.70	-1.70

No.	RA	XA	H	TC	GAIN	GENP	GENPC	GENPQ	GENSC
1	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
2	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
3	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
4	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
5	0.0	0.20	80.0	0.3	0.2	45.0	170.0	90.00	125.0
6	0.0	0.50	20.0	0.3	0.2	27.0	210.0	170.0	101.0
7	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
8	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
9	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
10	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
11	0.0	0.30	65.0	0.3	0.2	43.0	175.0	95.00	118.0
12	0.0	0.30	65.0	0.3	0.2	44.0	170.0	95.00	118.0
13	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
14	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
15	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
16	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
17	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
18	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
19	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
20	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
21	0.0	0.20	60.0	0.3	0.2	42.0	180.0	100.0	115.0
22	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
23	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
24	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
25	0.0	0.30	45.0	0.3	0.2	40.0	190.0	130.0	110.0
26	0.0	0.30	45.0	0.3	0.2	40.0	190.0	130.0	110.0
27	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
28	0.0	0.20	80.0	0.3	0.2	45.0	170.0	90.00	125.0
29	0.0	0.20	80.0	0.3	0.2	45.0	170.0	90.00	125.0
30	0.0	0.20	80.0	0.3	0.2	45.0	170.0	90.00	125.0
31	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
32	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
33	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
34	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
35	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
36	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
37	0.0	0.20	80.0	0.3	0.2	45.0	170.0	90.00	125.0
38	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
39	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
40	0.0	0.20	80.0	0.3	0.2	45.0	170.0	90.00	125.0
41	0.0	0.45	25.0	0.3	0.2	27.0	205.0	165.0	100.0
42	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
43	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
44	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
45	0.0	0.20	60.0	0.3	0.2	42.0	180.0	100.0	115.0
46	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
47	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
48	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
49	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
50	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
51	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
52	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
53	0.0	0.50	20.0	0.3	0.2	15.0	210.0	170.0	85.00
54	0.0	0.25	55.0	0.3	0.2	40.0	180.0	105.0	110.0

Load points for the 118 substation test network

Load No.	LDBUS	Load No.	LDBUS	Load No.	LDBUS
1	1	2	2	3	3
4	4	5	6	6	7
7	11	8	12	9	13
10	14	11	15	12	16
13	17	14	18	15	19
16	20	17	21	18	22
19	23	20	27	21	28
22	29	23	31	24	32
25	33	26	34	27	35
28	36	29	39	30	40
31	41	32	42	33	43
34	44	35	45	36	46
37	47	38	48	39	49
40	50	41	51	42	52
43	53	44	54	45	55
46	56	47	57	48	58
49	59	50	60	51	62
52	66	53	67	54	70
55	74	56	75	57	76
58	77	59	78	60	79
61	80	62	82	63	83
64	84	65	85	66	86
67	88	68	90	69	92
70	93	71	94	72	95
73	96	74	97	75	98
76	100	77	101	78	102
79	103	80	104	81	105
82	106	83	107	84	108
85	109	86	110	87	112
88	114	89	115	90	117
91	118	92	5	93	34
94	37	95	44	96	45
97	46	98	48	99	74
100	79	101	82	102	83
103	105	104	107	105	110

Line parameters for the 118 substation test network.

No.	LNSBUS	LNRBUS	LNRES	LNREAC	LNCRG	LNLMT
1	1	2	0.0303	0.0999	0.0254	0.32
2	1	3	0.0129	0.0424	0.0108	0.90
3	4	5	0.0018	0.0080	0.0021	2.00
4	3	5	0.0241	0.1080	0.0284	1.50
5	5	6	0.0119	0.0540	0.0143	2.00
6	6	7	0.0046	0.0208	0.0055	0.65
7	8	9	0.0024	0.0305	1.1620	7.00
8	5	8	0.0000	0.0267	0.0000	7.00
9	9	10	0.0026	0.0322	1.2300	7.00
10	4	11	0.0209	0.0688	0.0175	1.30
11	5	11	0.0203	0.0682	0.0174	1.30
12	11	12	0.0060	0.0196	0.0050	0.65
13	2	12	0.0187	0.0616	0.0157	0.65
14	3	12	0.0484	0.1600	0.0406	0.32
15	7	12	0.0086	0.0340	0.0087	0.32
16	11	13	0.0223	0.0731	0.0188	0.65
17	12	14	0.0215	0.0707	0.0182	0.32
18	13	15	0.0744	0.2444	0.0627	0.16
19	14	15	0.0595	0.1950	0.0502	0.16
20	12	16	0.0212	0.0834	0.0214	0.16
21	15	17	0.0132	0.0437	0.0444	2.00
22	16	17	0.0454	0.1801	0.0466	0.32
23	17	18	0.0123	0.0505	0.0130	1.30
24	18	19	0.0112	0.0493	0.0114	0.32
25	19	20	0.0252	0.1170	0.0298	0.32
26	15	19	0.0120	0.0394	0.0101	0.32
27	20	21	0.0183	0.0849	0.0216	0.50
28	21	22	0.0209	0.0970	0.0246	0.65
29	22	23	0.0342	0.1590	0.0404	0.90
30	23	24	0.0135	0.0492	0.0498	0.32
31	23	25	0.0156	0.0800	0.0864	2.00
32	25	26	0.0000	0.0382	0.0000	1.50
33	25	27	0.0318	0.1630	0.1764	2.00
34	27	28	0.0191	0.0855	0.0216	0.65
35	28	29	0.0237	0.0943	0.0238	0.32
36	17	30	0.0000	0.0388	0.0000	3.50
37	26	30	0.0080	0.0860	0.9180	3.50
38	8	30	0.0043	0.0504	0.5140	1.50
39	17	31	0.0474	0.1563	0.0399	0.32
40	29	31	0.0108	0.0331	0.0083	0.16
41	23	32	0.0317	0.1153	0.1173	2.00
42	31	32	0.0298	0.0985	0.0251	0.65
43	27	32	0.0229	0.0755	0.0193	0.32
44	15	33	0.0380	0.1244	0.0319	0.16
45	19	34	0.0752	0.2470	0.0632	0.16
46	35	36	0.0022	0.0102	0.0027	0.16
47	35	37	0.0110	0.0497	0.0132	0.65
48	33	37	0.0415	0.1420	0.0366	0.32
49	34	36	0.0087	0.0268	0.0057	0.65
50	34	37	0.0026	0.0094	0.0099	2.00
51	37	38	0.0000	0.0375	0.0000	4.00
52	37	39	0.0321	0.1060	0.0270	1.00
53	37	40	0.0593	0.1680	0.0420	0.90
54	30	38	0.0046	0.0540	0.4220	1.30
55	39	40	0.0184	0.0605	0.0155	0.65

56	40	41	0.0145	0.0487	0.0122	0.32
57	40	42	0.0555	0.1830	0.0466	0.32
58	41	42	0.0410	0.1350	0.0344	0.32
59	43	44	0.0608	0.2454	0.0607	0.32
60	34	43	0.0413	0.1681	0.0423	0.32
61	44	45	0.0224	0.0901	0.0224	0.65
62	45	46	0.0400	0.1356	0.0332	0.65
63	46	47	0.0380	0.1270	0.0316	0.65
64	46	48	0.0601	0.1890	0.0472	0.32
65	47	49	0.0191	0.0625	0.0160	0.16
66	42	49	0.0636	0.0287	0.1720	2.00
67	45	49	0.0684	0.1860	0.0444	1.00
68	48	49	0.0179	0.0505	0.0126	0.65
69	49	50	0.0267	0.0752	0.0187	1.00
70	49	51	0.0486	0.1370	0.0342	1.30
71	51	52	0.0203	0.0588	0.0140	0.65
72	52	53	0.0405	0.1635	0.0406	0.16
73	53	54	0.0263	0.1220	0.0311	0.16
74	49	54	0.0670	0.2606	0.1469	1.30
75	54	55	0.0169	0.0707	0.0202	0.16
76	54	56	0.0028	0.0096	0.0073	0.32
77	55	56	0.0049	0.0151	0.0037	0.32
78	56	57	0.0343	0.0966	0.0242	0.32
79	50	57	0.0474	0.1340	0.0332	0.65
80	56	58	0.0343	0.0966	0.0242	0.16
81	51	58	0.0255	0.0719	0.0179	0.32
82	54	59	0.0503	0.2293	0.0598	0.65
83	56	59	0.0407	0.1224	0.1105	1.30
84	55	59	0.0474	0.2158	0.0565	0.65
85	59	60	0.0317	0.1450	0.0376	0.90
86	59	61	0.0328	0.1500	0.0388	0.90
87	60	61	0.0026	0.0135	0.0146	2.00
88	60	62	0.0123	0.0561	0.0147	0.32
89	61	62	0.0082	0.0376	0.0098	0.32
90	59	63	0.0000	0.0386	0.0000	2.00
91	63	64	0.0017	0.0200	0.2160	2.00
92	61	64	0.0000	0.0268	0.0000	2.00
93	38	65	0.0090	0.0986	1.1460	3.00
94	64	65	0.0027	0.0302	0.3800	3.00
95	49	66	0.0090	0.0460	0.0496	4.00
96	62	66	0.0482	0.2180	0.0578	0.65
97	62	67	0.0258	0.1170	0.0310	0.65
98	65	66	0.0000	0.0370	0.0000	0.90
99	66	67	0.0224	0.1015	0.0268	0.90
100	65	68	0.0014	0.0160	0.6380	1.50
101	47	69	0.0844	0.2778	0.0709	1.00
102	49	69	0.0985	0.3240	0.0828	1.00
103	68	69	0.0000	0.0370	0.0000	2.00
104	69	70	0.0300	0.1270	0.1220	2.00
105	24	70	0.1022	0.4115	0.1020	0.16
106	70	71	0.0088	0.0355	0.0088	0.65
107	24	72	0.0488	0.1960	0.0488	0.10
108	71	72	0.0446	0.1800	0.0444	0.32
109	71	73	0.0087	0.0454	0.0118	0.32
110	70	74	0.0401	0.1323	0.0337	0.32
111	70	75	0.0428	0.1410	0.0360	0.32
112	69	75	0.0405	0.1220	0.1240	2.00
113	74	75	0.0123	0.0406	0.0103	1.00

114	76	77	0.0444	0.1480	0.0368	1.00
115	69	77	0.0309	0.1010	0.1038	1.00
116	75	77	0.0601	0.1999	0.0498	0.65
117	77	78	0.0038	0.0124	0.0126	0.90
118	78	79	0.0055	0.0244	0.0067	0.65
119	77	80	0.0109	0.0332	0.0700	3.00
120	79	80	0.0156	0.0704	0.0187	1.30
121	68	81	0.0018	0.0202	0.8080	0.90
122	80	81	0.0000	0.0370	0.0000	0.90
123	77	82	0.0298	0.0853	0.0817	0.32
124	82	83	0.0112	0.0367	0.0380	0.90
125	83	84	0.0625	0.1320	0.0258	0.32
126	83	85	0.0430	0.1480	0.0348	0.90
127	84	85	0.0302	0.0641	0.0123	0.65
128	85	86	0.0350	0.1230	0.0276	0.32
129	86	87	0.0282	0.2074	0.0445	0.32
130	85	88	0.0200	0.1020	0.0276	0.90
131	85	89	0.0239	0.1730	0.0470	1.30
132	88	89	0.0139	0.0712	0.0193	1.50
133	89	90	0.0164	0.0652	0.1588	3.00
134	90	91	0.0254	0.0836	0.0214	0.32
135	89	92	0.0080	0.0383	0.0962	4.00
136	91	92	0.0387	0.1272	0.0327	0.32
137	92	93	0.0258	0.0848	0.0218	0.90
138	92	94	0.0481	0.1580	0.0406	0.90
139	93	94	0.0223	0.0732	0.0188	0.90
140	94	95	0.0132	0.0434	0.0111	0.90
141	80	96	0.0356	0.1820	0.0494	0.32
142	82	96	0.0162	0.0530	0.0544	0.16
143	94	96	0.0269	0.0869	0.0230	0.32
144	80	97	0.0183	0.0934	0.0254	0.65
145	80	98	0.0238	0.1080	0.0286	0.65
146	80	99	0.0454	0.2060	0.0546	0.32
147	92	100	0.0648	0.2950	0.0772	0.65
148	94	100	0.0178	0.0580	0.0604	0.16
149	95	96	0.0171	0.0547	0.0147	0.32
150	96	97	0.0173	0.0885	0.0240	0.32
151	98	100	0.0397	0.1790	0.0476	0.10
152	99	100	0.0180	0.0813	0.0216	0.32
153	100	101	0.0277	0.1262	0.0328	0.32
154	92	102	0.0123	0.0559	0.0146	0.90
155	101	102	0.0246	0.1120	0.0284	0.65
156	100	103	0.0160	0.0525	0.0536	2.00
157	100	104	0.0451	0.2040	0.0541	0.90
158	103	104	0.0466	0.1584	0.0407	0.65
159	103	105	0.0535	0.1625	0.0408	0.65
160	100	106	0.0605	0.2290	0.0620	0.90
161	104	105	0.0099	0.0378	0.0099	0.90
162	105	106	0.0140	0.0547	0.0143	0.32
163	105	107	0.0530	0.1830	0.0472	0.32
164	105	108	0.0261	0.0703	0.0184	0.32
165	106	107	0.0530	0.1830	0.0472	0.32
166	108	109	0.0105	0.0288	0.0076	0.32
167	103	110	0.0391	0.1813	0.0461	0.90
168	109	110	0.0278	0.0762	0.0202	0.32
169	110	111	0.0220	0.0755	0.0200	0.65
170	110	112	0.0247	0.0640	0.0620	0.90
171	17	113	0.0091	0.0301	0.0077	0.65

172	32	113	0.0615	0.2030	0.0518	0.10
173	32	114	0.0135	0.0612	0.0163	0.16
174	27	115	0.0164	0.0741	0.0197	0.32
175	114	115	0.0023	0.0104	0.0028	0.10
176	68	116	0.0003	0.0041	0.1640	2.60
177	12	117	0.0329	0.1400	0.0358	0.32
178	75	118	0.0145	0.0481	0.0120	0.65
179	76	118	0.0164	0.0544	0.0136	0.16

Measurement points for the 118 substation test network.

Frequency measurements.

Meas No.	Bus No.	Meas No.	Bus No.	Meas No.	Bus No.
1	10	2	25	3	26
4	49	5	69	6	89

Voltage measurements.

Meas No.	Bus No.	Meas No.	Bus No.	Meas No.	Bus No.
1	1	2	4	3	5
4	6	5	8	6	9
7	10	8	12	9	15
10	17	11	18	12	19
13	24	14	25	15	26
16	27	17	30	18	31
19	32	20	34	21	36
22	37	23	38	24	40
25	42	26	43	27	44
28	46	29	49	30	54
31	55	32	56	33	59
34	61	35	62	36	64
37	65	38	66	39	68
40	69	41	70	42	72
43	73	44	74	45	76
46	77	47	80	48	81
49	82	50	85	51	87
52	89	53	90	54	91
55	92	56	99	57	100
58	103	59	104	60	105
61	107	62	110	63	111
64	113	65	116		

Generator power flow measurements.

Meas No.	Gen No.	Meas No.	Gen No.	Meas No.	Gen No.
1	1	2	2	3	3
4	4	5	5	6	6
7	7	8	8	9	9
10	10	11	11	12	12
13	13	14	14	15	15
16	16	17	17	18	18
19	19	20	20	21	21
22	22	23	23	24	24
25	25	26	26	27	27
28	28	29	29	30	30
31	31	32	32	33	33
34	34	35	35	36	36
37	37	38	38	39	39
40	40	41	41	42	42
43	43	44	44	45	45
46	46	47	47	48	48
49	49	50	50	51	51
52	52	53	53	54	54

Load power flow measurements.

Meas No.	Load No.	Meas No.	Load No.	Meas No.	Load No.
1	1	2	2	3	3
4	4	5	5	6	6
7	7	8	8	9	9
10	10	11	11	12	12
13	13	14	14	15	15
16	16	17	17	18	18
19	20	20	21	21	22
22	23	23	24	24	25
25	26	26	27	27	28
28	29	29	30	30	31
31	32	32	33	33	34
34	35	35	36	36	37
37	38	38	39	39	40
40	41	41	42	42	43
43	44	44	45	45	46
46	47	47	48	48	49
49	50	50	51	51	52
52	53	53	54	54	55
55	56	56	57	57	58
58	59	59	60	60	61
61	62	62	63	63	64
64	65	65	66	66	67
67	68	68	69	69	70
70	71	71	72	72	73
73	74	74	75	75	76
76	77	77	78	78	79
79	80	80	81	81	82
82	83	83	85	84	86
85	87	86	88	87	89
88	90	89	91		

Line power flow measurements.
(S) => measurement at the sending end of the line
(R) => measurement at the receiving end of the line

Meas No.	Line No.	Meas No.	Line No.	Meas No.	Line No.
1	1 (S)	2	2 (S)	3	3 (S)
4	3 (R)	5	4 (S)	6	4 (R)
7	5 (S)	8	5 (R)	9	6 (S)
10	7 (S)	11	7 (R)	12	8 (S)
13	8 (R)	14	9 (S)	15	9 (R)
16	10 (S)	17	10 (R)	18	11 (S)
19	11 (R)	20	12 (R)	21	13 (R)
22	14 (R)	23	15 (R)	24	17 (S)
25	19 (R)	26	20 (S)	27	21 (S)
28	21 (R)	29	23 (S)	30	23 (R)
31	24 (S)	32	24 (R)	33	25 (S)
34	25 (R)	35	26 (R)	36	30 (R)
37	31 (S)	38	31 (R)	39	33 (S)
40	33 (R)	41	34 (S)	42	37 (S)
43	37 (R)	44	38 (S)	45	38 (R)
46	39 (R)	47	40 (R)	48	41 (S)
49	41 (R)	50	42 (S)	51	42 (R)
52	43 (S)	53	45 (R)	54	47 (R)
55	48 (R)	56	49 (S)	57	50 (S)
58	50 (R)	59	51 (S)	60	52 (S)
61	52 (R)	62	53 (S)	63	53 (R)
64	54 (S)	65	54 (R)	66	56 (R)
67	57 (S)	68	57 (R)	69	58 (S)
70	59 (S)	71	59 (R)	72	60 (S)
73	60 (R)	74	61 (S)	75	61 (R)
76	62 (S)	77	62 (R)	78	63 (S)
79	64 (S)	80	64 (R)	81	65 (R)
82	66 (S)	83	66 (R)	84	67 (S)
85	67 (R)	86	68 (S)	87	68 (R)
88	69 (S)	89	69 (R)	90	70 (S)
91	70 (R)	92	72 (S)	93	73 (S)
94	74 (S)	95	74 (R)	96	75 (R)
97	76 (S)	98	77 (R)	99	80 (S)
100	80 (R)	101	81 (R)	102	82 (R)
103	83 (S)	104	83 (R)	105	84 (R)
106	85 (S)	107	85 (R)	108	86 (S)
109	86 (R)	110	87 (S)	111	87 (R)
112	90 (S)	113	91 (S)	114	91 (R)
115	93 (S)	116	93 (R)	117	94 (S)
118	94 (R)	119	95 (S)	120	95 (R)
121	96 (R)	122	99 (S)	123	99 (R)
124	100(S)	125	100(R)	126	101(S)
127	101(R)	128	102(S)	129	102(R)
130	104(S)	131	104(R)	132	105(R)
133	106(S)	134	107(S)	135	108(S)
136	109(S)	137	110(R)	138	111(S)
139	112(S)	140	112(R)	141	113(S)
142	113(R)	143	114(S)	144	114(R)
145	115(S)	146	115(R)	147	117(S)
148	117(R)	149	118(R)	150	119(S)
151	119(R)	152	120(S)	153	120(R)
154	121(S)	155	121(R)	156	123(R)
157	124(S)	158	124(R)	159	125(S)

160	126(S)	161	126(R)	162	130(S)
163	130(R)	164	131(S)	165	131(R)
166	132(S)	167	132(R)	168	133(S)
169	133(R)	170	135(S)	171	135(R)
172	137(S)	173	138(S)	174	140(S)
175	141(R)	176	142(S)	177	142(R)
178	143(S)	179	144(S)	180	145(S)
181	146(S)	182	147(R)	183	148(R)
184	149(R)	185	150(S)	186	151(R)
187	152(R)	188	153(S)	189	156(S)
190	156(R)	191	157(S)	192	159(S)
193	160(S)	194	160(R)	195	161(S)
196	161(R)	197	162(S)	198	163(S)
199	163(R)	200	164(S)	201	165(S)
202	165(R)	203	167(S)	204	167(R)
205	168(R)	206	169(S)	207	169(R)
208	170(S)	209	170(R)	210	172(S)
211	173(S)	212	174(S)	213	176(S)
214	176(R)				

Appendix 2

The solution of the generator differential equations together with both the generator and the network algebraic equations is in effect a transient stability analysis of the system. However the overall on-line control package is only concerned with time scales greater than one second, hence the fast transient behaviour of the system is not required and a time step of approximately one second is sufficient. A full explanation of the procedure for solving the differential and algebraic equations simultaneously can be found in a paper by Dommel and Sato.[35] A brief explanation of the method is presented in this appendix together with the elements of the Jacobian matrix used in the Newton Raphson method for solving equations.

A2.1 System Equations

The equations used to model the generators and the network are identical to those in chapters 8 and 10 of Stagg and El-Abiad.[124]
The generator differential equations can be written as follows

$$\frac{d^2\delta}{dt^2} = \dot{\omega} = \frac{\pi F(P_m - P_e)}{H} \qquad (A2.1)$$

$$\delta^{\cdot} = \frac{d\theta_e}{dt} - \omega^0 = \omega - 2\pi F \qquad (A2.2)$$

$$\frac{d P_m}{dt} = (P_m)^{\cdot} = \frac{P_{set} + G_g(F_{set} - \omega/2\pi) - P_m}{T_c} \qquad (A2.3)$$

where

δ = electrical angular position in radians of the generator rotor with respect to a synchronously rotating reference axis.
ω = rate of change of the electrical angle in radians/second.
ω^0 = rated synchronous speed in radians/second.
F = frequency of the system.
H = inertia constant of the generator.
P_m = mechanical power input to the generator.
P_e = electrical power output of the generator.
θ_e = electrical angle of the generator (mechanical angle * number of pairs of poles).

P_{set} = present power set point.
F_{set} = present frequency set point.
G_g = generator governor gain.
T_c = steam time constant.

The generator algebraic equations can be written as follows

$$P_e = \text{Real part } (I_t(E')^*) \tag{A2.4}$$

$$e' = |E'| \cos(\delta) \tag{A2.5}$$

$$f' = |E'| \sin(\delta) \tag{A2.6}$$

$$I_t = \frac{E' - E}{r_a - j\, x'_d} \tag{A2.7}$$

where
E' = $e' + j\, f'$ =
complex voltage behind the generator transient reactance.
E = $e + j\, f$ =
complex terminal (bus) voltage of the generator.
I_t = $I_a + j\, I_r$ =
complex current flowing through the generator transient reactance.
r_a = resistance associated with the transient reactance.
x'_d = transient reactance of the generator.

The algebraic equations for the network are based on Kirchoff's first law and equate the sum of all the currents flowing into a node to zero, the currents are expressed in terms of the bus admittance matrix and the nodal voltages as follows

$$\sum_{i=1}^{j} y_{ij} e_{t_j} = 0 \quad \text{for all } j \tag{A2.8}$$

where
j = total number of electrical nodes in the system, including those behind the generator transient reactance.
Y = the bus admittance matrix modified to include the generator transient

reactance and the equivalent shunt admittance used to represent the loads.

E_t = vector of all complex voltages throughout the system, including those behind the generator transient reactance.

The generator differential equations can be written in matrix form as shown below.

$$W^{\cdot} = AW + BX + C \qquad \text{(A2.9)}$$

The symbols $W^{\cdot}$, W and X represent column vectors which contain the generator differential variables, the mechanical states of the generator and the electrical states of both the generators and the network respectively. The structures of these vectors are shown below

$$W^{\cdot} \equiv [\omega^{\cdot}_1,\ \delta^{\cdot}_1,\ P_{m}^{\circ}{}_1,\ \ldots,\ \omega^{\cdot}_g,\ \delta^{\cdot}_g,\ P_{m}^{\circ}{}_g]$$

$$W \equiv [\omega_1,\ \delta_1,\ P_{m_1},\ \ldots,\ \omega_g,\ \delta_g,\ P_{m_g}]$$

$$X \equiv [I_{a_1},\ I_{r_1},\ e'_1,\ f'_1,\ P_{e_1},\ \ldots,\ I_{a_g},\ I_{r_g},\ e'_g,\ f'_g,\ P_{e_g},$$

$$e_1,\ f_1,\ \ldots,\ e_n,\ f_n]$$

where

g = number of generators.

n = number of electrical nodes in the network (excluding the nodes behind the generator transient reactance).

The column vector C stores the constants of the generator differential equations. It should be noted that the generator differential equations A2.9 maybe re-arranged with the left hand side of the equation equal to zero. The network and generator algebraic equations maybe combined into a single equation with the left hand side equal to zero as shown below

$$DW + EX = 0 \qquad \text{(A2.10)}$$

The matrixes A, B, C, D and E are highly sparse and full use of this sparsity will enhance the performance of the program by reducing both the storage space and the CPU time required to obtain the solution of the

equations.

A2.2 Implicit Trapezoidal Integration

Given a set of linear differential equations of the form

$$W^{\cdot} = AW + BX + C \tag{A2.11}$$

then a step by step integration can be used to calculate the current value of W as shown below

$$W(t) = W(t-\Delta t) + A\int_{t-\Delta t}^{t} W\,d\tau + B\int_{t-\Delta t}^{t} X\,d\tau + C\int_{t-\Delta t}^{t} d\tau \tag{A2.12}$$

If the assumption that W and X vary linearly over the period $t-\Delta t$ then using the trapezodial rule of integration equation A2.12 becomes

$$W(t) = W(t-\Delta t) + \frac{\Delta t}{2}A[W(t-\Delta t) + W(t)] + \frac{\Delta t}{2}B[X(t-\Delta t) + X(t)] + C\,\Delta t \tag{A2.13}$$

re-arranging the above equation gives

$$[U - \frac{\Delta t}{2}A]W(t) - \frac{\Delta t}{2}BX(t) - F_0(t-\Delta t) - C\,\Delta t = 0 \tag{A2.14}$$

where

U = unit matrix.

$F_0(t-\Delta t)$ = function evaluated from the values of W and X at the previous time step, as shown below.

$$F_0(t-\Delta t) = [U + \frac{\Delta t}{2}A]W(t-\Delta t) + \frac{\Delta t}{2}BX(t-\Delta t) \tag{A2.15}$$

A2.3 The formation of the Jacobian Matrix

Applying the trapezodial integration method to the generator differential equations A2.1, A2.2 and A2.3 the following three equations can be derived.

$$F_1 = \delta(t) - \frac{\Delta t \pi FP_m(t)}{2H} + \frac{\Delta t \pi FP_e(t)}{2H} - \delta(t-\Delta t) \qquad \text{(A2.16)}$$

$$- \frac{\Delta t \pi FP_m(t-\Delta t)}{2H} + \frac{\Delta t \pi FP_e(t-\Delta t)}{2H} = 0$$

$$F_2 = \delta(t) - \frac{\Delta t \omega(t)}{2} + \Delta t 2\pi F - \delta(t-\Delta t) - \frac{\Delta t \omega(t-\Delta t)}{2} = 0 \qquad \text{(A2.17)}$$

$$F_3 = P_m(t)[1 + \frac{\Delta t}{2}] + \frac{\Delta t G_g \omega(t)}{4\pi T_c} - \frac{\Delta t(P_{set} - G_g F_{set})}{T_c} - \qquad \text{(A2.18)}$$

$$P_m(t-\Delta t) + \frac{\Delta t G_g \omega(t-\Delta t)}{4\pi T_c} + \frac{\Delta t P_m(t-\Delta t)}{2T_c} = 0$$

Equation A2.16 has the following non-zero partial derivatives

$$\frac{\delta F_1}{\delta \omega} = 1.0 \qquad \frac{\delta F_1}{\delta P_m} = \frac{-\Delta t \pi F}{2H} \qquad \frac{\delta F_1}{\delta P_e} = \frac{\Delta t \pi F}{2H}$$

Equation A2.17 has the following non-zero partial derivatives

$$\frac{\delta F_2}{\delta \delta} = 1.0 \qquad \frac{\delta F_2}{\delta \omega} = \frac{-\Delta t}{2}$$

Equation A2.18 has the following non-zero partial derivatives

$$\frac{\delta F_3}{\delta P_m} = 1.0 + \frac{\Delta t}{2T_c} \qquad \frac{\delta F_3}{\delta \omega} = \frac{\Delta t G_g}{4\pi T_c}$$

The generator algebraic equations, A2.4, A2.5, A2.6, and A2.7 may be re-arranged and the non-zero partial deratives obtained as shown below. Equation A2.4 becomes

$$F_4 = P_e - \text{real part}[I_t(E')^*] = 0 \qquad \text{(A2.19)}$$
$$= P_e - \text{real part}[(I_a + j\, I_r)(e' - j\, f')] = 0$$

$$= P_e - I_a e' - I_r f' = 0$$

hence

$$\frac{\delta F_4}{\delta P_e} = 1.0 \qquad \frac{\delta F_4}{\delta I_a} = -e' \qquad \frac{\delta F_4}{\delta e'} = -I_a \qquad \frac{\delta F_4}{\delta I_r} = -f' \qquad \frac{\delta F_4}{\delta f'} = -I_r$$

Equation A2.5 becomes

$$F_5 = e' - |E| \cos(\delta) = 0 \tag{A2.20}$$

hence

$$\frac{\delta F_5}{\delta e'} = 1.0 \qquad \frac{\delta F_5}{\delta \delta} = E \sin(\delta)$$

Equation A2.6 becomes

$$F_6 = f' - |E| \sin(\delta) = 0 \tag{A2.21}$$

hence

$$\frac{\delta F_6}{\delta f'} = 1.0 \qquad \frac{\delta F_6}{\delta \delta} = -E \cos(\delta)$$

Equation A2.7 relates the current flowing in the generators equivalent circuit to the voltage behind the transient reactance, the terminal voltage and the impedance of the equivalent circuit. In the Jacobian matrix the equation is separated into its real and imaginary components as follows

$$I_t = \frac{E' - E}{r_a + j\, x'_d} \tag{A2.22}$$

$$= I_a + j\, I_r = \frac{e' + j\, f' - e - j\, f}{r_a + j\, x'_d}$$

Equating the real parts of the equation A2.22 gives

$$F_7 = I_a - \frac{e' r_a - f' x'_d + e r_a + f x'_d}{(r_a)^2 + (x'_d)^2} = 0 \tag{A2.23}$$

hence

$$\frac{\delta F_7}{\delta I_a} = 1.0 \qquad \frac{\delta F_7}{\delta e'} = \frac{-r_a}{(r_a)^2 + (x'_d)^2} \qquad \frac{\delta F_7}{\delta f'} = \frac{-x'_d}{(r_a)^2 + (x'_d)^2}$$

$$\frac{\delta F_7}{\delta e} = \frac{r_a}{(r_a)^2 + (x'_d)^2} \qquad \frac{\delta F_7}{\delta f} = \frac{x'_d}{(r_a)^2 + (x'_d)^2}$$

Equating the imaginary parts of the equation A2.22 gives

$$F_8 = I_r - \frac{f'r_a + e'x'_d - ex'_d + fr_a}{(r_a)^2 + (x'_d)^2} = 0 \qquad \text{(A2.24)}$$

hence

$$\frac{\delta F_8}{\delta I_r} = 1.0 \qquad \frac{\delta F_8}{\delta e'} = \frac{x'_d}{(r_a)^2 + (x'_d)^2} \qquad \frac{\delta F_8}{\delta f'} = \frac{-r_a}{(r_a)^2 + (x'_d)^2}$$

$$\frac{\delta F_8}{\delta e} = \frac{-x'_d}{(r_a)^2 + (x'_d)^2} \qquad \frac{\delta F_8}{\delta f} = \frac{r_a}{(r_a)^2 + (x'_d)^2}$$

The network algebraic equations are defined by the matrix equations A2.8. The number of these equations depends on the current number of electrical nodes in the system. The number of non-zero terms in the rows of the bus admittance matrix depends on the number of electrical elements connected to the node. Each equation is separated into its real and imaginary components and the partial derivatives of each term is then evaluated as shown below. Writing the bus admittance matrix elements in terms of a resistance and reactance gives

$$y_{ij} = \frac{1}{r_{ij} + j\,x_{ij}} = \frac{r_{ij}}{(r_{ij})^2 + (x_{ij})^2} - j\,\frac{x_{ij}}{(r_{ij})^2 + (x_{ij})^2} \qquad \text{(A2.25)}$$

Writing the nodal voltage of node j as E_j

$$E_j = e_j + j\,f_j \qquad \text{(A2.26)}$$

and separating the real and imaginary components of the product of equations

A2.25 and A2.26 gives the following equations and partial derivatives

$$F_9 = \frac{e_j r_{ij}}{(r_{ij})^2 + (x_{ij})^2} + \frac{f_j x_{ij}}{(r_{ij})^2 + (x_{ij})^2} = 0 \qquad \text{(A2.27)}$$

hence

$$\frac{\delta F_9}{\delta e_j} = \frac{r_{ij}}{(r_{ij})^2 + (x_{ij})^2} \qquad \frac{\delta F_9}{\delta f_j} = \frac{x_{ij}}{(r_{ij})^2 + (x_{ij})^2}$$

and

$$F_{10} = \frac{f_j r_{ij}}{(r_{ij})^2 + (x_{ij})^2} - \frac{e_j x_{ij}}{(r_{ij})^2 + (x_{ij})^2} = 0 \qquad \text{(A2.28)}$$

hence

$$\frac{\delta F_{10}}{\delta e_j} = \frac{-x_{ij}}{(r_{ij})^2 + (x_{ij})^2} \qquad \frac{\delta F_{10}}{\delta f_j} = \frac{r_{ij}}{(r_{ij})^2 + (x_{ij})^2}$$

The elements of the Jacobian matrix are re-computed every iteration even though some of them remain constant. This is done to facilitate the programming of the procedure and has little effect on the solution times, the majority of the time being taken up in the solution of the linear set of equations used to compute the change in the value of the variables.

Appendix Three

Appendix three lists the steady state starting voltage and power flows for the 30, the 57, and the 118 substation test networks.

Values are in P.U.

Initial voltage levels and power flows for the 30 substation test network

V = voltage magnitude, θ = voltage phase angle,
P = active power flow, Q = reactive power flow,
R = receiving end of a line, S = sending end of a line.

Voltage levels for the 30 substation test network.

Bus No.	Node No.	V	θ
1	1	1.0438	0.0000
2	2	1.0301	-0.0375
3	2	1.0301	-0.0375
4	3	1.0105	-0.0728
5	4	1.0025	-0.0870
6	4	1.0025	-0.0870
7	5	0.9851	-0.1310
8	5	0.9851	-0.1310
9	5	0.9851	-0.1310
10	6	0.9951	-0.1028
11	6	0.9951	-0.1028
12	6	0.9951	-0.1028
13	6	0.9951	-0.1028
14	6	0.9951	-0.1028
15	6	0.9951	-0.1028
16	7	0.9829	-0.1241
17	8	0.9894	-0.1067
18	8	0.9894	-0.1067
19	8	0.9894	-0.1067
20	8	0.9894	-0.1067
21	8	0.9894	-0.1067
22	8	0.9894	-0.1067
23	8	0.9894	-0.1067
24	8	0.9894	-0.1067
25	9	1.0071	-0.1317
26	10	0.9928	-0.1690
27	10	0.9928	-0.1690
28	10	0.9928	-0.1690
29	10	0.9928	-0.1690
30	10	0.9928	-0.1690
31	10	0.9928	-0.1690
32	10	0.9928	-0.1690
33	10	0.9928	-0.1690
34	10	0.9928	-0.1690
35	10	0.9928	-0.1690
36	11	1.0486	-0.0922

37	12	0.9889	-0.1482
38	12	0.9889	-0.1482
39	13	1.0011	-0.1199
40	14	0.9752	-0.1666
41	15	0.9721	-0.1698
42	15	0.9721	-0.1698
43	15	0.9721	-0.1698
44	15	0.9721	-0.1698
45	16	0.9828	-0.1626
46	17	0.9841	-0.1712
47	18	0.9664	-0.1832
48	19	0.9664	-0.1874
49	20	0.9722	-0.1839
50	21	0.9790	-0.1780
51	22	0.9793	-0.1778
52	23	0.9646	-0.1798
53	24	0.9637	-0.1864
54	24	0.9637	-0.1864
55	25	0.9602	-0.1855
56	26	0.9415	-0.1938
57	27	0.9673	-0.1799
58	27	0.9673	-0.1799
59	27	0.9673	-0.1799
60	27	0.9673	-0.1799
61	27	0.9673	-0.1799
62	27	0.9673	-0.1799
63	28	0.9899	-0.1111
64	29	0.9462	-0.2040
65	29	0.9462	-0.2040
66	29	0.9462	-0.2040
67	29	0.9462	-0.2040
68	29	0.9462	-0.2040
69	30	0.9340	-0.2213
70	30	0.9340	-0.2213
71	30	0.9340	-0.2213
72	30	0.9340	-0.2213
73	30	0.9340	-0.2213

Generator power flows for the 30 substation test network.

Gen No.	P	Q
1	1.1469	0.0689
2	0.7502	0.3254
3	0.4009	0.1129
4	0.2003	0.1672
5	0.2001	0.2133
6	0.2002	0.0897

Load power flows for the 30 substation test network.

Load No.	P	Q
1	-0.2170	-0.1270
2	-0.0240	-0.0120
3	-0.0760	-0.0160
4	-0.3790	-0.0760
5	-0.3290	-0.0660
6	-0.2340	-0.0480

7	-0.2280	-0.1090
8	-0.3000	-0.3000
9	-0.0580	-0.0200
10	-0.1120	-0.0750
11	-0.0620	-0.0160
12	-0.0820	-0.0250
13	-0.0350	-0.0180
14	-0.0900	-0.0580
15	-0.0320	-0.0090
16	-0.0950	-0.0340
17	-0.0220	-0.0070
18	-0.1750	-0.1120
19	-0.0320	-0.0160
20	-0.0870	-0.0670
21	-0.0350	-0.0230
22	-0.0240	-0.0090
23	-0.1060	-0.0190
24	0.0000	0.1900
25	0.0000	0.0430

Link power flows for the 30 substation test network.

Link No.	P	Q
1	-0.4702	-0.1191
2	-0.4702	-0.1191
3	0.0904	-0.0040
4	-0.1277	-0.0335
5	-0.1277	-0.0335
6	-0.2643	-0.0732
7	0.1366	0.0397
8	0.0793	0.0018
9	0.0753	0.0320
10	-0.0483	-0.0056
11	-0.2237	-0.0362
12	-0.1276	-0.0073
13	-0.0040	0.0302
14	0.0720	0.0343
15	0.1236	0.0376
16	-0.1754	-0.0306
17	0.0760	0.0040
18	-0.0593	-0.0518
19	0.0000	0.0000
20	0.1500	0.1500
21	-0.0907	-0.0983
22	0.0593	0.0518
23	0.1500	0.1500
24	-0.0907	-0.0983
25	-0.1001	-0.0836
26	0.0000	0.0000
27	0.0094	-0.0146
28	0.1001	0.0836
29	0.0094	-0.0146
30	-0.0778	-0.0832
31	-0.0321	-0.0997
32	0.1378	0.1132
33	-0.0285	0.0068
34	-0.1101	-0.0433

35	0.0590	0.0564
36	0.0729	0.0530
37	0.0213	0.0031
38	-0.0457	0.0165
39	-0.1699	-0.0230
40	-0.0718	-0.0167
41	-0.1692	-0.0996
42	-0.1625	-0.0267
43	-0.1625	-0.0267
44	-0.0082	-0.0060
45	-0.0649	-0.0038
46	-0.0294	-0.0054
47	-0.0437	-0.0045
48	-0.0448	-0.0163
49	-0.0448	-0.0163
50	0.0824	0.0223
51	0.0204	0.0054
52	0.0507	0.0115
53	-0.0824	-0.0224
54	0.0838	0.0270
55	-0.0507	-0.0115
56	0.0185	0.0031
57	-0.0305	-0.0076
58	0.0120	0.0045
59	0.0000	0.0000
60	-0.0185	-0.0031
61	0.0305	0.0076
62	-0.0120	-0.0045
63	-0.0346	-0.0068
64	-0.0346	-0.0068
65	0.0530	0.0095
66	0.0530	0.0095
67	-0.0184	-0.0027
68	-0.0184	-0.0027

Line power flows for the 30 substation test network.

Line No.	P (S)	P (R)	Q (S)	Q (R)
1	-0.7092	0.7003	0.0045	0.0257
2	-0.4378	0.4294	-0.0735	0.0824
3	-0.3164	0.3107	-0.0477	0.0686
4	-0.4054	0.4032	-0.0704	0.0726
5	-0.5101	0.4977	-0.1130	0.1034
6	-0.4071	0.3977	-0.0635	0.0731
7	-0.4011	0.3991	-0.0646	0.0667
8	0.0434	-0.0435	-0.0263	0.0457
9	-0.2737	0.2715	-0.0532	0.0633
10	-0.1188	0.1186	-0.0957	0.1036
11	-0.1389	0.1389	0.0554	-0.0601
12	-0.1174	0.1174	-0.0080	0.0002
13	0.2001	-0.2001	0.1972	-0.2133
14	-0.3390	0.3390	-0.1371	0.1226
15	-0.2369	0.2369	-0.0606	0.0454
16	0.2002	-0.2002	0.0830	-0.0897
17	-0.0771	0.0763	-0.0167	0.0151
18	-0.1785	0.1763	-0.0387	0.0342
19	-0.0695	0.0690	0.0020	-0.0030

20	-0.0143	0.0143	0.0009	-0.0009
21	-0.0340	0.0339	0.0210	-0.0213
22	-0.0567	0.0563	0.0022	-0.0030
23	-0.0243	0.0242	0.0120	-0.0121
24	0.0708	-0.0710	0.0461	-0.0466
25	-0.0942	0.0930	-0.0561	0.0536
26	-0.0564	0.0561	-0.0802	0.0793
27	-0.1662	0.1648	-0.1064	0.1035
28	-0.0817	0.0810	-0.0500	0.0486
29	0.0102	-0.0102	0.0085	-0.0085
30	-0.0519	0.0516	-0.0105	0.0099
31	-0.0708	0.0700	-0.0401	0.0388
32	-0.0196	0.0195	0.0061	-0.0062
33	-0.0025	0.0025	-0.0086	0.0086
34	-0.0355	0.0350	-0.0238	0.0230
35	0.0330	-0.0331	0.0152	-0.0155
36	-0.1663	0.1663	-0.0622	0.0495
37	-0.0621	0.0611	-0.0170	0.0152
38	-0.0711	0.0693	-0.0170	0.0136
39	-0.0371	0.0367	-0.0062	0.0054
40	-0.0188	0.0188	0.0293	0.0126
41	-0.1479	0.1475	-0.0383	0.0496

Initial voltage levels and power flows for the 57 substation test network.

V = voltage magnitude, θ = voltage phase angle,
P = active power flow, Q = reactive power flow,
R = receiving end of a line, S = sending end of a line.

Voltage levels for the 57 substation test network.

Bus No.	Node No.	V	θ
1	1	1.0490	0.0000
2	2	1.0401	-0.0072
3	3	1.0336	-0.1005
4	4	1.0413	-0.1396
5	5	1.0581	-0.1863
6	6	1.0718	-0.2014
7	7	1.0672	-0.2124
8	8	1.0825	-0.1859
9	9	1.0753	-0.2344
10	10	1.0622	-0.2414
11	11	1.0522	-0.2294
12	12	1.0834	-0.2105
13	13	1.0461	-0.2118
14	14	1.0323	-0.1862
15	15	1.0385	-0.1266
16	16	1.0642	-0.1746
17	17	1.0477	-0.1058
18	18	1.0271	-0.2155
19	19	0.9662	-0.2359
20	20	0.9651	-0.2530
21	21	0.9855	-0.2560
22	22	0.9883	-0.2584
23	23	0.9876	-0.2606
24	24	0.9910	-0.2830
25	25	0.9853	-0.3707
26	26	0.9940	-0.2788
27	27	1.0243	-0.2669
28	28	1.0419	-0.2545
29	29	1.0566	-0.2458
30	30	0.9620	-0.3788
31	31	0.9268	-0.3869
32	32	0.9284	-0.3652
33	33	0.9260	-0.3659
34	34	0.9489	-0.2871
35	35	0.9547	-0.2821
36	36	0.9635	-0.2767
37	37	0.9704	-0.2718
38	38	0.9903	-0.2549
39	39	0.9691	-0.2730
40	40	0.9626	-0.2783
41	41	1.0121	-0.2964
42	42	0.9726	-0.3153
43	43	1.0398	-0.2491
44	44	1.0003	-0.2331
45	45	1.0336	-0.1752
46	46	1.0208	-0.2202
47	47	1.0000	-0.2491
48	48	0.9961	-0.2530

49	49	1.0054	-0.2604
50	50	1.0022	-0.2723
51	51	1.0491	-0.2616
52	52	1.0259	-0.2718
53	53	1.0157	-0.2829
54	54	1.0363	-0.2714
55	55	1.0658	-0.2542
56	56	0.9642	-0.3182
57	57	0.9554	-0.3250

Generator power flows for the 57 substation test network.

Gen No.	P	Q
1	4.3347	-0.1969
2	0.7727	0.2899
3	0.8057	-0.2624
4	0.2688	0.5652
5	3.1818	0.3068
6	0.1936	0.8760
7	3.3856	1.5962

Load power flows for the 57 substation test network.

Load No.	P	Q
1	-0.5500	-0.1700
2	-0.0300	-0.8800
3	-0.4100	-0.2100
4	-0.1300	-0.0400
5	-0.7500	-0.0200
6	-1.5000	-0.2200
7	-1.2100	-0.2600
8	-0.0500	-0.0200
9	-3.7700	-0.2400
10	-0.1800	-0.0230
11	-0.1050	-0.0530
12	-0.2200	-0.0500
13	-0.4300	-0.0300
14	-0.4200	-0.0800
15	-0.2720	-0.0980
16	-0.0330	-0.0600
17	-0.0230	-0.0100
18	-0.0630	-0.0210
19	-0.0630	-0.0320
20	-0.0930	-0.0050
21	-0.0460	-0.0230
22	-0.1700	-0.0260
23	-0.0360	-0.0180
24	-0.0580	-0.0290
25	-0.0160	-0.0080
26	-0.0380	-0.0190
27	-0.0600	-0.0300
28	-0.1400	-0.0700
29	-0.0630	-0.0300
30	-0.0710	-0.0440
31	-0.0200	-0.0100
32	-0.1200	-0.0180
33	-0.2970	-0.1160

34	-0.1800	-0.0850
35	-0.2100	-0.1050
36	-0.1800	-0.0530
37	-0.0490	-0.0220
38	-0.2000	-0.1000
39	-0.0410	-0.0140
40	-0.0680	-0.0340
41	-0.0760	-0.0220
42	-0.0670	-0.0200
43	-0.0950	-0.0340
44	0.0000	0.1000
45	0.0000	0.0900
46	0.0000	0.0630

Line power flows for the 57 substation test network.

Line No.	P (S)	P (R)	Q (S)	Q (R)
1	-0.3500	0.3487	-0.1612	0.2975
2	-1.0914	1.0569	0.2929	-0.3034
3	-0.9945	0.9815	0.5189	-0.5204
4	-0.2706	0.2627	0.2656	-0.2538
5	-0.3761	0.3667	0.3285	-0.3220
6	-0.1276	0.1273	-0.0082	0.0382
7	0.1123	-0.1127	0.0704	-0.0180
8	-1.1107	1.1003	0.0685	-0.0580
9	-0.0626	0.0623	-0.0447	0.0937
10	-0.0203	0.0185	-0.2740	0.2926
11	0.0961	-0.0966	0.0519	0.0357
12	0.0916	-0.0941	-0.2051	0.2425
13	0.4886	-0.4943	-0.4817	0.4747
14	0.0605	-0.0872	-0.3115	0.3279
15	-1.4966	1.4603	0.1299	-0.2080
16	-0.8989	0.8640	0.2231	-0.3204
17	-1.0391	1.0152	0.1752	-0.2522
18	-0.4580	0.4541	0.2569	-0.2115
19	-0.3349	0.3349	-0.0736	0.0473
20	-0.1327	0.1300	0.2938	-0.2854
21	0.4553	-0.4580	0.1456	-0.1373
22	0.3049	-0.3075	0.1253	-0.0993
23	0.2173	-0.2188	-0.1464	0.1624
24	-0.2184	0.2117	-0.5942	0.6408
25	0.4291	-0.4339	-0.3472	0.3504
26	0.5789	-0.5951	-0.3512	0.3322
27	1.0864	-1.1064	-0.2509	0.2028
28	-0.0628	0.0600	-0.0493	0.0452
29	-0.0270	0.0267	0.0148	-0.0153
30	-0.0037	0.0037	0.0253	-0.0258
31	-0.0037	0.0036	0.0258	-0.0259
32	-0.1180	0.1179	0.0309	-0.0311
33	-0.0548	0.0539	0.0521	-0.0453
34	-0.1418	0.1418	-0.0156	0.0031
35	0.0879	-0.0879	0.0609	-0.0615
36	0.0879	-0.0898	0.0615	-0.0644
37	0.1829	-0.1851	0.0694	-0.0729
38	0.2311	-0.2335	0.0959	-0.0993
39	-0.5826	0.5826	-0.1838	0.1626
40	-0.0788	0.0774	-0.0612	0.0591

41	-0.0414	0.0402	-0.0411	0.0393
42	0.0178	-0.0181	-0.0102	0.0099
43	-0.0381	0.0380	-0.0191	0.0190
44	0.0722	-0.0722	0.0172	-0.0233
45	0.0722	-0.0725	0.0233	-0.0209
46	0.1325	-0.1335	0.0509	-0.0506
47	0.1645	-0.1654	0.0500	-0.0511
48	0.2009	-0.2040	0.0619	-0.0647
49	-0.0355	0.0355	-0.0108	0.0107
50	-0.0310	0.0310	0.0007	-0.0007
51	0.1144	-0.1146	-0.0050	0.0046
52	-0.0951	0.0951	-0.0595	0.0510
53	-0.0892	0.0868	-0.0616	0.0576
54	0.1207	-0.1207	0.0652	-0.0728
55	0.3621	-0.3660	-0.0125	0.0067
56	-0.5008	0.5008	-0.0611	0.0365
57	-0.4870	0.4870	-0.1707	0.1524
58	-0.4870	0.4812	-0.1524	0.1386
59	-0.1841	0.1835	-0.0226	0.0218
60	-0.0079	0.0074	0.0790	-0.0750
61	-0.0782	0.0776	0.0238	-0.0246
62	0.1324	-0.1372	0.1297	-0.1372
63	-0.3172	0.3172	-0.1990	0.1902
64	-0.2678	0.2678	-0.2294	0.2077
65	-0.1790	0.1747	-0.0373	0.0317
66	-0.1257	0.1245	-0.0097	0.0082
67	0.0755	-0.0767	0.0288	-0.0303
68	0.1177	-0.1203	0.0443	-0.0476
69	-0.1407	0.1407	-0.0866	0.0828
70	0.4860	-0.5008	0.0114	-0.0366
71	-0.0310	0.0310	0.0007	-0.0019
72	-0.0636	0.0611	-0.0246	0.0221
73	-0.0158	0.0157	-0.0136	0.0134
74	-0.0355	0.0355	-0.0107	0.0087
75	-0.0317	0.0315	-0.0116	0.0113
76	0.0163	-0.0170	0.0764	-0.0715
77	0.0802	-0.0806	0.0662	-0.0668
78	-0.1883	0.1883	-0.0861	0.0816

Initial voltage levels and power flows for the 118 substation test network.

V = voltage magnitude, θ = voltage phase angle,
P = active power flow, Q = reactive power flow,
R = receiving end of a line, S = sending end of a line.

Voltage levels for the 118 substation test network.

Bus No.	Node No.	V	θ
1	1	0.9029	0.0000
2	2	0.9090	-0.0011
3	3	0.9175	0.0068
4	4	0.9584	0.0647
5	5	0.9669	0.0689
6	6	0.9260	0.0336
7	7	0.9234	0.0225
8	8	1.0373	0.1416
9	9	1.0956	0.2241
10	10	1.0811	0.3139
11	11	0.9287	0.0196
12	12	0.9208	0.0117
13	13	0.9188	-0.0012
14	14	0.9258	0.0042
15	15	0.9383	0.0187
16	16	0.9319	0.0089
17	17	0.9766	0.0505
18	18	0.9359	0.0317
19	19	0.9317	0.0224
20	20	0.9437	0.0240
21	21	0.9546	0.0425
22	22	0.9748	0.0766
23	23	1.0127	0.1459
24	24	0.9928	0.1618
25	25	1.0856	0.2092
26	26	1.0902	0.2184
27	27	0.9623	0.0824
28	28	0.9454	0.0586
29	29	0.9336	0.0492
30	30	1.0481	0.1059
31	31	0.9324	0.0551
32	32	0.9548	0.0775
33	33	0.9484	0.0012
34	34	0.9640	0.0118
35	35	0.9574	0.0085
36	36	0.9555	0.0104
37	37	0.9746	0.0163
38	38	1.0428	0.0664
39	39	0.9517	-0.0021
40	40	0.9484	0.0038
41	41	0.9443	-0.0071
42	42	0.9573	0.0155
43	43	0.9618	-0.0406
44	44	0.9658	-0.0678
45	45	0.9620	-0.0599
46	46	0.9698	-0.0173
47	47	0.9921	0.0046
48	48	0.9845	-0.0121

49	49	0.9873	0.0014
50	50	0.9673	-0.0417
51	51	0.9377	-0.0973
52	52	0.9291	-0.1172
53	53	0.9196	-0.1401
54	54	0.9265	-0.1259
55	55	0.9257	-0.1245
56	56	0.9272	-0.1238
57	57	0.9419	-0.0966
58	58	0.9315	-0.1143
59	59	0.9535	-0.0767
60	60	0.9898	-0.0054
61	61	0.9942	0.0083
62	62	0.9931	0.0090
63	63	0.9953	0.0001
64	64	1.0180	0.0358
65	65	1.0683	0.1141
66	66	1.0558	0.0893
67	67	1.0219	0.0389
68	68	1.0798	0.1302
69	69	1.0723	0.1599
70	70	0.9923	0.1237
71	71	0.9854	0.1408
72	72	0.9767	0.1766
73	73	0.9757	0.1541
74	74	0.9750	0.0880
75	75	0.9919	0.0893
76	76	0.9689	0.0660
77	77	1.0284	0.0981
78	78	1.0257	0.0908
79	79	1.0311	0.0916
80	80	1.0594	0.1176
81	81	1.0838	0.1256
82	82	1.0206	0.0814
83	83	1.0210	0.0931
84	84	1.0194	0.1226
85	85	1.0242	0.1426
86	86	1.0133	0.1425
87	87	1.0076	0.1816
88	88	1.0317	0.1655
89	89	1.0482	0.2121
90	90	1.0183	0.1822
91	91	1.0221	0.1810
92	92	1.0269	0.1546
93	93	1.0141	0.1187
94	94	1.0087	0.0952
95	95	1.0030	0.0819
96	96	1.0186	0.0830
97	97	1.0353	0.0942
98	98	1.0339	0.0888
99	99	1.0206	0.1169
100	100	1.0050	0.0971
101	101	1.0042	0.1067
102	102	1.0192	0.1370
103	103	0.9900	0.0837
104	104	0.9779	0.0670
105	105	0.9779	0.0608
106	106	0.9761	0.0494

107	107	0.9773	0.0521
108	108	0.9740	0.0633
109	109	0.9721	0.0651
110	110	0.9699	0.0761
111	111	0.9755	0.0939
112	112	0.9652	0.0766
113	113	0.9749	0.0620
114	114	0.9535	0.0694
115	115	0.9536	0.0689
116	116	1.0754	0.1324
117	117	0.9053	-0.0199
118	118	0.9751	0.0707

Generator power flows for the 118 substation test network.

Gen No.	P	Q
1	0.2939	-0.0612
2	0.2749	-0.5259
3	0.2789	-0.3154
4	0.2740	-0.8421
5	3.2508	-2.0408
6	0.2859	-0.2879
7	0.2869	-0.2838
8	0.2779	-0.2723
9	0.2799	-0.2638
10	0.2369	-0.5772
11	1.6888	1.0626
12	1.8210	-0.4822
13	0.2579	-0.3154
14	0.2689	-0.2551
15	0.2609	-0.2985
16	0.3099	-0.5556
17	0.2979	-0.3043
18	0.3009	-0.0603
19	0.3069	-0.2740
20	0.3219	-0.2985
21	0.7338	-0.6447
22	0.3729	-0.0050
23	0.3709	-0.0029
24	0.3719	-0.0064
25	0.6208	-0.0905
26	0.5489	-0.5952
27	0.3139	-0.3526
28	3.0596	-0.8862
29	2.8897	1.3664
30	4.0978	0.8343
31	0.2489	-0.3114
32	0.2199	-0.2665
33	0.2329	-0.2620
34	0.2649	-0.2958
35	0.2699	-0.0613
36	0.2649	-0.4276
37	3.3358	1.0987
38	0.2129	-0.1399
39	0.1859	-0.0945
40	3.4247	0.4848
41	0.2109	-0.1294

42	0.1889	-0.1028
43	0.2039	-0.1396
44	0.2349	-0.1259
45	0.5458	-0.2283
46	0.2389	-0.0386
47	0.2459	-0.0182
48	0.2479	-0.0188
49	0.2519	-0.0186
50	0.2359	-0.0015
51	0.2250	-0.0109
52	0.2320	0.0084
53	0.2779	-0.0976
54	0.5378	-1.3808

Load power flows for the 118 substation network.

Load No.	P	Q
1	-0.5100	-0.2700
2	-0.2000	-0.0900
3	-0.3900	-0.1000
4	-0.3000	-0.1200
5	-0.5200	-0.2200
6	-0.1900	-0.0200
7	-0.7000	-0.2300
8	-0.4700	-0.1000
9	-0.3400	-0.1600
10	-0.1400	-0.0100
11	-0.9000	-0.3000
12	-0.2500	-0.1000
13	-0.1100	-0.0300
14	-0.6000	-0.3400
15	-0.4500	-0.2500
16	-0.1800	-0.0300
17	-0.1400	-0.0800
18	-0.1000	-0.0500
19	-0.0700	-0.0300
20	-0.6200	-0.1300
21	-0.1700	-0.0700
22	-0.2400	-0.0400
23	-0.4300	-0.2700
24	-0.5900	-0.2300
25	-0.2300	-0.0900
26	-0.5900	-0.2600
27	-0.3300	-0.0900
28	-0.3100	-0.1700
29	-0.2700	-0.1100
30	-0.2000	-0.2300
31	-0.3700	-0.1000
32	-0.3700	-0.2300
33	-0.1800	-0.0700
34	-0.1600	-0.0800
35	-0.5300	-0.2200
36	-0.2800	-0.1000
37	-0.3400	0.0000
38	-0.2000	-0.1100
39	-0.8700	-0.3000
40	-0.1700	-0.0400

41	-0.1700	-0.0800
42	-0.1800	-0.0500
43	-0.2300	-0.1100
44	-1.1300	-0.3200
45	-0.6300	-0.2200
46	-0.8400	-0.1800
47	-0.1200	-0.0300
48	-0.1200	-0.0300
49	-2.7700	-1.1300
50	-0.7800	-0.0300
51	-0.7700	-0.1400
52	-0.3900	-0.1800
53	-0.2800	-0.0700
54	-0.6600	-0.2000
55	-0.6800	-0.2700
56	-0.4700	-0.1100
57	-0.6800	-0.3600
58	-0.6100	-0.2800
59	-0.7100	-0.2600
60	-0.3900	-0.3200
61	-1.3000	-0.2600
62	-0.5400	-0.2700
63	-0.2000	-0.1000
64	-0.1100	-0.0700
65	-0.2400	-0.1500
66	-0.2100	-0.1000
67	-0.4800	-0.1000
68	-0.7800	-0.4200
69	-0.6500	-0.1000
70	-0.1200	-0.0700
71	-0.3000	-0.1600
72	-0.4200	-0.3100
73	-0.3800	-0.1500
74	-0.1500	-0.0900
75	-0.3400	-0.0800
76	-0.3700	-0.1800
77	-0.2200	-0.1500
78	-0.0500	-0.0300
79	-0.2300	-0.1600
80	-0.3800	-0.2500
81	-0.3100	-0.2600
82	-0.4300	-0.1600
83	-0.2800	-0.1200
84	-0.0200	-0.0100
85	-0.0800	-0.0300
86	-0.3900	-0.3000
87	-0.2500	-0.1300
88	-0.0800	-0.0300
89	-0.2200	-0.0700
90	-0.2000	-0.0800
91	-0.3300	-0.1500
92	0.0000	-0.4000
93	0.0000	0.1400
94	0.0000	-0.2500
95	0.0000	0.1000
96	0.0000	0.1000
97	0.0000	0.1000
98	0.0000	0.1500

99	0.0000	0.1200
100	0.0000	0.2000
101	0.0000	0.2000
102	0.0000	0.1000
103	0.0000	0.2000
104	0.0000	0.0600
105	0.0000	0.0600

Line power flows for the 118 substation network.

Line No.	P (S)	P (R)	Q (S)	Q (R)
1	0.0074	-0.0075	0.0744	-0.0331
2	0.2088	-0.2104	0.2568	-0.2444
3	0.6783	-0.6806	0.8580	-0.8645
4	0.5714	-0.5829	0.2998	-0.3010
5	-0.7135	0.7027	-0.5705	0.5470
6	-0.4616	0.4605	-0.0116	0.0158
7	3.1979	-3.2265	2.8545	-0.5720
8	2.7256	-2.7256	2.4533	-2.8374
9	3.2265	-3.2508	0.5720	2.0408
10	-0.6532	0.6423	-0.2121	0.2074
11	-0.7486	0.7340	-0.3172	0.2994
12	-0.4189	0.4173	-0.2430	0.2462
13	0.2075	-0.2088	0.1231	-0.1009
14	0.0290	-0.0291	0.0445	0.0239
15	-0.2705	0.2698	0.0042	0.0077
16	-0.2574	0.2557	-0.0339	0.0601
17	-0.0648	0.0645	0.0991	-0.0690
18	0.0844	-0.0852	0.0999	0.0056
19	0.0754	-0.0759	0.0790	0.0066
20	0.0014	-0.0018	0.1394	-0.1041
21	0.8362	-0.8514	0.5981	-0.5669
22	0.2518	-0.2565	0.2041	-0.1379
23	-0.5040	0.4950	-0.6559	0.6426
24	-0.1728	0.1724	-0.0302	0.0483
25	0.0312	-0.0315	0.1146	-0.0633
26	0.0317	-0.0321	-0.1578	0.1742
27	0.2115	-0.2126	0.0933	-0.0592
28	0.3525	-0.3557	0.1392	-0.1081
29	0.4557	-0.4637	0.1581	-0.1154
30	0.1966	-0.2000	-0.4165	0.5043
31	1.0056	-1.0284	0.7868	-0.7132
32	0.2847	-0.2847	0.1310	-0.1342
33	-0.9451	0.9082	-0.4804	0.6626
34	-0.2823	0.2803	-0.1096	0.1400
35	-0.1103	0.1097	-0.0700	0.1098
36	1.4592	-1.4592	1.7596	-1.9722
37	-1.5362	1.5188	0.6164	1.2961
38	-0.7463	0.7437	0.8251	0.2631
39	-0.0523	0.0488	-0.2226	0.2838
40	0.1303	-0.1306	-0.0698	0.0834
41	-0.6685	0.6510	-0.2249	0.3885
42	0.2429	-0.2456	0.1580	-0.1220
43	-0.0813	0.0810	-0.0532	0.0878
44	-0.0936	0.0928	0.1314	-0.0774
45	-0.0015	0.0002	0.1767	-0.0673
46	0.1217	-0.1218	-0.2065	0.2108

47	0.2083	-0.2098	0.2965	-0.2786
48	0.1372	-0.1389	0.1674	-0.1055
49	-0.1347	0.1339	-0.2554	0.2635
50	0.6917	-0.6953	0.9107	-0.9051
51	1.3583	-1.3584	1.7380	-1.9301
52	-0.2058	0.2036	-0.1246	0.1674
53	-0.1085	0.1070	-0.0742	0.1475
54	-0.8033	0.8006	0.4129	0.4777
55	0.0665	-0.0667	-0.0574	0.0847
56	-0.2071	0.2064	-0.0097	0.0292
57	0.0659	-0.0662	0.0678	0.0158
58	0.1636	-0.1649	0.0708	-0.0129
59	-0.0936	0.0929	0.0935	0.0166
60	-0.2771	0.2736	0.0876	-0.0235
61	0.0671	-0.0673	-0.0366	0.0775
62	0.2828	-0.2863	-0.0039	0.0540
63	0.1987	-0.2007	0.1389	-0.0851
64	0.0457	-0.0461	0.1056	-0.0166
65	-0.0683	0.0682	-0.0400	0.0709
66	0.2942	-0.3084	0.5010	-0.1821
67	0.3145	-0.3218	0.0465	0.0180
68	0.2461	-0.2472	-0.0234	0.0447
69	-0.5718	0.5627	-0.0532	0.0634
70	-0.7140	0.6876	-0.1038	0.0929
71	-0.3076	0.3053	-0.0210	0.0390
72	-0.1253	0.1246	0.0110	0.0553
73	0.1054	-0.1058	0.0547	-0.0034
74	-0.4801	0.4630	0.0080	0.1947
75	0.0130	-0.0130	0.0036	0.0310
76	0.1843	-0.1844	0.0186	-0.0064
77	0.0616	-0.0617	0.0750	-0.0688
78	0.2619	-0.2648	0.0657	-0.0314
79	-0.3927	0.3848	-0.0233	0.0614
80	0.0885	-0.0888	0.0304	0.0105
81	-0.2101	0.2088	0.0082	0.0195
82	0.2025	-0.2051	0.1114	-0.0177
83	0.3637	-0.3702	0.1655	0.0104
84	0.2105	-0.2132	0.1169	-0.0294
85	0.4888	-0.4976	0.1494	-0.1186
86	0.5614	-0.5732	0.1484	-0.1289
87	1.0241	-1.0269	0.1339	-0.1198
88	0.2534	-0.2542	0.0147	0.0105
89	0.0119	-0.0120	-0.0232	0.0425
90	1.8875	-1.8875	0.9595	-1.1498
91	1.8875	-1.8952	1.1498	-0.8016
92	1.0394	-1.0394	0.8670	-0.9167
93	0.5577	-0.5606	1.4524	1.0697
94	2.9345	-2.9615	1.7183	-1.1929
95	2.1783	-2.2306	1.0056	-1.1691
96	0.4249	-0.4352	0.2334	-0.1587
97	0.2974	-0.3005	0.2062	-0.1574
98	-0.7547	0.7547	-0.3705	0.3476
99	-0.5886	0.5805	-0.2061	0.2274
100	1.2173	-1.2196	1.3799	0.0655
101	0.6091	-0.6411	0.1250	-0.0793
102	0.5330	-0.5620	0.1367	-0.0563
103	0.9303	-0.9303	-0.2335	0.2043
104	-0.4395	0.4258	-0.4372	0.6394

105	-0.0865	0.0857	0.1190	0.0787
106	0.3978	-0.4000	-0.2859	0.2942
107	0.0496	-0.0501	-0.0461	0.1385
108	0.1681	-0.1698	-0.0494	0.1280
109	0.2319	-0.2329	-0.2449	0.2620
110	-0.2765	0.2733	-0.0172	0.0718
111	-0.2217	0.2194	0.0964	-0.0331
112	-0.7717	0.7429	-0.3292	0.5070
113	0.1418	-0.1437	0.3741	-0.3607
114	0.3044	-0.3129	0.3295	-0.2843
115	-0.7531	0.7361	-0.1368	0.3103
116	0.0909	-0.0929	0.2025	-0.1073
117	-0.6309	0.6295	-0.0209	0.0428
118	0.0805	-0.0808	0.2172	-0.2042
119	0.8592	-0.8715	0.7450	-0.6297
120	0.4708	-0.4754	0.3242	-0.3041
121	-0.2481	0.2479	1.1746	0.7146
122	0.2479	-0.2479	0.6966	-0.7146
123	-0.2135	0.2122	0.0648	0.1030
124	0.3085	-0.3096	-0.0449	0.1205
125	0.1834	-0.1861	-0.0756	0.1237
126	0.3261	-0.3308	-0.0449	0.1017
127	0.2961	-0.2987	-0.0537	0.0737
128	-0.0254	0.0251	-0.0548	0.1112
129	0.1849	-0.1859	-0.0112	0.0945
130	0.2420	-0.2432	0.0540	-0.0014
131	0.4400	-0.4445	0.1154	-0.0471
132	0.7232	-0.7301	0.1014	-0.0951
133	-0.5763	0.5696	-0.1682	0.4806
134	-0.0006	0.0005	0.0687	-0.0244
135	-1.6739	1.6529	-0.1744	0.2811
136	-0.1894	0.1878	0.1271	-0.0639
137	-0.4482	0.4433	-0.0027	0.0319
138	-0.3924	0.3854	0.0324	0.0287
139	-0.3233	0.3210	0.0381	-0.0071
140	-0.3208	0.3194	-0.0238	0.0419
141	-0.2425	0.2395	-0.1383	0.2293
142	0.0194	-0.0194	0.0119	0.1011
143	-0.0989	0.0981	0.1682	-0.1235
144	-0.3169	0.3145	-0.1855	0.2290
145	-0.3320	0.3289	-0.1494	0.1983
146	-0.0455	0.0439	-0.1283	0.2395
147	-0.2088	0.2061	0.0455	0.1014
148	0.0133	-0.0134	-0.0059	0.1281
149	0.1006	-0.1018	0.2681	-0.2421
150	0.1636	-0.1645	0.1852	-0.1390
151	0.0111	-0.0121	-0.1183	0.2125
152	-0.2789	0.2772	-0.1136	0.1504
153	0.0713	-0.0714	0.0104	0.0551
154	-0.3451	0.3436	-0.0527	0.0767
155	0.2914	-0.2936	0.0949	-0.0467
156	-0.3140	0.3119	-0.1402	0.2398
157	-0.1672	0.1655	-0.0442	0.1430
158	-0.1147	0.1140	-0.0026	0.0790
159	-0.1452	0.1440	0.0126	0.0628
160	-0.2237	0.2204	-0.0100	0.1192
161	-0.1454	0.1452	0.0463	-0.0283
162	-0.1954	0.1949	0.0312	-0.0061

163	-0.0430	0.0429	0.0543	0.0355
164	0.0113	-0.0114	-0.0413	0.0761
165	0.0148	-0.0148	0.0470	0.0430
166	0.0314	-0.0315	-0.0661	0.0803
167	-0.0610	0.0604	-0.0512	0.1374
168	0.1115	-0.1120	-0.0503	0.0870
169	0.2238	-0.2250	0.0230	0.0108
170	-0.0182	0.0180	-0.0058	0.1216
171	0.3149	-0.3160	-0.1463	0.1571
172	-0.0390	0.0381	0.1531	-0.0595
173	-0.1184	0.1182	0.0212	0.0075
174	-0.1826	0.1819	-0.0543	0.0874
175	-0.0382	0.0381	0.0225	-0.0174
176	0.5374	-0.5378	-1.0061	1.3810
177	-0.2017	0.2000	-0.0276	0.0800
178	-0.4395	0.4359	-0.2057	0.2172
179	0.1056	-0.1059	0.0918	-0.0671

Appendix 4

The mathematical theory of the Revised Simplex method and the least squares method for solving a set of linear equations is presented in this appendix. Details of the implementations of the two methods can be found in chapter 5.

A4.1 The theory of the Revised Simplex method

The Revised Simple method as the name suggests is based on the original Simplex method of Dantzig[32] for solving a set of linear equations subject to the minimisation of an objective function which is evaluated from a set of cost coefficients and the current values of the variables. A detailed description of the Simplex method together with an introduction to the theory of linear equations can be found in reference 87. A summary of the theory relating to the Simplex method is presented in the following sub-sections.

A4.1.1 Basic feasible solutions

The idea of the Simplex method is to proceed from one basic feasible solution of a set of linear equations to another in such a way as to continually decrease the value of the objective function until a minimum is reached.

The definition of a basic feasible solution is best explained by considering a set of linear equations of the form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \qquad \text{(A4.1)}$$

where

$m \leq n$

The set of linear equations can be represented by the matrix equation

$$AX = B \qquad \text{(A4.2)}$$

where

A = m*n coefficient matrix.

X = solution vector of length n.

B = input vector of length m.

If m is less than n and the set of equations are linearly independent then there is not a unique solution. However, if an additional n-m equations of the form

$$E^k X = 0 \qquad \text{(A4.3)}$$

where

E^k = k^{th} unit vector.

(i.e. all the elements are zero except the k^{th} element which is 1.0).

are adjoined to equations A4.1 then a unique solution can be obtained, this solution is known as a basic feasible solution. If a different set of equations of the form given in equation A4.3 are adjoined to equations A4.1, in other words if a different sub-set of the variables are defined to be zero, then a different basic feasible solution is obtained.

The process of moving from one feasible solution point to another is known as pivoting. The method of pivoting is more easily visualised if the set of linear equations is written in a triangular or canonical form, thus the transformation of a set of linear equations into a canonical form is explained in the following section.

A4.1.2 The canonical representation of a set of linear equations

Given that the set of equations A4.1 are linearly independent then any equation may be replaced by a non-zero multiple of itself plus any linear combination of the other equations in the system. This enables the coefficient of one of the variables in the equation to be made equal to 1.0 and the coefficients of a further m-1 variables to be made equal to zero. The coefficients of the remaining m-n variables in the equation will have a value arising from the addition of the coefficients of that variable from the other equations. This process is known as the Gaussian reduction scheme and enables the equations to be written in the triangular or canonical form as shown below

$$
\begin{aligned}
x_1 \quad & + y_{1,m+1}x_{m+1} + y_{1,m+2}x_{m+2} + \cdots + y_{1,n}x_n = y_{1,o} \\
x_2 \quad & + y_{2,m+1}x_{m+1} + y_{2,m+2}x_{m+2} + \cdots + y_{2,n}x_n = y_{2,o} \\
& \vdots \qquad\qquad\qquad\qquad \vdots \\
x_m \quad & + y_{m,m+1}x_{m+1} + y_{m,m+2}x_{m+2} + \cdots + y_{m,n}x_n = y_{m,o}
\end{aligned}
\tag{A4.4}
$$

Corresponding to this canonical representation of the system, the variables $x_1, x_2, \ldots, x_m$ are referred to as the basic variables and the remaining variables $x_{m+1}, \ldots, x_n$ the non-basic variables. Given that the non-basic variables are defined to be zero the corresponding basic solution is then

$$x_1 = y_{1,o},\ x_2 = y_{2,o},\ \ldots,\ x_m = y_{m,o},\ x_{m+1} = 0,\ x_{m+2} = 0,\ \ldots,\ x_n = 0$$

The definition of canonical can be relaxed and a system considered to be in canonical form is amongst the n variables there exists m basic ones which have the following properties. Each basic variable appears in one equation only and its coefficient is unity and no two basic variables appear together in one equation. Given these properties a set of linear equations may be re-ordered into the form A4.4.

A4.1.3 Pivoting in the Simplex tableau

It is customary to represent the equations in A4.4 by a tableau which defines the coefficients of the matrix, A and the values of the basic solution vector, Y.

The tableau for the above system has the form

$$
\begin{array}{cccccccc}
1 & 0 & \cdots & 0 & y_{1,m+1} & y_{1,m+2} & \cdots & y_{1,n} \quad y_{1,o} \\
0 & 1 & \cdots & 0 & y_{2,m+1} & y_{2,m+2} & \cdots & y_{2,n} \quad y_{2,o} \\
\vdots & & & & & & & \\
0 & 0 & \cdots & 1 & y_{m,m+1} & y_{m,m+2} & \cdots & y_{m,n} \quad y_{m,o}
\end{array}
\tag{A4.5}
$$

The simplex method then moves from one basic solution to another by interchanging one of the basic variables for a non-basic variable, subject to the constraint that the objective function is reduced. The process of interchanging a basic variable in the above tableau for a non-basic variable

is known as pivoting, the principle of pivoting in a set of linear equations is now explained.

Denoting the colums of the coefficient matrix A as a_1, a_2, ..., a_n then the linear equations A4.1 can be written as

$$x_1a_1 + x_2a_2 + \cdots + x_na_n = b_n \tag{A4.6}$$

The vector b is therefore defined as a linear combination of the n colums vectors a_i.

Applying the Gaussian reduction scheme to obtain a tableau of the form A4.5 results in the first m column vectors forming the basis and the remaining column vectors can then be defined in terms of a linear combination of the basis vectors by simply reading the coefficients in the tableau corresponding to the column. Thus

$$a_j = y_{1,j}a_1 + y_{2,j}a_2 + \cdots + y_{m,j}a_m \tag{A4.7}$$

where
$m+1 \leq j \leq n$

The tableau can be interpreted as giving the representation of the column vectors a_j, j=1,n in terms of the basis vectors a_i, i=1,m. The j^{th} column of the tableau is the representation for the vector a_j and in particular the expression for the vector b in terms of the basis is given in the last column.

Consider the replacement of a basic column vector a_p, $1 \leq p \leq m$ by a vector a_q, $m+1 \leq q \leq n$. Provided the m vectors of the new tableau with vector a_p replaced by a_q are linearly independent then these vectors form a new basis and every other vector can be expressed as a linear combination of the new basis. This condition holds if and only if in the tableau the coefficient y_{pq} is not equal to zero. The new tableau can be formed by updating the old tableau in the following way. The vector a_q is defined in terms of the old tableau through equation A4.7 as

$$a_q = \sum_{\substack{i=1 \\ i \neq p}}^{m} (y_{i,q}a_i) + y_{p,q}a_p \tag{A4.8}$$

from which the vector a_p can be defined as

$$a_p = \frac{a_q}{y_{p,q}} - \sum_{\substack{i=1 \\ i \neq p}}^{m} \left[\frac{y_{i,q}}{y_{p,q}} a_i \right] \qquad \text{(A4.9)}$$

Writing equation A4.7 as

$$a_j = \sum_{i=1}^{m} (y_{i,j} a_i) = \sum_{\substack{i=1 \\ i \neq p}}^{m} (y_{i,j} a_i) + y_{p,j} a_p \qquad \text{(A4.10)}$$

and then substituting a_p from equation A4.9 into equation A4.10 gives

$$a_j = \sum_{\substack{i=1 \\ i \neq p}}^{m} \left[(y_{i,j} - \frac{y_{i,q} y_{p,j}}{y_{p,q}}) a_i \right] + \frac{y_{p,j}}{y_{p,q}} a_q \qquad \text{(A4.11)}$$

Thus the new coefficients of the tableau donated by $y'_{i,j}$ are obtained from equation A4.11 as

$$y'_{i,j} = y_{i,j} - \frac{y_{i,q} y_{p,j}}{y_{p,q}} \quad \text{for } i \neq p \qquad \text{(A4.12)}$$

$$y'_{p,j} = \frac{y_{p,j}}{y_{p,q}}$$

The above equations, A4.12 are known as the pivot equations and the element $y_{p,q}$ in the original system is said to be the pivot element.

The same pivot equations A4.12 may be obtained by considering the pivoting process as interchanging a basic variable with a non-basic one. This consideration involves manipulation of the rows of the tableau and is probably easier to visualise when applied to small problems solved by hand, as in the case of the examples shown below. It also enables the last column vector containing the representation of the vector b to be updated at a glance. Given the original tableau A4.5, then the first m variables are defined as basic. It is desired to interchange the basic variable p, $1 \leq p \leq m$

(corresponding with the basic column vector a_p) with the non-basic variable q, $m+1 \leq q \leq n$ (corresponding with the column vector a_q). This may only be done if the tableau coefficient $y_{p,q}$ is not equal to zero.

The transformation of the tableau requires that the coefficient of the variable x_q in the p^{th} equation be unity and all the other coefficients of x_q in the remaining equations be zero. This can be achieved by dividing all the terms of the p^{th} equation by the value of the coefficient $y_{p,q}$, thus making the new value of this coefficient equal to one. Suitable multiples of this new equation p can then be subtracted from the remaining equations to bring the coefficient of the variable x_q to zero. The definition of the new coefficients of the tableau resulting from this consideration are identical to those of equation A4.12.

The following example illustrates the procedure of pivoting. Solve the following set of linear equations

$$\begin{aligned} x_4 + x_5 + x_6 &= 5 \\ 2x_4 - 3x_5 + x_6 &= 3 \\ -x_4 + 2x_5 + x_6 &= -1 \end{aligned} \qquad \text{(A4.13)}$$

A basic solution is obtained by adjoining an additional three variables to form the equations

$$\begin{aligned} x_1 + x_4 + x_5 + x_6 &= 5 \\ x_2 + 2x_4 - 3x_5 + x_6 &= 3 \\ x_3 - x_4 + 2x_5 + x_6 &= -1 \end{aligned} \qquad \text{(A4.14)}$$

Equations A4.14 are in canonical form, the basic variables are x_1, x_2 and x_3, the non-basic variables are x_4, x_5 and x_6.
The equations A4.14 are represented by the tableau

x_1	x_2	x_3	x_4	x_5	x_6	y
1	0	0	$\underline{1}$	1	-1	5
0	1	0	2	-3	1	3
0	0	1	-1	2	-1	-1

(A4.15)

which has the basic solution

$$x_1 = 5,\ x_2 = 3,\ x_3 = -1,\ x_4 = x_5 = x_6 = 0.$$

The solution to the original set of equations A4.13 can be found by interchanging the basic variables x_1, x_2, and x_3 with the non-basic variables x_4, x_5, and x_6, (or alternatively by replacing the basic columns a_1, a_2 and a_3 by the non-basic columns a_4, a_5, and a_6).

To replace basic variable x_1 by variable x_4 the pivot element underlined in the tableau A4.15 is used.

Applying the pivot equations with p=1 and q=4 gives the tableau

x_1	x_2	x_3	x_4	x_5	x_6	y
1	0	0	1	1	-1	5
-2	1	0	0	<u>-5</u>	3	-7
1	0	1	0	3	-2	4

(A4.16)

To replace x_2 by x_5, pivot on the element underlined in tableau A4.16 to give

x_1	x_2	x_3	x_4	x_5	x_6	y
3/5	1/5	0	1	0	-2/5	18/5
2/5	-1/5	0	0	1	-3/5	7/5
-1/5	3/5	1	0	0	<u>-1/5</u>	-1/5

(A4.17)

Finally replacing x_3 by x_6 by pivoting as indicated gives

x_1	x_2	x_3	x_4	x_5	x_6	y
1	-1	-2	1	0	0	4
1	-2	-3	0	1	0	2
1	-3	-5	0	0	1	1

(A4.18)

This last canonical form has the basic solution

$$x_1 = x_2 = x_3 = 0,\ x_4 = 4,\ x_5 = 2,\ x_6 = 1.$$

Thus the solution to the original set of equations A4.13 is given by the values of the basic variables.

A4.1.4 Selection of a pivot element

The process of pivoting in a linear set of equations allows the transformation from one basic solution to another, however the random exchange of basic with non-basis variables is likely to lead to an infeasible solution.

The variables of the linear equations are usually constrained to be greater than zero, variables which are not constrained in this way are said to be free. These free variables are handled by solving for the free variable in terms of the other variables and substituting everywhere else. This is equivalent to making the free variable basic by a suitable transformation and then deleting the row and column corresponding to the free variable from the tableau. The value of the free variable is evaluated once the remaining variables have been solved. Further details on this procedure can be found in reference 87.

The basic variables must also be non-degenerate, that is $x_i>0$, i=1,m for a feasible solution, however the Simplex method can be amended if the assumption does not hold.

Thus in order to maintain a feasible solution a method of selecting which variables are to be interchanged is required. This can be achieved by selecting an arbitrary non-basic variable to enter the basis and then selecting a suitable variable to leave the basis.
Given an initial basic feasible solution

$$X = (x_1, x_2, \ldots, x_m, 0, 0, \ldots, 0)$$

equivalent to the representation

$$x_1a_1 + x_2a_2 + \ldots + x_ma_m = b \qquad \text{(A4.19)}$$

and that the column vector a_k, k>m is to be brought into the basis then the column vector a_k can be defined in terms of the current basis as

$$a_k = y_{1,k}a_1 + y_{2,k}a_2 + \ldots + y_{m,k}a_m \qquad \text{(A4.20)}$$

Multiplying equation A4.20 by a variable $e \geq 0$ and then substracting from equation A4.19 gives

$$(x_1 - ey_{1,k})a_1 + (x_2 - ey_{2,k})a_2 + \ldots + (x_m - ey_{m,k})a_m = b \qquad \text{(A4.21)}$$

Thus for any value $e \geq 0$ equation A4.21 defines the vector b as a linear combination of at most m+1 vectors. If e=0 then equation A4.21 is equivalent to the original basis, for small values of e the equation A4.21 gives a feasible but non-basic solution. As e increases from zero the coefficient of

the vector a_k increases while the coefficients of the remaining vectors will either increase or decrease linearly. If any of the coefficients decrease then the value of e is set to the value corresponding to the point when the first coefficient vanishes, that is

$$e = \text{min. wrt. } i \quad \left\{ \frac{x_i}{y_{i,k}} \; : \; y_{i,k} > 0 \right\} \tag{A4.22}$$

If the minimum is achieved by more than one index simultaneously then the new solution is degenerate and either of the columns vectors with zero coefficient can be regarded as the one which left the basis. Alternatively if all the coefficients increase as e is increased, i.e. all the $y_{i,k}$ are negative then no new feasible solution is obtained. In this case the problem is said to be unbounded, that is the variable e can be regarded as representing x_k, $k>m$ and a feasible solution to equation A4.2 is now defined in terms of m+1 variables, consequently there are now only n-m-1 variables still equal to zero. However the value of x_k or e may take any value, the larger the value the smaller the resulting value of the objective function, thus the objective function tends to minus infinity as x_k tends to infinity.

To illustrate the selection of a pivot element consider the tableau

$$\begin{array}{cccccccc}
a_1 & a_2 & \cdots & a_m & a_{m+1} & \cdots & a_n & b \\
1 & 0 & \cdots & 0 & y_{1,m+1} & & y_{1,n} & y_{1,o} \\
0 & 1 & \cdots & 0 & y_{2,m+1} & & y_{2,n} & y_{2,o} \\
 & \cdot & & & & & \cdot & \\
 & \cdot & & & & & \cdot & \\
 & \cdot & & & & & \cdot & \\
0 & 0 & \cdots & 1 & y_{m,m+1} & & y_{m,n} & y_{m,o}
\end{array} \tag{A4.23}$$

in which we assume $y_{1,o}$, $y_{2,o}$, ..., $y_{m,o}$ are all non-negative so that the basic solution $x_1=y_{1,o}$, $x_2=y_{2,o}$, ..., $x_m=y_{m,o}$ is feasible.

In order to bring a column vector a_k, $k>m$ into the basis and maintain feasibility the element i of the k^{th} column vector which will be used as the pivot element must be selected so that the ratio $x_i/y_{i,k} = y_{i,o}/y_{i,k}$ has the smallest non-negative value. This corresponds to setting the value of e in equation A4.21 to a value which satisfies the constraint of equation A4.22. The value of i defines which element is the pivot element and hence which of the basic column vectors is to leave the basis.

The following tableau numerically illustrates the selection of a pivot

element

a_1	a_2	a_3	a_4	a_5	a_6	b
1	0	0	2	4	6	4
0	1	0	1	2	3	3
0	0	1	-1	2	1	1

which has a basis a_1, a_2 and a_3 yielding a basic feasible solution

$X = (4, 3, 1, 0, 0, 0)$

The vector a_4 is to be brought into the basis. Computing the three ratios $x_i/y_{i,k} = y_{i,o}/y_{i,k}$ gives for

$i = 1,\ x_i/y_{i,k} = 4/2 = 2$
$i = 2,\ x_i/y_{i,k} = 3/1 = 3$
$i = 3,\ x_i/y_{i,k} = 1/-1 = -1$

the smallest positive value corresponds to i=1 which means vector a_1 is to leave the basis.
The new tableau is thus

a_1	a_2	a_3	a_4	a_5	a_6	b
1/2	0	0	1	2	3	2
-1/2	1	0	0	0	0	1
1/2	0	1	0	4	4	3

correspnding with the basic feasible solution

$X = (0, 1, 3, 2, 0, 0)$

The final link in the simplex process is to choose a suitable non-basic column vector (non-basic variable) to enter the basis so that the new basic feasible solution yields a lower value for the objective function than the previous one. Thus by selecting which column vector is to enter the basis and then selecting a suitable element of this vector upon which to pivot the entire process proceeds from one basic feasible solution to another, successively reducing the value of the objective function. The method for selecting which column vector is to enter the basis is outlined below.

Given a basic feasible solution

$$(X_b, 0) = (y_{1,o}, y_{2,o}, \ldots, y_{m,o}, 0, 0, \ldots, 0)$$

where
the subscript b refers to the basic variables

corresponding to the tableau A4.23 and also the value of the objective function at any feasible solution is

$$\Phi = c_1x_1 + c_2x_2 + \ldots + c_nx_n \qquad \text{(A4.24)}$$

where
C = vector of cost coefficients.

then for the above basic solution the corresponding value of the objective function is

$$\Phi_o = C_bX_b \qquad \text{(A4.25)}$$

where
$C_b = (c_1, c_2, \ldots, c_m)$

The basic solution above is obtained by assigning a value of zero to the non-basic variables, however the value of the objective function is true for any solution. Hence if the non-basic variables are assigned arbitrary values the remaining variables can be defined as below

$$x_1 = y_{1,o} - \sum_{j=m+1}^{n} (y_{1,j}x_j) \qquad \text{(A4.26)}$$

$$x_2 = y_{2,o} - \sum_{j=m+1}^{n} (y_{2,j}x_j)$$

.
.
.

$$x_m = y_{m,o} - \sum_{j=m+1}^{n} (y_{m,j} x_j)$$

Using the above equations A4.26 to eliminate the variables $x_1, x_2, \ldots, x_m$ from the objective function A4.24 gives

$$\begin{aligned}\Phi = CX = c_1 \; & y_{1,o} - \sum_{j=m+1}^{n} (y_{1,j} x_j) \\ + c_2 \; & y_{2,o} - \sum_{j=m+1}^{n} (y_{2,j} x_j) \\ + \; & \ldots + \\ + c_m \; & y_{m,o} - \sum_{j=m+1}^{n} (y_{m,j} x_j) \\ + c_{m+1} & x_{m+1} + c_{m+2} x_{m+2} + \ldots + c_n x_n\end{aligned} \tag{A4.27}$$

The value of the objective function for a basic feasible solution is defined by equation A4.25. At the basic solution the values of $x_1, x_2, \ldots, x_m = y_{1,o}, y_{2,o}, \ldots, y_{m,o}$ hence in equation A4.27 the terms involving $y_{1,o}, y_{2,o}, \ldots, y_{m,o}$ can be replaced by Φ_o. Collecting the terms in $x_{m+1}, x_{m+2}, \ldots, x_n$ results in

$$\begin{aligned}\Phi = CX = \Phi_o \; & + (c_{m+1} - \Phi_{m+1}) x_{m+1} + (c_{m+2} - \Phi_{m+2}) x_{m+2} \\ & + \ldots \\ & + (c_n - \Phi_n) x_n\end{aligned} \tag{A4.28}$$

where

$$\Phi_j = y_{1,j} c_1 + y_{2,j} c_2 + \ldots + y_{m,j} c_m \tag{A4.29}$$

$m+1 \leq j \leq n$

The equation A4.28 is used to select the non-basic variable and hence the column vector to enter the basis by using the following reasoning. Starting from the initial basic feasible solution means that the values of the variables x_{m+1}, x_{m+2}, ..., x_n are zero. If the value of one of the coefficients $(c_j - \Phi_j)$, $m+1 \leqq j \leqq n$ is negative, then by increasing the value of the corresponding variable x_j in equation A4.28 the overall value of the objective function will be reduced. Thus the non-basic variable with the most negative value of the coefficient $(c_j - \Phi_j)$ is likely to give the greatest reduction in the objective function when the variable is made basic and is hence the variable selected to enter the basis.

Note the equations A4.28 and A4.29 take into account the change in the value of the objective function arising from the change in the values of the basic variables x_1, x_2, ..., x_m to accommodate the change in the value of x_j. The coefficients in equation A4.28, $(c_j - \Phi_j)$, $m+1 \leqq j \leqq n$ are known as the relative cost coefficients or alternatively the reduced cost coefficients and are represented by r_j.

The solution is said to be optimal if for a basic feasible solution all the values of r_j, $m+1 \leqq j \leqq n$ are greater than or equal to zero.

A4.1.5 Computational procedure for implementing the Simplex method

The computational procedure for solving the system

Minimise CX

Subject to $AX = B$

$X \geqq 0$

uses the techniques outlined above to move from one basic feasible solution to another whilst reducing the value of the objective function.

Given an initial basic feasible solution and a tableau that is in canonical form, then the initial tableau for the simplex method is found by adding a further row to the bottom containing the relative cost coefficients, r_j for all the variables and at the right hand end the negative value of the present cost. The reader is referred to reference 87 for further details how to transform a problem into a canonical form if it is not immediately obvious.

Note the values of the relative cost coefficients are zero for the basic variables, however it may be easier to form the last row by entering the values of the objective function, c_i, i=1,n and the value of zero for the present cost and then subtracting suitable multiples of the rows 1 to m from

the last row to reduce the elements for the basic columns to zero. In doing so the value of the present cost is evaluated simultaneously.
The initial simplex tableau is thus of the form below

$$
\begin{array}{cccccccc}
a_1 & a_2 & \cdots & a_m & a_{m+1} & \cdots & a_n & b \\
1 & 0 & \cdots & 0 & y_{1,m+1} & & y_{1,n} & y_{1,o} \\
0 & 1 & \cdots & 0 & y_{2,m+1} & & y_{2,n} & y_{2,o} \\
 & \cdot & & & & & \cdot & \\
 & \cdot & & & & & \cdot & \\
 & \cdot & & & & & \cdot & \\
0 & 0 & \cdots & 1 & y_{m,m+1} & & y_{m,n} & y_{m,o} \\
0 & 0 & \cdots & 0 & r_{m+1} & & r_n & -\Phi_o
\end{array}
$$

The Simplex process then proceeds by repeatedly selecting the column vector to enter the basis then the column to leave the basis as described above until either an optimal solution is reached or the problem is found to be unbounded. Note when pivoting the last row of the tablea is treated in the same way as the others (except the row containing the pivot element) and hence the values of the relative cost coefficients and the negative value of the present cost are continually updated.

A4.1.6 The Revised Simplex method

It has been convenient to explain the principle of the Simplex method with attention focussed on the individual elements of the tableau, however by studying the vector-matrix relationship that exists a saving in computational and storage requirements can be made. It is this reduced version of the method which is commonly referred to as the Revised Simplex Method.

Letting the submatrices A_b and A_d of the coefficient matrix A represent the variables which are basic an and non-basic resptively, then the matrix A is thus partitional as $A = (A_b, A_d)$. Similarly partitioning the vectors representing the variables and the cost coefficients as $X = (X_b, X_d)$ and $C = (C_b, C_d)$ results in the standard linear programming problem becoming

$$
\begin{array}{ll}
\text{Minimise} & C_bX_b + C_dX_d \\
\text{Subject to} & A_bX_b + A_dX_d = B \\
 & X_b \geq 0,\ X_d \geq 0
\end{array}
\qquad \text{(A4.30)}
$$

The basic solution, which is assumed to be feasible, corresponding to the

basis A_b is $X = (X_b, 0)$ where $X_b = A_b^{-1}$. It should be noted that at this solution point $X_d = 0$, however for any value of X_d the corresponding values of X_b can be found by re-arranging equation A4.30 to give

$$X_b = A_b^{-1}B - A_b^{-1}A_dX_d$$

Substituting this general expression into the objective function gives

$$\Phi = C_b(A_b^{-1}B - A_b^{-1}A_dX_d) + C_dX_d$$

$$= C_bA_b^{-1}B + (C_d - C_bA_b^{-1}A_d)X_d$$

The relative cost coefficient vector R for the non-basic variables is given by $C_d - C_bA_b^{-1}A_d$.
The initial Simplex tableau can be written as

$$\left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right] = \left[\begin{array}{c|c|c} A_b & A_d & B \\ \hline C_b & C_d & 0 \end{array}\right] \qquad \text{(A4.31)}$$

However, this matrix will not in general be in canonical form and does not correspond to a point in the Simplex procedure.

If the matrix A_b is used as a basis then the columns must be linearly independent, hence by multiplying the top of A4.31 by A_b^{-1} will form the required identity matrix in the top left hand corner. The cost coefficients are converted into the relative cost coefficients by subtracting C_b times the canonical rows 1 to m from the bottom row to reduce the values for the basic variables to zero. Applying these transformations to A4.31 results in a tableau of the form corresponding to the point in the Simplex procedure where a new non-basic variable is chosen to enter the basis. The initial tableau for the Simplex process is thus

$$T = \left[\begin{array}{c|c|c} I & A_b^{-1}A_d & A_b^{-1}B \\ \hline 0 & C_d - C_bA_b^{-1}A_d & -C_bA_b^{-1}B \end{array}\right]$$

Experimental work has shown that the Simplex process is expected to converge to an optimum solution in m to 3m/2 pivot operations.[87] If the matrix A has fewer rows than columns, i.e. m is less than n then pivots will occur in only a few of the columns and the time taken updating the tableau for the remaining

columns is to some extent wasted. The Revised Simplex method is a scheme to avoid this unnecessary computation.

Given the inverse of a current basis, A_b^{-1} and the current solution $X_b = Y_o = A_b^{-1}B$ the following steps are executed.

1. Calculate the relative cost coefficients $R = C_d - C_bA_b^{-1}A_d$. (This can be done in two steps, evaluate $S = C_bA_b^{-1}$, then $C_d - SA_d$). If for all values of j, $r_j \geq 0$ where $m+1 \leq j \leq n$ then the current solution is optimal.
2. Select the vector a_j to enter the basis and then evaluate the vector $y_j = A_b^{-1}a_j$ which defines the vector a_j in terms of the current basis.
3. Calculate the ratios $y_{i,o}/y_{i,j}$ to determine which column vector is to leave the basis.
4. Update A_b^{-1} and the current solution $A_b^{-1}B$ then return to step 1.

The updating of A_b^{-1} and $A_b^{-1}B$ is accomplished by the usual pivot operations applied to an array consisting of A_b^{-1}, $A_b^{-1}B$ and y_j where the pivot element is the appropriate element in y_j. It should be noted that in evaluating the expressions in step 1 the elements of the vectors C_b and C_d and the matrix A_d are obtained from the initial Simplex tableau by updating a list of which variables are currently in the basis. The elements of the inverse of the basis, A_b^{-1} are obtained from the Revised Simplex tableau which is updated by every pivot operation as indicated by step 4.

A4.1.7 An example of the Revised Simplex Method

The procedure of the Revised Simplex method is illustrated by the following example

$$\begin{aligned}
\text{Maximise} \quad & 3x_1 + x_2 + 3x_3 \\
\text{Subject to} \quad & 2x_1 + x_2 + x_3 \leq 2 \\
& x_1 + 2x_2 + 3x_3 \leq 5 \\
& 2x_1 + 2x_2 + x_3 \leq 6 \\
& x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0
\end{aligned}$$

To transform the problem into a standard form suitable for the Simplex method the following transformations are applied. The maximisation is turned into a minimisation by multiplying by -1, the inequalities are treated as equalities, as the maximum value of the object function will arise when the variables have the maximum possible values, thus the problem is equivalent to

Maximise $-3x_1 - x_2 - 3x_3$

Subject to
$$2x_1 + x_2 + x_3 = 2$$
$$x_1 + 2x_2 + 3x_3 = 5$$
$$2x_1 + 2x_2 + x_3 = 6$$
$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$$

Finally three additional non-negative variables x_4, x_5 and x_6 are introduced to form the initial basis. The cost coefficients of these additional variables are zero, hence the objective function is determined by the vector C = (-3, -1, -3, 0, 0, 0)
The initial Simplex tableau is thus

a_1	a_2	a_3	a_4	a_5	a_6	b
2	1	1	1	0	0	2
1	2	3	0	1	0	5
2	2	1	0	0	1	6
-3	-1	-3	0	0	0	0

The inverse of the initial basis A_b^{-1} is also the identify matrix, thus the initial tableau for the Revised Simplex method is

basic variable	A_b^{-1}			X_b
4	1	0	0	2
5	0	1	0	5
6	0	0	1	6

Compute $S = C_b A_b^{-1} = (0, 0, 0)A_b^{-1} = (0, 0, 0)$
hence $R = C_d - SA_d = (-3, -1, -3)$
which is equivalent to writing $c_1 - \Phi_1 = -3$, $c_2 - \Phi_2 = -1$, $c_3 - \Phi_3 = -3$.
Bring a_2 into the basis (selected to simplify hand calculations). The representation of a_2 in terms of the current basis is $y_2 = A_b^{-1}a_2$.
Thus the tableau takes the form

basic variable	A_b^{-1}			X_b	y_2
4	1	0	0	2	$\underline{1}$
5	0	1	0	5	2
6	0	0	1	6	2

Obtaining the smallest ratio $y_{i,o}/y_{i,j}$, (evaluated by dividing the elements of vector X_b by those of y_2) gives the pivot element as being in row one of the above tableau as indicated by the under-lined element.
The updated tableau becomes

basic variable		A_b^{-1}		X_b
2	1	0	0	2
5	-2	1	0	1
6	-2	0	1	2

then $S = C_b A_b^{-1} = (-1, 0, 0)A_b^{-1} = (-1, 0, 0)$
thus $SA_d = (-2, -1, -1)$
and $R = C_d - SA_d = (-1, -2, 1)$
or alternatively $c_1 - \Phi_1 = -1$, $c_3 - \Phi_3 = -2$, $c_4 - \Phi_4 = 1$.
Selecting the column a_3 ($r_3 = -2$) to enter the basis gives the tableau

basic variable		A_b^{-1}		X_b	y_3
2	1	0	0	2	1
5	-2	1	0	1	$\underline{1}$
6	-2	0	1	2	-1

Pivoting as indicated gives

basic variable		A_b^{-1}		X_b
2	3	-1	0	1
3	-2	1	0	1
6	-4	1	1	3

now $S = C_b A_b^{-1} = (-1, -3, 0)A_b^{-1} = (3, -2, 0)$
thus $R = C_d - SA_d = (-7, -3, 2)$
or alternatively $c_1 - \Phi_1 = -7$, $c_4 - \Phi_4 = -3$, $c_5 - \Phi_5 = 2$.
bringing the column a_1 into the basis gives the tableau

basic variable		A_b^{-1}		X_b	y_1
2	3	-1	0	1	$\underline{5}$
3	-2	1	0	1	-3
6	-4	1	1	3	-5

using only the possible pivot element indicated results in

basic variable		A_b^{-1}		X_b
1	3/5	-1/5	0	1/5
3	-1/5	2/5	0	8/5
6	-1	0	1	4

Finally $S = C_b A_b^{-1} = (-3, -3, 0)A_b^{-1} = (-6/5, -3/5, 0)$
and $R = C_d - SA_d = (7/5, 6/5, 3/5)$
thus the solution obtained is optimal and the values of the variables are $x_1 = 1/5$, $x_2 = 0$, $x_3 = 8/5$, $x_4 = 0$, $x_5 = 0$, $x_6 = 4$.

A4.2 The Least Squares Method

As the names suggests, the least squares method of solving a set of linear equations minimises the sum of the squares of the error terms or residuals for each equation. The algorithm used is based on the original Conjugate Gradient method proposed by Hestenes and Stiefel.[59] A comparison of the various versions of the Conjugate Gradient method has been made by Reid,[105] and a comparison of the alternative methods for solving sparse linear least squares problems written by Bjorckhas been pulished in reference 21. The latter reference details the equations used in the implementation of the Conjugate Gradient Method.

A4.2.1 The Conjugate Gradient Method

The Conjugate Gradient method for solving a linear least squares problem is an iterative process which uses the following equations to move from the initial solution point to the final solution point.
Write the linear problem in the usual way as

$$AX = B$$

where

A = m*n coefficient matrix.

B = input vector of length m.

X = solution vector of length n.

The method is initialised by computing the values of the following vectors

$$R_0 = B - AX_0 \tag{A4.32}$$

$$P_0 = S_0 = A^t R_0 \tag{A4.33}$$

where

R = vector of residuals (errors) of length n.

P/S = work space vectors of length n.

X_0 = initial solution vector.

The following vector equations are then evaluated in an iterative process until the desired convergence is obtained

$$Q_i = AP_i \tag{A4.34}$$

$$a_i = \|S_i\|^2 / \|Q_i\|^2 \tag{A4.35}$$

$$X_{i+1} = X_i + a_i P_i \tag{A4.36}$$

$$R_{i+1} = R_i - a_i Q_i \tag{A4.37}$$

$$S_{i+1} = A^t R_{i+1} \tag{A4.38}$$

$$c_i = \|S_{i+1}\|^2 / \|S_i\|^2 \tag{A4.39}$$

$$P_{i+1} = S_{i+1} + c_i P_i \tag{A4.40}$$

where

Q = work space vector of length m.

a = temporary scalar.

c = temporary scalar.

$\|\cdot\|$ = Euclidian norm of the vector, which for a given vector, Z of length j is defined as

$$\sqrt{\left[\sum_{i=1}^{j} (z_i)^2\right]}$$

The test for convergence is usually based on the magnitude of the elements of the vectors P and R. If any of these elements are above a desired tolerance,

the process is deemed not to have converged.

A4.2.2 Normalised Equations

The vector R which has a length equal to the number of equations stores the residuals of the equations. This residual can be considered as representing the difference between the value of the equation when evaluated using the present values of the estimates and the required value of the equation stored in the input vector B. The algorithm minimuses the sum of the squares of the normalised residuals. The residuals are normalised so that equations with large terms of the coefficient matrix A do not bias the solution point towards the constraint associated with that equation. The process of normalising an equation is explained as follows.

Consider a set of m equations which contains the following two equations

$$x_1 + x_2 = 1.0 \qquad \text{(A4.41)}$$

$$10x_1 + 10x_2 = 10.0 \qquad \text{(A4.42)}$$

These two equations define the values of x_1 and x_2 in exactly the same way. However, should the present solution point as defined by the entire set of equations set the values of x_1 and x_2 at 0.4 and 0.5 respectively then the value of the square of the residual for equation A4.41 is given by

$$(1.0 - (0.4 + 0.5))^2 = 0.01$$

The corresponding value of the square of the residual for equation A4.42 is given by

$$(10.0 - (10.0*0.4 + 10.0*0.5))^2 = 1.0$$

Thus by multiplying the left and right hand sides of the equation by a scalar increases the value of square of the residual in proportion to the square of the scalar. The solution point would therefore be unjustifiably moved in a direction so as to reduce the magnitude of the residuals arising from equations with large coefficients. This problem is overcome by normalising the equation which involves dividing the left and right hand sides of each equation by a normalising scalar. The value of the scalar is evaluated as shown below.

$$n_i = \sqrt{\left[\sum_{j=1}^{l} (a_{i,j})^2 \right]} \qquad \text{(A4.43)}$$

where

n_i = normalising scalar for equation.

l = number of terms in equation i.

$a_{i,j}$ = elements of the coefficient matrix A.

The normalising scalar for equation A4.41 is therefore given by

$$\sqrt{(1.0^2 + 1.0^2)} = \sqrt{2.0}$$

Thus equation A4.41 is then re-written as

$$\frac{x_1}{\sqrt{2}} + \frac{x_2}{\sqrt{2}} = \frac{1.0}{\sqrt{2}} \qquad \text{(A4.44)}$$

Similarly equation A4.42 is re-written as

$$\frac{10x_1}{\sqrt{200.0}} + \frac{10x_2}{\sqrt{200.0}} = \frac{10.0}{\sqrt{200.0}} \qquad \text{(A4.45)}$$

Substituting the same estimates for x_1 and x_2 as before, namely 0.4 and 0.5 respectively, yields the following value for the square of residual from equation A4.44.

$$(1.0/\sqrt{2.0} - (0.4/\sqrt{2.0} + 0.5/\sqrt{2.0}))^2 = 0.0701$$

Equation A4.45 also yields the same value for the square of the residual. It should be noted however that this process of normalising the equation is not explicitly performed on entry to the subroutine. The evaluation of the equations A4.32 to A4.40 effectively calculates and updates the solution vector, X and the residual vector, R without the need to normalise the equations or to evaluate the inverse of the coefficient matrix, A.

4.2.3 Derivation of the name Conjugate Gradients

The algorithm derives its name from the fact that the successive values of the vector P are conjugate gradients. The term conjugate gradient means that for the vectors P_{i+1} and P_i the following definition is true

$$P_{i+1}{}^{t}A^{t}AP_i = 0 \tag{A4.46}$$

where

i = iteration number.

If the least squares problem is thought of as a graphical problem of n dimensions where n is the number of variables in the problem then the vector P can be considered as representing the direction (but not the magnitude) to move in to proceed from one estimate of the solution point to the next. This process is illustrated in the example presented in the next section.

The theory of the conjugate gradient method states that the optimum solution point will be obtained from any starting point in a maximum number of n iterations. However, computational round-off in the computer degrades the performance of the algorithm and increases the number of iterations required to reach the solution point.

4.2.4 An example of a least squares solution

Consider the following three equations written in terms of the independent variables x_1 and x_2.

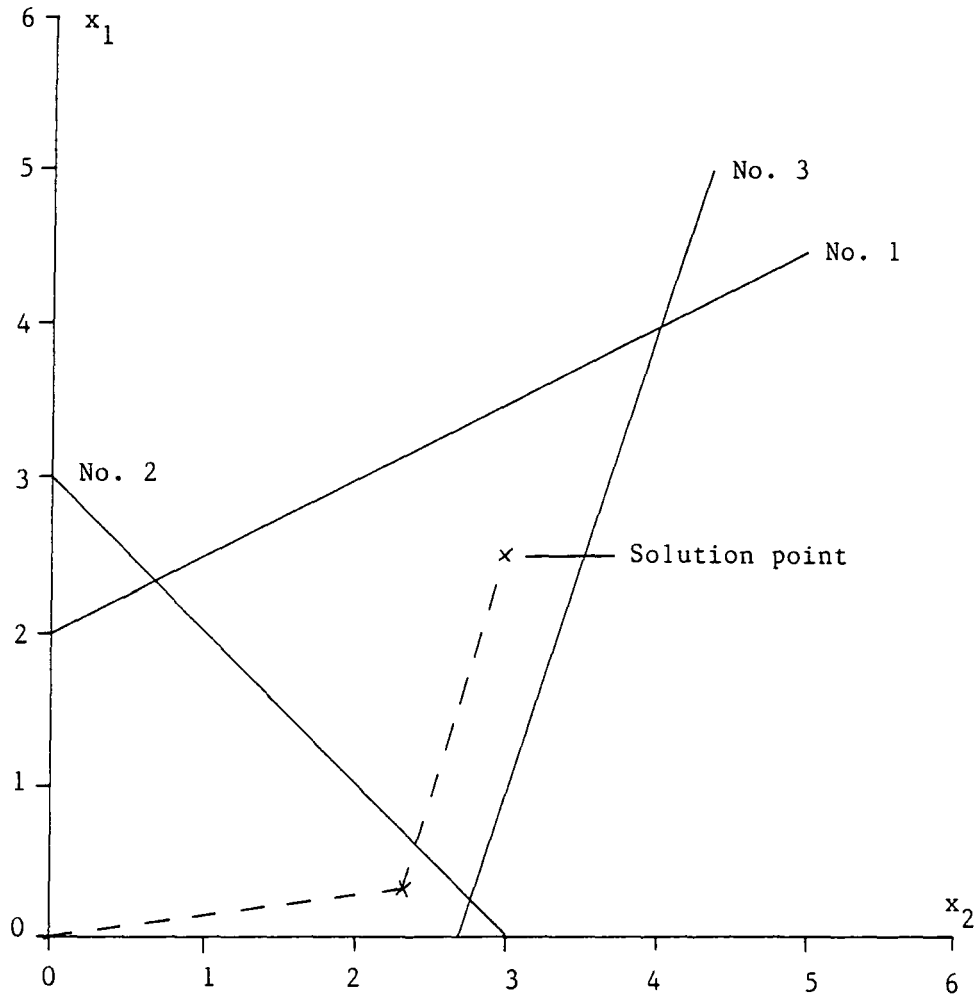
$$2x_1 - x_2 = 4 \tag{A4.47}$$

$$x_1 + x_2 = 3 \tag{A4.48}$$

$$x_1 - 3x_2 = -8 \tag{A4.49}$$

These equations may be represented graphically in two dimensions as shown in figure A4.1. The calculation of successive solution points using equations A4.32 to A4.45 is illustrated both numerically as shown below and graphically in figure A4.1. It should be noted that for convenience, all vectors will be written horizontally and the true orientation must be inferred from the above mentioned equations.

The initial values for both variables has been set to zero, thus using the



The equation for line

No. 1 is $2x_1 - x_2 = 4$

No. 2 is $x_1 + x_2 = 3$

No. 3 is $x_1 - 3x_2 = -8$

Fig. A4.1: An example of a least squares solution

equation A4.32 gives the following initial values for the vector R which stores the values of the residuals.

R_0 = (4.0000, 3.0000, -8.0000)

Equation A4.33 gives the initial values for the work space vectors P and S as

$P_0 = S_0$ = (3.0000, 23.0000)

The initial value of the work space vector Q as given by equation A4.34 is

Q_0 = (-17.0000, 26.0000, -66.0000)

Computing the squares of the Euclidian norm of the vectors S and Q and substituting into equation A4.35 gives the initial value of the variable a as

a_0 = 538.0000/5321.0000 = 0.1011

The vector P_0 represents the direction of the first update to the solution point from the initial solution point. The magnitude and direction are obtained by multiplying the vector P_0 by the variable a_0, thus the new solution point as given by equation A4.36 is

X_1 = (0.3033, 2.3255)

The vector representing the magnitude and direction of the new solution point in terms of the initial solution point is illustrated in figure A4.1 along with vector representing the second update.

Equation A4.37 may be used to calculate the new values of the residuals and gives the following result

R_1 = (5.7188, 0.3712)

Equations A4.38 to A4.40 result in the values of the work space vectors P and S becoming

S_1 = (10.4821. -1.3672)

P_1 = (11.1052, 3.4099)

Returning to the top of the iterative loop, i.e. equation A4.34 gives the following values for the vector Q and variable a

Q_1 = (18.8004, 14.5150, 0.8755)

a_1 = 0.1978

Thus the second estimate of the solution point given by equation A4.36 is

X_2 = (2.4999, 3.0000)

which yields the corresponding residuals

R_2 = (2.0000, -2.4999, -1.4999)

The process has effectively converged at this point although the process will not terminate until the vector P is re-calculated.

It is important to note that the numerical accuracy of the machine is such that the values of the vector P_2 are of the order 5.0 E^{-6} and that the tolerance used to determine when convergence has been achieved must be greater than this value. To illustrate the problem of computer round-off, the

tolerance was set at $1.0\ E^{-15}$. The subroutine returned on the fifth iteration with

$X_4 = (2.5000, 3.0000)$

and

$R_4 = (2.0000, -2.5000, -1.5000)$

Appendix 5

Appendix 5 lists the noisy measurements used to obtain the results presented in chapter 6. The measurements have been subject to a 0.2% systematic error and a 0.5% random error. A comparison is made between the measurement value and the unperturbed value.

Values are in P.U.
(S) => sending end of a line, (R) => receiving end of line.
Difference = Measurement value - True value

Voltage magnitude measurements

Meas. No.	Bus No.	True value	Meas. value	Difference
1	1	1.0438	1.0514	0.0076
2	2	1.0301	1.0237	-0.0065
3	3	1.0301	1.0215	-0.0087
4	5	1.0025	0.9783	-0.0243
5	7	0.9851	0.9688	-0.0163
6	8	0.9851	0.9979	0.0129
7	9	0.9851	1.0012	0.0162
8	10	0.9951	0.9931	-0.0020
9	11	0.9951	1.0169	0.0219
10	13	0.9951	0.9777	-0.0174
11	12	0.9951	1.0104	0.0154
12	14	0.9951	1.0015	0.0065
13	15	0.9951	1.0011	0.0060
14	22	0.9894	0.9716	-0.0178
15	17	0.9894	0.9894	0.0000
16	21	0.9894	0.9694	-0.0200
17	19	0.9894	0.9675	-0.0219
18	23	0.9894	0.9733	-0.0161
19	36	1.0486	1.0430	-0.0056
20	39	1.0011	0.9945	-0.0066
21	53	0.9637	0.9522	-0.0115
22	54	0.9637	0.9760	0.0124
23	61	0.9673	0.9716	0.0043

Gen. active power flow measurements

Meas. No.	Gen. No.	True value	Meas. value	Difference
1	1	1.1469	1.1349	-0.0120
2	2	0.7502	0.7670	0.0168
3	3	0.4009	0.3934	-0.0075
4	4	0.2003	0.1958	-0.0045
5	5	0.2001	0.2048	0.0046
6	6	0.2002	0.1995	-0.0007

Gen. reactive power flow measurements

Meas. No.	Gen. No.	True value	Meas. value	Difference
1	1	0.0689	0.0675	-0.0014
2	2	0.3254	0.3221	-0.0033
3	3	0.1129	0.1109	-0.0020
4	4	0.1672	0.1675	0.0004
5	5	0.2133	0.2156	0.0023
6	6	0.0897	0.0890	-0.0007

Load active power flow measurements

Meas. No.	Load No.	True value	Meas. value	Difference
1	1	-0.2170	-0.2102	0.0068
2	2	-0.0240	-0.0237	0.0003
3	3	-0.0760	-0.0762	-0.0002
4	4	-0.3790	-0.3763	0.0027
5	5	-0.3290	-0.3371	-0.0081
6	6	-0.2340	-0.2384	-0.0044
7	7	-0.2280	-0.2293	-0.0013
8	8	-0.3000	-0.3015	-0.0015
9	9	-0.0580	-0.0572	0.0008
10	10	-0.1120	-0.1122	-0.0002
11	11	-0.0620	-0.0629	-0.0009
12	12	-0.0820	-0.0822	-0.0002
13	13	-0.0350	-0.0352	-0.0002
14	14	-0.0900	-0.0889	0.0011
15	15	-0.0320	-0.0312	0.0008
16	16	-0.0950	-0.0974	-0.0024
17	17	-0.0220	-0.0221	-0.0001
18	18	-0.1750	-0.1737	0.0013
19	19	-0.0320	-0.0313	0.0007
20	20	-0.0870	-0.0876	-0.0006
21	22	-0.0240	-0.0231	0.0009
22	23	-0.1060	-0.1051	0.0009

Load reactive power flow measurements

Meas. No.	Load No.	True value	Meas. value	Difference
1	1	-0.1270	-0.1287	-0.0017
2	2	-0.0120	-0.0120	0.0000
3	3	-0.0160	-0.0154	0.0006
4	4	-0.0760	-0.0756	0.0004
5	5	-0.0660	-0.0668	-0.0008
6	6	-0.0480	-0.0481	-0.0001
7	7	-0.1090	-0.1107	-0.0017
8	8	-0.3000	-0.2990	0.0010
9	9	-0.0200	-0.0201	-0.0001
10	10	-0.0750	-0.0754	-0.0004
11	11	-0.0160	-0.0159	0.0001
12	12	-0.0250	-0.0250	0.0000
13	13	-0.0180	-0.0182	-0.0002
14	14	-0.0580	-0.0568	0.0012
15	15	-0.0090	-0.0089	0.0001
16	16	-0.0340	-0.0332	0.0008
17	17	-0.0070	-0.0069	0.0001
18	18	-0.1120	-0.1146	-0.0026
19	19	-0.0160	-0.0160	0.0000
20	20	-0.0670	-0.0664	0.0006
21	22	-0.0090	-0.0089	0.0001
22	23	-0.0190	-0.0192	-0.0002

Link active power flow measurements

Meas. No.	Link No.	True value	Meas. value	Difference
1	1	-0.4702	-0.4802	-0.0100
2	2	-0.4702	-0.4597	0.0105
3	3	0.0904	0.0887	-0.0016
4	4	-0.1277	-0.1268	0.0009
5	5	-0.1277	-0.1274	0.0002
6	6	-0.2643	-0.2675	-0.0032
7	7	0.1366	0.1356	-0.0010
8	8	0.0793	0.0776	-0.0017
9	9	0.0753	0.0740	-0.0012
10	10	-0.0483	-0.0478	0.0005
11	11	-0.2237	-0.2247	-0.0010
12	12	-0.1276	-0.1302	-0.0026
13	13	-0.0040	-0.0039	0.0001
14	14	0.0720	0.0733	0.0014
15	15	0.1236	0.1225	-0.0011
16	16	-0.1754	-0.1717	0.0037
17	17	0.0760	0.0763	0.0003
18	30	-0.0778	-0.0803	-0.0026
19	31	-0.0321	-0.0319	0.0002
20	32	0.1378	0.1353	-0.0025
21	33	-0.0285	-0.0288	-0.0003
22	34	-0.1101	-0.1087	0.0014
23	35	0.0590	0.0602	0.0012
24	36	0.0729	0.0727	-0.0002
25	37	0.0213	0.0215	0.0002
26	44	-0.0082	-0.0078	0.0004
27	45	-0.0649	-0.0653	-0.0005
28	46	-0.0294	-0.0296	-0.0002
29	47	-0.0437	-0.0439	-0.0003
30	48	-0.0448	-0.0447	0.0001
31	49	-0.0448	-0.0453	-0.0006
32	51	0.0204	0.0203	-0.0001
33	54	0.0838	0.0846	0.0007
34	59	0.0000	0.0002	0.0002
35	58	0.0120	0.0116	-0.0004
36	60	-0.0185	-0.0184	0.0001
37	61	0.0305	0.0308	0.0003
38	64	-0.0346	-0.0355	-0.0008
39	65	0.0530	0.0516	-0.0014
40	68	-0.0184	-0.0182	0.0001
41	64	-0.0346	-0.0351	-0.0005
42	65	0.0530	0.0531	0.0001
43	68	-0.0184	-0.0180	0.0003

Link reactive power flow measurements

Meas. No.	Link No.	True value	Meas. value	Difference
1	1	-0.1191	-0.1212	-0.0021
2	2	-0.1191	-0.1202	-0.0011
3	3	-0.0040	-0.0040	0.0000
4	4	-0.0335	-0.0333	0.0003
5	5	-0.0335	-0.0332	0.0004
6	6	-0.0732	-0.0747	-0.0015
7	7	0.0397	0.0379	-0.0018
8	8	0.0018	0.0018	0.0000
9	9	0.0320	0.0318	-0.0002
10	10	-0.0056	-0.0056	0.0000
11	11	-0.0362	-0.0360	0.0002
12	12	-0.0073	-0.0074	-0.0001
13	13	0.0302	0.0309	0.0007
14	14	0.0343	0.0343	0.0000
15	15	0.0376	0.0384	0.0009
16	16	-0.0306	-0.0304	0.0002
17	17	0.0040	0.0040	-0.0001
18	30	-0.0832	-0.0852	-0.0020
19	31	-0.0997	-0.1017	-0.0020
20	32	0.1132	0.1141	0.0009
21	33	0.0068	0.0067	-0.0001
22	34	-0.0433	-0.0429	0.0004
23	35	0.0564	0.0567	0.0004
24	36	0.0530	0.0527	-0.0003
25	37	0.0031	0.0030	-0.0001
26	44	-0.0060	-0.0060	0.0001
27	45	-0.0038	-0.0039	0.0000
28	46	-0.0054	-0.0051	0.0003
29	47	-0.0045	-0.0044	0.0000
30	48	-0.0163	-0.0168	-0.0005
31	49	-0.0163	-0.0165	-0.0002
32	51	0.0054	0.0055	0.0001
33	54	0.0270	0.0278	0.0008
34	59	0.0000	0.0000	0.0000
35	58	0.0045	0.0045	0.0000
36	60	-0.0031	-0.0033	-0.0003
37	61	0.0076	0.0076	0.0000
38	64	-0.0068	-0.0068	-0.0001
39	65	0.0095	0.0095	0.0000
40	68	-0.0027	-0.0027	0.0000
41	64	-0.0068	-0.0069	-0.0001
42	65	0.0095	0.0093	-0.0002
43	68	-0.0027	-0.0026	0.0001

Line active power flow measurements

Meas. No.	Line No.	True value	Meas. value	Difference
1	1 (S)	-0.7092	-0.7198	-0.0106
2	2 (S)	-0.4378	-0.4432	-0.0054
3	1 (R)	0.7003	0.6981	-0.0022
4	3 (S)	-0.3164	-0.3146	0.0018
5	6 (S)	-0.4071	-0.3938	0.0133
6	5 (S)	-0.5101	-0.5186	-0.0085
7	2 (R)	0.4294	0.4399	0.0105
8	4 (S)	-0.4054	-0.4076	-0.0022
9	3 (R)	0.3107	0.3098	-0.0009
10	4 (R)	0.4032	0.4104	0.0071
11	15 (S)	-0.2369	-0.2389	-0.0020
12	7 (S)	-0.4011	-0.3965	0.0046
13	5 (R)	0.4977	0.4892	-0.0085
14	8 (S)	0.0434	0.0426	-0.0008
15	6 (R)	0.3977	0.3941	-0.0035
16	7 (R)	0.3991	0.3922	-0.0069
17	11 (S)	-0.1389	-0.1385	0.0004
18	9 (S)	-0.2737	-0.2758	-0.0021
19	10 (S)	-0.1188	-0.1185	0.0004
20	41 (S)	-0.1479	-0.1503	-0.0023
21	8 (R)	-0.0435	-0.0447	-0.0011
22	9 (R)	0.2715	0.2638	-0.0077
23	10 (R)	0.1186	0.1198	0.0012
24	40 (S)	-0.0188	-0.0195	-0.0007
25	25 (S)	-0.0942	-0.0931	0.0011
26	26 (S)	-0.0564	-0.0567	-0.0003
27	27 (S)	-0.1662	-0.1671	-0.0009
28	28 (S)	-0.0817	-0.0801	0.0015
29	12 (R)	0.1174	0.1195	0.0021
30	14 (R)	0.3390	0.3286	-0.0105
31	15 (R)	0.2369	0.2385	0.0016
32	17 (S)	-0.0771	-0.0777	-0.0006
33	18 (S)	-0.1785	-0.1852	-0.0067
34	19 (S)	-0.0695	-0.0696	-0.0001
35	16 (S)	0.2002	0.2011	0.0009
36	18 (R)	0.1763	0.1700	-0.0063
37	20 (R)	0.0143	0.0144	0.0001
38	30 (S)	-0.0519	-0.0520	-0.0001
39	22 (S)	-0.0567	-0.0565	0.0002
40	21 (S)	-0.0340	-0.0342	-0.0002
41	21 (R)	0.0339	0.0340	0.0001
42	24 (S)	0.0708	0.0695	-0.0012
43	29 (S)	0.0102	0.0104	0.0002
44	31 (R)	0.0700	0.0706	0.0006
45	32 (R)	0.0195	0.0191	-0.0005
46	33 (S)	-0.0025	-0.0027	-0.0001
47	34 (S)	-0.0355	-0.0375	-0.0020
48	34 (R)	0.0350	0.0362	0.0012
49	35 (R)	-0.0331	-0.0331	0.0000
50	37 (S)	-0.0621	-0.0615	0.0006
51	38 (S)	-0.0711	-0.0730	-0.0019
52	36 (R)	0.1663	0.1713	0.0050
53	41 (R)	0.1475	0.1491	0.0015
54	40 (R)	0.0188	0.0192	0.0004
55	37 (R)	0.0611	0.0620	0.0009

56	38 (R)	0.0693	0.0705	0.0012
57	39 (R)	0.0367	0.0362	-0.0005
58	38 (R)	0.0693	0.0665	-0.0028
59	39 (R)	0.0367	0.0370	0.0003

Line reactive power flow measurements

Meas. No.	Line No.	True value	Meas. value	Difference
1	1 (S)	0.0045	0.0045	-0.0001
2	2 (S)	-0.0735	-0.0714	0.0021
3	1 (R)	0.0257	0.0252	-0.0005
4	3 (S)	-0.0477	-0.0480	-0.0003
5	6 (S)	-0.0635	-0.0631	0.0004
6	5 (S)	-0.1130	-0.1090	0.0039
7	2 (R)	0.0824	0.0838	0.0014
8	4 (S)	-0.0704	-0.0712	-0.0008
9	3 (R)	0.0686	0.0689	0.0004
10	4 (R)	0.0726	0.0721	-0.0005
11	15 (S)	-0.0606	-0.0588	0.0018
12	7 (S)	-0.0646	-0.0648	-0.0002
13	5 (R)	0.1034	0.1026	-0.0008
14	8 (S)	-0.0263	-0.0259	0.0004
15	6 (R)	0.0731	0.0728	-0.0003
16	7 (R)	0.0667	0.0669	0.0001
17	11 (S)	0.0554	0.0559	0.0005
18	9 (S)	-0.0532	-0.0527	0.0005
19	10 (S)	-0.0957	-0.0945	0.0012
20	41 (S)	-0.0383	-0.0389	-0.0006
21	8 (R)	0.0457	0.0474	0.0017
22	9 (R)	0.0633	0.0636	0.0002
23	10 (R)	0.1036	0.1031	-0.0004
24	40 (S)	0.0293	0.0298	0.0005
25	25 (S)	-0.0561	-0.0573	-0.0012
26	26 (S)	-0.0802	-0.0791	0.0010
27	27 (S)	-0.1064	-0.1066	-0.0002
28	28 (S)	-0.0500	-0.0494	0.0006
29	12 (R)	0.0002	0.0005	0.0003
30	14 (R)	0.1226	0.1244	0.0019
31	15 (R)	0.0454	0.0452	-0.0001
32	17 (S)	-0.0167	-0.0171	-0.0003
33	18 (S)	-0.0387	-0.0379	0.0008
34	19 (S)	0.0020	0.0021	0.0000
35	16 (S)	0.0830	0.0833	0.0003
36	18 (R)	0.0342	0.0347	0.0005
37	20 (R)	-0.0009	-0.0013	-0.0004
38	30 (S)	-0.0105	-0.0105	0.0000
39	22 (S)	0.0022	0.0023	0.0001
40	21 (S)	0.0210	0.0210	0.0000
41	21 (R)	-0.0213	-0.0216	-0.0002
42	24 (S)	0.0461	0.0460	-0.0001
43	29 (S)	0.0085	0.0085	0.0000
44	31 (R)	0.0388	0.0376	-0.0012
45	32 (R)	-0.0062	-0.0062	0.0000
46	33 (S)	-0.0086	-0.0084	0.0003
47	34 (S)	-0.0238	-0.0237	0.0000
48	34 (R)	0.0230	0.0229	-0.0001
49	35 (R)	-0.0155	-0.0150	0.0005
50	37 (S)	-0.0170	-0.0170	0.0000

51	38 (S)	-0.0170	-0.0166	0.0004
52	36 (R)	0.0495	0.0504	0.0010
53	41 (R)	0.0496	0.0504	0.0008
54	40 (R)	0.0126	0.0126	0.0000
55	37 (R)	0.0152	0.0150	-0.0002
56	38 (R)	0.0136	0.0133	-0.0002
57	39 (R)	0.0054	0.0054	0.0000
58	38 (R)	0.0136	0.0133	-0.0003
59	39 (R)	0.0054	0.0053	-0.0002

Appendix 6

Appendix 6 lists the expected solution times of the 4 stage decomposed linear programming state estimator if the redundant dummy measurement equations were to be removed using a network observability algorithm to determine the critical dummy measurement equations required to maintain observability. For the purpose of the trial the redundant dummy measurement equations were identified and removed using a manual process based on information obtained from previous trials.

Comparison of the number of equations and variables for the four sub-estimation stages on the 30 substation test network with the dummy measurement equations removed for all the sub-estimation stages.

Sub-estimation type	Number of equations	Number of variables
Active power flow (P)	251	181
Reactive power flow (Q)	251	181
Voltage magnitude (V)	132	73
Voltage phase angle (θ)	110	73

Solution times for the Revised Simplex method during the estimation of the states of the 30 substation test network with with the dummy measurement equations removed for all the sub-estimation stages.

The times are in seconds and the number of Simplex iterations required to converge each sub-estimation stage is shown in brackets.

Solution times obtained from a flat start

Iteration number	Sub-estimation type P	Q	V	θ
1	1.71 (12)	2.26 (18)	3.36 (57)	1.36 (24)
2	1.55 (10)	3.03 (26)	2.17 (33)	0.74 (11)
3	1.94 (14)	3.94 (35)	0.54 (4)	0.67 (9)

Solution times obtained continuing from the above estimates with line 7 open

Iteration number	Sub-estimation type P	Q	V	θ
1	4.21 (38)	4.13 (37)	2.95 (46)	1.18 (20)
2	2.80 (24)	2.67 (22)	1.02 (13)	0.57 (6)
3	1.65 (10)	2.13 (16)	0.54 (4)	0.48 (4)

Comparison between the solution times of the reduced 4 stage linear programming and the Newton-Raphson least squares state estimation programs on the 30 substation test network.

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.
Linear proramming initialisation time = 23.50 s.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	7.34 (2)	29.45 (4)
Force to run	0.14 (0)	7.15 (1)
Open line 7	7.62 (2)	25.55 (3)
Open link 6	4.01 (1)	14.15 (2)
Close all switches	7.21 (2)	34.73 (4)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	8.84 (3)	29.66 (5)
Force to run	0.47 (1)	2.64 (1)
Open line 7	8.17 (3)	24.27 (4)
Open link 6	7.74 (3)	18.82 (3)
Close all switches	8.69 (3)	21.54 (4)
Measurement update	7.37 (2)	19.29 (3)

Comparison between the solution times of the reduced 4 stage linear programming and the Newton-Raphson least squares state estimation programs on the 30 substation test network. The linear programming estimator uses the reduced Netflow problem for the power flow sub-estimation stages and has the dummy measurement equations removed for the voltage sub-estimation stages.

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	7.34 (2)	26.06 (4)
Force to run	0.14 (0)	7.11 (1)
Open line 7	7.62 (2)	12.96 (3)
Open link 6	4.01 (1)	8.36 (2)
Close all switches	7.21 (2)	9.57 (4)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	8.84 (3)	34.23 (5)
Force to run	0.47 (1)	5.78 (1)
Open line 7	8.17 (3)	22.33 (8)
Open link 6	7.74 (3)	11.59 (4)
Close all switches	8.69 (3)	10.38 (3)
Measurement update	7.37 (2)	9.67 (3)

Comparison between the solution times of the reduced 4 stage linear programming and the Newton-Raphson least squares state estimation programs on the 57 substation test network. The linear programming estimator uses the reduced Netflow problem for the power flow sub-estimation stages and has the dummy measurement equations removed for the voltage sub-estimation stages.

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	28.03 (2)	42.23 (4)
Force to run	0.53 (0)	10.22 (1)
Open line 1	23.54 (2)	20.93 (4)
Open load 7	25.47 (2)	Failed to converge
Close all switches	31.07 (3)	19.47 (5)

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	32.65 (3)	53.09 (6)
Force to run	0.97 (1)	30.32 (4)
Open line 1	26.69 (3)	14.27 (4)
Open load 7	27.50 (3)	26.93 (10)
Close all switches	31.38 (3)	Failed to converge
Measurement update	24.95 (3)	17.36 (5)

Comparison between the solution times of the reduced 4 stage linear programming and the Newton-Raphson least squares state estimation programs on the 118 substation test network. The linear programming estimator uses the reduced Netflow problem for the power flow sub-estimation stages and has the dummy measurement equations removed for the voltage sub-estimation stages.

The times are in seconds and the number of iterations required to converge is shown in brackets.

Measurements subject to no noise or errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	202.94 (3)	287.11 (6)
Force to run	0.68 (0)	42.92 (1)
Open line 50	111.29 (2)	97.87 (5)
Open load 7	118.68 (2)	Failed to converge
Close all switches	141.92 (3)	Failed to converge

Measurements subject to 0.2% systematic noise and 1.5% random noise but no errors.

Sequence of events	Newton-Raphson Least squares	4 Stage Linear programming
Start estimator	213.56 (4)	Failed to converge
Force to run	4.70 (1)	Failed to converge
Open line 50	151.09 (3)	Failed to converge
Open load 7	169.41 (3)	Failed to converge
Close all switches	145.71 (3)	Failed to converge
Measurement update	145.47 (3)	Failed to converge

References and Bibliography

1. Aboytes, F., Cory, B.J.,
"Identification of measurement, parameter and configuration errors in static state estimation",
Proc. IEEE. PICA. Conf., 1975, pp.298-302.

2. Al-Shakarchi, M.R.G.,
"Nodal iterative load flow",
M.Sc. Thesis, University of Manchester Institute of Science and Technology, Manchester, UK., 1973.

3. Allam, M.,
"Exact static state estimation algorithm retaining non-linearity",
Proc. 7th PSCC. Conf., 1981, pp.1007-1013.

4. Allemong, J.J., Radu, L., Sasson, A.M.,
"A fast and reliable state estimation algorithm for AEP's new control centre",
IEEE Trans. Pas., Vol.101, No.4, Apr 1982, pp.933-944.

5. Alsac, O., Stott, B.,
"Optimal load flow with steady state security",
Proc. IEEE. PES. Summer meeting, 1973, Paper No.T73 484-3, also IEEE. Trans. PAS., Vol.93, No.3, May 1974, pp.745-751.

6. Amelink, H., Hoffmann, A.G.,
"Current trends in control centre design",
Int. Jou. Electrical Power & Energy Systems, Vol.5, No.4, Oct 1983, pp.205-211.

7. American National Standards Institute Inc.,
"American national standard programming language FORTRAN, (FORTRAN-77, ANSI X3.9 - 1978)",
American National Standards Institute, 1430 Broadway, New York, New York 10018.

8. Anderson, T.,
"An introduction to multivariate statistical analysis",
Wiley, 1958.

9. Arafeh, S.A., Schinzinger, R.,
"Estimation algorithms for large scale power systems",
Proc. IEEE. PES. Winter meeting, 1977, Paper No.A77 145-6, also IEEE. Trans PAS., Vol.98, No.6, Nov 1979, pp.1968-1977.

10. Aschmoneit, F., Denzel, D., Graf, R., Schellstede, G.,
"Development of an optimal state estimator and implementation in a real time computer system",
Proc. Cigre Conf., 1976, pp.1-12.

11. Bermudez, J.F.,
"An introduction to power system state estimation",
M.Sc., Thesis, University of Manchester Institute of Science and Technology, Manchester, U.K., 1975.

12. Bermudez, J.F.,
"Efficient static state estimation",
Ph. D., Thesis, University of Manchester Institue of Science and Technology, Manchester, U.K., 1977.

13. Bermudez, J.F., Brameller, A.,
"State estimation in power systems : A comparison of methods",
s",c. 6th PSCC. Conf., 1978, pp.783-790.

14. Bodger, P.S.,
"Modified and standard back-substitution fast-decoupled power flows compared to a Newton-like method for ill-conditioned system",
Proc. IEE., Vol.131, Pt.C., No.1, Jan 1984, pp.23-24.

15. Brameller, A., Allan, R.N., Haman, Y.M.,
"Sparsity",
Pitman, 1976.

16. Britton, J.P.,
"Improved load flow performance through a more general equation form",
IEEE. Trans. PAS., Vol.90, No.1., Jan 1971, pp.109-116.

17. Broussolle, F.,
"State estimation in power systems : Detecting bad data through the sparse inverse matrix method",
IEEE. Trans. PAS., Vol.97, No.3, May 1978, pp.678-682.

18. Brown, H.E.,
"Solution of large networks by matrix methods",
Wiley, 1976.

19. Brown, R.J., Tinney, W.F.,
"Digital solutions for large power networks",
AIEE. Trans. PAS., Vol.76, Jun 1957, pp.347-355.

20. Bubenko, J., Petterson, L.O.,
"State estimation in electric power systems using filtering techniques",
Proc. 4th PSCC Conf., 1972, Paper No.3.3/6.

21. Bunch, J.R., Rose, D.J.,
"Sparse matrix computations",
Academic Press, 1976.

22. Burchett, R.C., Happ, H.H., Vierath, D.R.,
"Quadratically convergent optimal power flow",
IEEE. Trans. PAS., Vol.103, No.11, Nov 1984, pp.3276-3275.

23. Cheng, D.T-Y.,
"Active power generation re-scheduling and load shedding to alleviate overloads - a network flow method",
Internal Research Report, Department of Engineering, Durham University, Nov 1982.

24. Cheng, D.T-Y.,
"An interactive load shedding technique for electric power systems control",
Internal Research Report, Department of Engineering, Durham University, Feb 1983.

25. Clegg, F.,
"Simple statistics",
Cambridge University Press, 1982.

26. Clements, K.A., Denison, O.J., Ringlee, R.J.,
"A multi-area approach to state estimation in power system networks",
Proc. IEEE. PES. Summer meeting, Jul 1972, Paper No. C72 465-3.

27. Clements, K.A., Krumpholz, G.R., Davis, P.W.,
"Power system state estimation residual analysis : An algorithm using network topology",
IEEE. Trans. PAS., Vol.100, No.4, May 1981, pp.1779-1787.

28. Clewer, B.C., Irving, M.R., Sterling, M.J.H.,
"Robust state estimation in power systems using sparse linear programming",
Proc. IEE., Vol.132, Pt.C, No.3, May 1985, pp.123-131.

29. Couch, G.H.,
"Improved power system state estimation with constant gain matrices",
Proc. IEEE, Vol.64, Nov 1976, pp.1641-1643.

30. Couch, G.H., Morrison, I.F.,
"Data validation and topology determination for power system monitoring and control",
Proc. IEEE. PES. Summer meeting, 1974, Paper No.C74 361-2.

31. Cramer, H.,
"Mathematical Methods of Statistics",
Princeton University Press, 1946.

32. Dantzig, G.B.,
"Linear programming and extensions",
Princeton University Press, 1963.

33. Debs, A.S., Larson, R.E.,
"A dynamic estimator for tracking the state of a power system",
IEEE. Trans. PAS., Vol.89, No.7, Sep 1970, pp.1670-1678.

34. Deo, N.,
"Graph theory with applications to engineering & computer science",
Prentice-Hall, 1974.

35. Dommel, H.W., Sato, N.,
"Fast transient stability solutions",
Proc. IEEE. PES. Winter meeting, 1972, Paper No.T72 137-3, pp.253-260.

36. Dopazo, J.F., Ehrmann, S.T., Klitin, O.A., Sasson, A.M., Van Slyck, L.S.,
"The AEP real-time monitoring and control computer system",
IEEE. Trans. PAS., Vol.95, No.5, Sep 1976, pp.1612-1617.

37. Dopazo, J.F., Ehrmann, S.T., Klitin, O.A., Sasson, A.M., Van Slyck, L.S.,
"Implementation of the AEP real-time monitoring system",
IEEE. Trans. PAS., Vol.95, No.5, Sep 1976, pp.1618-1629.

38. Dopazo, J.F., Klitin, O.A., Sasson, A.M.,
"State estimation for power systems : Detection and identification of gross measurement errors",
Proc. IEEE. PICA Conf., 1973.

39. Dopazo, J.F., Klitin, O.A., Stagg, G.W., Van Slyck, L.S.,
"State calculation of power systems from line flow measurements",
IEEE. Trans. PAS., Vol.89, No.7, Sep 1970, pp.1698-1708.

40. Dopazo, J.F., Klitin, O.A., Van Slyck, L.S.,
"State calculation of power systems from line flow measurements : Part II",
IEEE. Trans. PAS., Vol.91, No.1, Jan 1972, pp.145-151.

41. Dy Liacco, T.E.,
"State estimation in control centres",
Int. Jou. Electrical Power and Energy Systems, Vol.5, No.4, Oct 1983, pp.218-221.

42. Elgerd, O.I.,
"Electric energy systems and theory : An introduction",
McGraw-Hill, 1971.

43. Even, S.,
"Graph algorithms",
Pitman, 1979.

44. Falcao, D.M., Cooke, P.A., Brameller, A.,
"Power system tracking state estimation and bad data processing",
IEEE. Trans. PAS., Vol.101, No.2. Feb 1982, pp.325-333.

45. Fischl, R., Haplin, T.F., Helferty, J.J., Gershman, V., Mercede, F.,
"An algorithm for automatically tuning the weights of performance indicies for monitoring power system loading or security",
Proc. IEEE. PICA Conf., 1985, pp.345-351.

46. Frazer, D.A.S.,
"Statistics : An introduction",
Wiley, 1958.

47. Freris, L.L., Sasson, A.M.,
"Investigation of the load flow problem",
Proc. IEE. Vol.115, No.10, Oct. 1968, pp.1459-1470.

48. Galiana, F.D., Handschin, E.,
"Combined network and power station dynamic state estimation",
Proc. 4th PSCC Conf., 1972, Paper No.3.3/8.

49. Garcia, A., Monticelli, A., Abreu, P.,
"Fast decouple state estimation and bad data processing",
IEEE. Trans. PAS., Vol.98, No.5, Sep 1979, pp.1645-1652.

50. Garver, L.L.,
"Transmission network estimation using linear programming",
IEEE. Trans. PAS., Vol.89, No.7, Sep 1970, pp.1688-1697.

51. Gentle, J.E.,
"Least absolute value estimation : An introduction",
Commun. Statist. Simula. Computa., Vol.B6, 1977, pp.313-328.

52. Glimn, A.F., Stagg, G.W.,
"Automatic calculation of load flows",
AIEE. Trans. PAS., Vol.76, Oct 1957, pp.817-828.

53. Gu, J.W., Clements, K.A., Krumpholz, G.R., Davis, P.W.,
"The solution of ill-conditional power system state estimation problems via the method of Peters and Wilkinson",
Proc. IEEE. PICA. Conf., 1983, pp.239-246.

54. Handschin, E.,
"Real-time data processing using state estimation in electric power systems",
from "Real-time control of electric power systems", Handschin, E., (Ed.), Elsevier, 1972.

55. Handschin, E., Bongers, C.,
"Theoretical and practical considerations in the design of state estimators for electric power systems",
from "Real-time control of electric power systems", Handschin, E., (Ed.), Elsevier, 1972.

56. Handschin, E., Schweppe, F.C., Kohlas, J., Fiechter, A.,
"Bad data anlaysis for power system state estimation",
IEEE. Trans. PAS., Vol.94, No.2, Mar 1975, pp.329-337.

57. Haplin. T.F., Fischl, R., Fink, R.,
"Analysis of automatic contingency selection algorithms",
IEEE. Trans. PAS., Vol.105, No.5, May 1984, pp.938-945.

58. Happ, H.H.,
"Optimal power dispatch - A comprehensive survey",
IEEE. Trans. PAS. Vol.96, May 1977, pp.841-854.

59. Hestenes, M.R., Stiefel, E.,
"Methods of conjugate gradients for solving linear systems",
Jou. Res. Nat. Bur. Standards, Sect.B, Vol.49, 1952, pp.409-436.

60. Hobson, E., Fletcher, D.L., Stadlin, W.D.,
"Network flow linear programming techniques and their application to fuel scheduling and contingency analysis",
IEEE. Trans. PAS., Vol.103, No.7, Jul 1984, pp.1684-1691.

61. Horisberger, H.P., Richard, J.C., Rossier, C.,
"A fast decoupled static state estimator for electric power systems",
IEEE. Trans. PAS., Vol.95, No.1, Jan 1976, pp.208-215.

62. Housos, E., Irisarri, G.,
"Security assessment and control system : Corrective strategies",
IEEE. Trans. PAS., Vol.104, No.5, May 1985, pp.1075-1083.

63. Inoue, N., Daniels, H., Kilmer, R., Echave, V.,
"Implementation, testing and field installation of the Iberduero power system state estimator",
Proc. IEEE. PICA Conf., 1983, pp.11-18.

64. Irving, M.R.,
"Decomposed estimation and control of electric power systems",
Ph.D. Thesis, Sheffield University, Sheffield, U.K., 1977.

65. Irving, M.R., Owen, R.C., Sterling, M.J.H.,
"Power system state estimation using linear programming",
Proc. IEE. Vol.125, Pt.C., No.9, Sep 1978, pp.879-885.

66. Irving, M.R., Sterling, M.J.H.,
"Substation data validation",
Proc. IEE., Vol.129, Pt.C., No.3, May 1982, pp.119-122.

67. Iwamoto, S., Tamura, Y.,
"A fast load flow method retaining non-linearity",
IEEE. Trans. PAS., Vol.97, No.5, Sept.1978, pp.1586-1599.

68. Jazwinski, A.H.,
"Stochastic processes and filtering theory",
Academic Press, 1969.

69. Johnson, W.A., Potts, G.W., Wrubel, J.N., Schulte, R.P.,
"On-line load flows from a system operator's viewpoint",
IEEE. Trans. PAS., Vol.102, No.6, Jun 1983, pp.1818-1822.

70. Kennington, J.L., Helgason, R.V.,
"Algorithms for network programming",
Wiley, 1980.

71. Keyhani, A., Abur, A.,
"Comparative study of polar and cartesian co-ordinate algorithms for power system state estimation problems",
Proc. IEEE., Vol.132, Pt.C, No.3, May 1985, pp.132-138.

72. Knight, U.G.,
"Power system engineering and maths",
Pergamon Press, 1972

73. Knight, U.G.,
"Some views on state estimation",
from "On-line operation and optimisation of distribution systems", IEE. Conf. 1976.

74. Kobayashi, H., Navita,S., Hamman, M.S.A.A.,
"Model co-ordination method applied to power system control and estimation problems",
Proc. IFAC/IFIP 4th Int. Conf. on Digital computer applications to process control, 1974, pp.114-128.

75. Kohlas, J.,
"On bad data suppression in estimation",
IEEE. Trans. AC., Vol.17, Dec 1972, pp.827-828.

76. Kotiuga, W.W.,
"Development of a least absolute value power system tracking state estimator",
IEEE. Trans. PAS., Vol.104, No.5, May 1985, pp.1160-1166.

77. Kotiuga, W.W., Vidyasagar, M.,
"Bad data rejection properties of weighted least absolute value techniques applied to static state estimation".
IEEE. Trans. PAS., Vol.101, No.4, Apr 1982, pp.844-853.

78. Krumpholz, G.R., Clements, K.A., Davis, P.W.,
"Power system observability : A practical algorithm using network topology",
Proc. IEEE. PES. Winter Meeting, 1980, Paper No.F80 229-5, also IEEE. Trans. PAS., Vol.99, No.4, May 1980, pp.1534-1542.

79. Larson, R.E., Hajdu, L.P.,
"Potential applications and on-line implementation of power system state estimation",
Wolf Management Services, Palo Alto. Calif. Contract 14-03-79378,
Bonneville Power Administration, Portland, Ore., final report, Jan 1969.

80. Larson, R.E., Peschon, J.,
"Feasibility study of system state estimation",
Wolf Management Services, Palo Alto. Calif. Contract 14-03-79378,
Bonneville Power Administration, Portland, Ore., interim report,
Oct 1968.

81. Larson, R.E., Tinney, W.F., Peschon, J.
"State estimation in power systems. Part 1: Theory and feasibility",
IEEE. Trans. PAS., Vol.89, No.3, Mar 1970, pp.345-352.

82. Larson, R.E., Tinney, W.F., QQHajdu, L.P., Piercy, D.S.,
"State estimation in power systems. Part 2: Implementation and applications",
IEEE. Trans. PAS., Vol.89, No.3, Mar 1970, pp.353-363.

83. Lee, T.H., Thorne, D.H., Hill, E.F.,
"A transportation method for economic dispatching - application and comparison",
IEEE. Trans. PAS., Vol.99, No.6, Nov 1980, pp.2373-2385.

84. Leite da Silva, A.M., Do Coutto Filho, M.B., de Queiroz, J.F.,
"State forecasting in electric power systems",
Proc. IEE., Vol.130, Pt.C, No.5, Sep 1983, pp.237-244.

85. Lo, K.L., Ong, P.S., McColl, R.D., Moffat, A.M., Sulley, J.L.,
"Development of a static state estimator. Part I: Estimation and bad data suppression",
IEEE. Trans. PAS., Vol.102, No.8, Aug 1983, pp.2486-2491.

86. Lo, K.L., Ong, P.S., McColl, R.D., Moffat, A.M., Sulley, J.L.,
"Development of a static state estimator. Part II: Bad data replacement and generation of pseudomeasurements",
IEEE. Trans. PAS., Vol.102, No.8, Aug 1983, pp.2492-2500.

87. Luenberger, D.G.,
"Introduction to Linear and non-linear programming",
Addison-Wesley, 1973.

88. Luo, J.S., Hill, E.F., Lee, T.H.,
"Power system economic dispatch via network approach",
IEEE. Trans. PAS., Vol.103, No.6, Jun 1984, pp.1242-1248.

89. Masiello, R.D., Schweppe, F.C.,
"A tracking static state estimator",
IEEE. Trans. PAS., Vol.90, No.2, Mar 1971, pp.1025-1033.

90. Merrill, H.M., Schweppe, F.C.,
"Bad data suppression in power system static state estimation",
IEEE. Trans. PAS., Vol.90, No.6, Nov 1971, pp.2718-2725.

91. Meyer, P.L.,
"Introductory probability and statistical applications",
Addison-Wesley, 1970.

92. Mili, L., Van Cutsen, Th., Ribbens-Pavella, M.,
"Hypothesis testing identification : A new method for bad data analysis in power system state estimation",
IEEE. Trans. PAS., Vol.103, No.11, Nov 1984, pp.3239-3252.

93. Monticelli A., Garcia, A.,
"Reliable bad data processing for real time state estimation",
IEEE. Trans. PAS. Vol.102, No.5, May 1983, pp.1126-1139.

94. Monticelli, A., Wu, F.F.,
"A method that combines internal state estimation and external network modelling",
Proc. IEEE. PES. Winter Meeting, 1984, Paper No.84 WM 034-5, also IEEE. Trans. PAS., Vol.104, No.1, Jan 1985, pp.91-103.

95. Mukai, H.,
"Parallel multiarea state estimation",
Electric Power Research Institute, Report of research project 1764-1, Jan 1982.

96. Nian-de, X., Shi-ying, W., Er-keng, Y.,
"A new approach for detection and identification of multiple bad data in power system state estimation",
IEEE. Trans. PAS., Vol.101, No.2, Feb 1982, pp.454-462.

97. Nicholson, H.,
"Sequential least squares estimation and the electrical network problem",
Proc. IEE., Vol.118, No.11, Nov 1971, pp.1635-1641.

98. Pines, S., Wolf, H., Bailie, A., Mohan, J.,
"Modifications of the Goddard minimum variance program for the processing of real data",
Analytical Mechanics Associates, Inc., Contract NAS 5-2535, report dated Oct 1964.

99. Pines, S., Wolf, H., Woolston, D., Squires, R.,
"Goddard minimum variance orbit determination program",
NASA-GSFC Report X-640-62-191, Oct 1962.

100. Prewett, J.N., Farmer, E.D., Lang, W.D., Jervis, P.,
"Studies of a state estimation procedure for a power system and its on-line performance",
from "On-line operation and optimisation of distribution systems", IEE. Conf., 1976.

101. Quintana, V.H., Simoes-Costa, A., Mier, M.,
"Bad data detection and identification techniques using estimation orthogonal methods",
IEE. Trans. PAS., Vol.101, No.9, Sep 1982, pp.3356-3364.

102. Ramarao, N,. Murali Krishna, G., Mohammed, M.A.F.,
"A new algorithm for power system state estimation",
Proc. IEEE., Vol.70, No.10, Oct 1982, pp.1244-1245.

103. Rao, N.D., Roy, L.,
"A cartesian co-ordinate algorithm for power system state estimation",
IEEE. Trans. PAS., Vol.102, No.5, May 1983, pp.1070-1082.

104. Rao, N.D., Tripathy, S.C.,
"Power system state estimation by the Lorenberg-Marquardt algorithm",
IEEE. Trans. PAS., Vol.99, No.2, Mar 1980, pp.695-702.

105. Reid, J.K.,
"Large sparse sets of linear equations",
Reid, J.K. (Ed.), Academic Press, 1971.

106. Reid, J.K.,
"A sparsity - exploiting variant of the Bartels-Golub decomposition for linear programming basis",
Report CSS 20, AERE., Harewell, 1975.

107. Rice, J.R.,
"The approximation of functions",
Addison-Wesley, 1964.

108. Rossier, C.A., Germond, A.,
"Network topology optimisation for power system security enhancement",
Proc. Cigre/IFAC. Symposium 39-83, 1983, Paper No.206-01, pp.1-5.

109. Sasson, A.M., Ehrmann, S.T., Lynch, P., Van Slyck, L.S.,
"Automatic power system network topology determination",
IEEE. Trans. PAS., Vol.92, No.2, Mar 1973, pp.610-618.

110. Sawicki, J., Wilkosz, K., Kremens, Z.,
"Fast method for identification of bad data of active power flows in in power network",
Proc. Cigre/IFAC. Symposium 39-83, 1983, Paper No.101-04, pp.1-5.

111. Schweppe, F.C., Handschin, E.J.,
"Static state estimation in electric power systems",
Proc. IEEE., Vol.62, No.7, Jul 1974, pp.972-982.

112. Schweppe, F.C., Wildes, J.,
"Power system static state estimation. Part I: Exact model",
IEEE. Trans. PAS. Vol.89, No.1, Jan 1970, pp.120-125.

113. Schweppe, F.C., XXRom, D.B.,
"Power system static state estimation. Part II: Approximate model",
IEEE. Trans. PAS., Vol.89, No.1, Jan 1970, pp.125-130.

114. Schweppe, F.C., ZZ
"Power system static state estimation. Part III: Implementation",
IEEE. Trans. PAS., Vol.89, No.1, Jan 1970, pp.130-135.

115. Simoes-Costa, A., Quintana, V.H.,
"A robust numerical technique for power system state estimation",
IEEE. Trans. PAS., Vol.100, No.2, Feb 1981, pp.691-698.

116. Sirisena, H.R., Brown, E.P.M.,
"Convergence analysis of weighted least-squared and fast decoupled weighted least-squares state estimation",
Int. Jou. Electrical Power and Energy Systems, Vol.6, No.2, Apr 1984, pp.75-78.

117. Sirisena, H.R., Brown, E.P.M.,
"Fast identification of bad data in power system tracking state estimation",
Electric Power System Research, Vol.4, 1981, pp.297-308.

118. Siroux, J.G., Adnet, M.,
"Calcul en temps reel des modules et des phrases des tensions aux noeuds du reseau de transport a partir des telemesures de puissances actives et reactives",
Rev. Gen. Elecl., Vol.63, Mar 1967, pp.624-688.

119. Sjelvgren, D., Andersson, S., Andersson, T., Nyberg, U., Dillon, T.S.,
"Optional operations planning in a large hydro-thermal power system",
IEEE. Trans. PAS., Vol.102, No.11, Nov 1983, pp.3644-3651.

120. Smith, O.J.M.,
"Power system state estimation",
IEEE. Trans. PAS., Vol.89, No.1, Jan 1970, pp.363-379.

121. Srinivasan, N., Prakasa Rao, K.S., Indulkar, C.S.,
"Some new algorithms for state estimation in power systems",
Proc. IEEE. PES. Summer meeting, 1983, Paper No.83 SM 354-8.

122. Stagg, G.W.,
"AEP 118 bus test system",
American Electric Power Service Corporation, 2 Broadway, New York 8, New York.

123. Stagg, G.W., Dopazo, J.F., Klitin, O.A., Van Slyck, L.S.,
"Techniques for the real time monitoring of power system operations",
IEEE. Trans. PAS., Vol.89, No.4, Apr 1970, pp.545-555.

124. Stagg, G.W., El-Abiad, A.H.,
"Computer methods in power system analysis",
McGraw-Hill, 1968.

125. Sterling, M.J.H.,
"Power system control",
Peter Peregrinus, 1978.

126. Sterling, M.J.H., Irving, M.R.,
"Constrained dispatch of active power by linear decomposition",
Proc. IEE., Vol.124, Pt.C, No.3, Mar 1977, pp.247-251.

127. Stevenson, W.D.,
"Elements of power system analysis",
McGraw-Hill, 1962.

128. Stott, B.,
"Effective starting process for Newton Raphson load flows",
Proc. IEE., vol.118, No.8, Aug 1971, pp.983-987.

129. Stott, B.,
"Review of load flow calculation methods",
Proc. IEEE., Vol.62, No.7, Jul 1974, pp.916-929.

130. Stott, B., Alsac, O.,
"Fast decoupled load flow",
Proc. IEEE. PES. Summer Meeting & EHV/UHV Conference, 1973, Paper No. T73 463-7, also IEEE. Trans. PAS., Vol.93, No.3, May 1974, pp.859-869.

131. Stott, B., Hobson, E.,
"Power system security control calculations using linear programming. Part I",
IEEE. Trans. PAS., Vol.97, No.5, Sep 1978, pp.1713-1720.

132. Stott, B., Hobson, E.,
"Power system security control calculations using linear programming. Part II",
IEEE. Trans. PAS., Vol.97, No.5, Sep 1978, pp.1721-1731.

133. Subramanian, A.K., Wilbur, J.,
"Power system security functions of the energy control centre at the Orange and Rockland utilities",
Proc. IEEE. PICA. Conf., 1983. pp.2-10.

134. Sullivan, A.G., Reichert, K., Saly,S.,
"An on-line technique for network topology determination for use in security assessment studies",
IEE. International conference on "On-line operation and optimisation of transmission and distribution systems", 1976, pp.61-66.

135. Sun, D.I., Ashley, B., Brewer, B., Hughes, A., Tinney, W.F.,
"Optional power flow by Newton method",
IEEE. Trans. PAS., Vol.103, No.10, Oct 1984, pp.2864-2880.

136. Tinney, W.F., Hart, C.E.,
"Power flow solution by Newton's method",
IEEE. Trans. PAS., Vol.86, No.11, Nov 1967, pp.1449-1967.

137. Van Cutsem, Th., Howard, J.L., Ribbens-Pavella, M.,
"A two-level static state estimator for electric power systems",
IEEE. Trans. PAS., Vol.100, No.8, Aug 1981, pp.3722-3732.

138. Van Cutsem, Th., Ribbens-Pavella, M.,
"Critical survey of hierarchical methods for state estimation of electric power systems",
IEEE. Trans. PAS., Vol.102, No.10, Oct 1983, pp.3415-3424.

139. Vemuri, S., Usher, R.E.,
"On-line automatic continency selection algorithms",
Proc. IEEE. PES. Summer Meeting, 1982, Paper No.82 SM 413-3.

140. Wallach, Y., Handschin, E., Bongers, C.,
"An efficient parallel processing method for power system state estimation",
IEEE. Trans. PAS., Vol.100, No.11, Nov 1981, pp.4402-4406.

141. Ward, B.J., Hale, H.W.,
"Digital computer solution of power flow problems",
AIEE. Trans. PAS., Vol.75, Jan 1956, pp.398-404.

142. Weedy, B.M.,
"Electric power systems",
Wiley, 1979.

143. Wilks, S.,
"Mathematical Statistics",
Wiley, 1962.

144. Wood, A.J., Wollenberg, B.F.,
"Power generation, operation and control",
Wiley, 1984.

145. Zhuang, F., Balasubramanian R.,
"Bad data suppression in power system state estimation with a variable quadratic-constant criterion",
Proc. IEEE. PES. Summer meeting, 1984, Paper No.84 SM 619-3, also IEEE. Trans. PAS., Vol.104, No.4, Apr 1985, pp.857-863.

146. Zhuang, F., Balasubramanian, R.,
"A transformation-decoupled estimator for power system state estimation",
Proc. IEEE PES. Summer Meeting, 1984, Paper No.84 SM 620-1.