OBSERVABILITY METHOD FOR THE LEAST MEDIAN OF SQUARES ESTIMATOR AS APPLIED TO POWER SYSTEMS

by

Michael G. Cheniae

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

Lamine Mili

Dr. Lamine Mili, Chairman

Dr. Arun G. Phadke

Dr. Jaime De La Ree

June, 1991

Blacksburg, Virginia

OBSERVABILITY METHOD FOR THE LEAST MEDIAN OF SQUARES ESTIMATOR AS APPLIED TO POWER SYSTEMS

by

Michael G. Cheniae

Dr. Lamine Mili, Chairman

Electrical Engineering

(ABSTRACT)

The formulation of an accurate data base consisting of system state variable values is an initial and critical step in the economical and secure operation of modern power systems. The Least Median of Squares (LMS) estimator is ideal in the sense that it can provide a good state estimate despite high percentages of bad data and multiple bad leverage points. The estimator is, however, computationally intensive.

In this thesis, an efficient algorithm is developed and implemented to increase the overall speed of the LMS estimator. The algorithm generates measurement samples in a manner that allows use of the resampling technique i.e., they make the system observable and also ensure that each measurement has a nearly equal probability of appearing in each of the measurement samples.

ACKNOWLEDGEMENTS

The author wishes to acknowledge Dr. Lamine Mili for his financial support and academic advice. Appreciation is also expressed to Dr. Arun Phadke and Dr. Jaime De La Ree for serving as committee members. I especially thank Dr. De La Ree for agreeing to serve as a member on such short notice.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1 Necessity For Accurate State Estimate

Modern high–voltage transmission systems are highly interconnected and vast in size. These factors necessitate a centralized control structure to ensure secure operation. This centralized control is exercised via a hierarchical system of local and regional control centers, each of which is responsible for the proper operation of the system within its jurisdiction.

The primary function of the control centers is to deliver power in a secure manner. Since a power system is composed of many interactive components, any of which may fail unexpectedly, a normally operating system may suddenly collapse. The removal of a single component, via component failure or the action of a protective device, can potentially cause the failure or removal of other system components in a cascading manner. To guard against such an occurrence, selected component failures are simulated using a computer model of the actual system. Based upon the results of these *contingency analyses,* the control center operators can alter the system configuration to prevent the computer–simulated outages from actually occurring.

The second major function of the control centers is to operate the system as economically as possible given the system constraints imposed by satisfying the reliability requirement. Finding the economically optimum system configuration is known as the *economic dispatch* problem and its solution also requires the use of

computer models of the actual system.

The operational decisions made by the control center operators are based heavily upon the output of the contingency analyses and economic dispatch programs. The viability of the output of these programs is directly related to the accuracy of the system model used by the programs. Determining an accurate system model or, equivalently, the actual system status is, therefore, an important first step which the control centers must accomplish.

1.2 State Estimation

The initial step in determining the status of the system is accomplished by the topology processor. The topology processor determines which transmission lines, generators, etc. are in service based upon the positions of the system circuit breakers and switches which are telemetered to the control centers. Once the topology of the system has been determined, one can proceed to evaluate the power flows and bus voltages associated with the system.

Due to the extremely large number of buses and lines in a typical power network, it is not feasible to measure all of the bus voltages and power flows. These quantities can, however, be readily calculated from the *system state.* The system state is defined by the state variables – the voltage magnitude and phase angle at all system buses. As previously mentioned, relatively few of the bus voltage magnitudes are actually measured and, currently, none of the phase angles are measured. Recent advances have made it possible to measure the bus voltage phase angles, but the use of this technology is still experimental and, even when

implemented in earnest, will be used to give the phase angles at relatively few buses. The remaining state variables must be determined via *state estimation.* The purpose of a state estimator is to assign good values to unknown system state variables based upon system measurements, equations describing the system, and a statistical criterion that is minimized. Redundant measurements are used by a robust estimator to filter out small errors and reject discordant measurements, the so–called bad data, referred to as outliers in the statistical field. In power systems, the measurements consist of selected bus voltage magnitudes, power flows, and power injections. The state variables are related to these measurements via the following non–linear equations:

$$P_i = \sum_{j=1}^{n} V_i V_j Y_{ij} \cos(\theta_i - \theta_j - \delta_{ij}) \quad i = 1,..., N$$

$$Q_i = \sum_{j=1}^{n} V_i V_j Y_{ij} \sin(\theta_i - \theta_j - \delta_{ij}) \quad i = 1,..., N$$

$$P_{ij} = V_i V_j Y_{ij} \cos(\theta_i - \theta_j - \delta_{ij}) - V_i^2 Y_{ij} \cos \delta_{ij}$$

$$Q_{ij} = V_i V_j Y_{ij} \sin(\theta_i - \theta_j - \delta_{ij}) - V_i^2 Y_{ij} \sin \delta_{ij}$$

where

P_i = *real power injection at node i*

Q_i = *reactive power injection at node i*

P_{ij} = *real power flow from node i to node j*

Q_{ij} = *reactive power flow from node i to node j*

θ_i = *the voltage phase angle at node i*

δ_{ij} = *the angle of impedance of the line connecting nodes i and j*

N = *the number of system buses*

These equations can be written in matrix form as follows:

$$\underline{z} = \underline{h}(\underline{x}) + \underline{e} \qquad (1)$$

where

$\underline{z}$ = *an mx1 vector consisting of real and reactive power flows, injections, and voltage magnitudes*

$\underline{h}$ = *the nonlinear functions relating the measurements to the state variables*

$\underline{x}$ = *the state vector*

$\underline{e}$ = *a measurement noise vector.*

Given a set of redundant system measurements and the above equations, the state estimator's objective is to determine a "best" estimate of the unknown state variables. This "best" estimate is not unique, but is dependent upon the estimator used. Many estimators have been developed with each being designed with different goals in mind. An estimator may be designed to maximize the probability that the state estimate is equal to the actual state when the measurements are distributed according to a particular probability distribution function. For example, the least squares estimator is the maximum likelihood estimator if the measurements are normally distributed. The assumed model of the measurement noise vector $\underline{e}$ is a major factor to consider when deciding which estimator is most appropriate for a specific application. An estimator which is optimal in one case, may give erroneous results if the assumptions behind the estimator are not met.

The measurement errors and, hence the measurements, encountered in power systems do not necessarily follow any probability distribution function. Measuring devices may fail, resulting in very large measurement errors. The devices may be wired incorrectly, giving erroneous readings. The communication systems used to telemeter the measurements to the control centers may also fail resulting in incorrect measurements or no measurements at all. Simple screening of the received measurements can identify those measurements having unrealistic values, but measurements may remain that cannot be readily determined to be bad.

Before proceeding with a discussion of some of the estimators currently being researched for use with power systems, it is necessary to make an additional comment concerning the measurements used by the state estimator. In order for any state estimator to be able to reach a solution, the measurements must be sufficient in number and in geographic distribution such that all of the state variables have a measurement associated with them. A system containing measurements meeting this criterion is called an *observable* system. The concept of system observability and methods available to determine if a given set of measurements make a system observable comprise the bulk of this paper and will be discussed in detail following the introduction. It is simply defined at this point because the term will appear during the following discussion of several estimators.

1.3 Least Squares and Least Absolute Value Estimators

The traditional approach used by the utility industry to arrive at a state estimate makes use of the least–squares (LS or L_2) estimator; usually coupled with some sort of residuals analysis. As previously stated, the L_2 is the maximum

likelihood estimator if the measurements are normally distributed, but power system measurements do not meet this assumption. The L_2 estimator, even when combined with some bad data rejection rules based on the residuals, exhibits several undesirable properties when it encounters data points departing from the gaussian assumption. First, it is subject to the *masking affect.* The state estimate is attracted by *outliers* (points which are distant from the majority of the data are called outliers). The result is that the residuals of the outliers are small. The outliers are hidden or masked from detection. The second problem with this estimator is that it is attracted by bad *leverage points* (a leverage point is a data point which is far away from the bulk of the point cloud in the factor space of the regression). It has been shown that power system models contain many leverage points [1]. The result is that the L_2 estimator, although widely used, is not the appropriate choice for determining a reliable state estimate of power systems.

Another estimator that is being applied to the power system state estimation problem is the least absolute value (LAV or L_1) estimator [2–7]. The L_1 is able to handle vertical outliers; however, it is unable to give accurate state estimates in the presence of even a single bad leverage point. All of the M–type estimators (of which the L_1 is but one example) exhibit this same weakness [1].

1.4 Least Median of Squares Estimator

Recently, attention has been directed to a family of estimators known collectively as high breakdown point estimators. Some of these estimators are capable of giving accurate state estimates despite bad data percentages approaching 50%, and are also not adversely affected by bad leverage points. One such estimator

is the Least Median of Squares (LMS) estimator. This estimator has been shown to indeed have the ability to handle both vertical outliers and bad leverage points [1]. Although the LMS has many advantages, it is a computationally intensive estimator. The computation of the LMS estimate involves the repeated selection of different measurement samples of size n (where n is the number of state variables to be estimated) for which the system is observable. For each of these samples, the Newton–Raphson algorithm is used to solve for an estimate of the state of the system. The weighted residuals are computed and then ordered by increasing squared values. The state estimate which results minimizes the LMS criterion. Ideally, all of the measurement samples meeting the observability criterion should be considered. Power system sizes prohibit an exhaustive search for the optimum state by trying all suitable measurement combinations.

1.5 Resampling Technique

The resampling technique [8] is one method whose use allows the number of such measurement sets to be substantially reduced. Assuming that each measurement has an equally likely chance of being selected and included in a measurement set, the number of measurement sets k that need to be examined to have a probability P of obtaining at least one uncontaminated set can be expressed as follows.

$$P = 1-(1-(1-\epsilon)^n)^k, \qquad (2)$$

where ϵ denotes the fraction of contaminated data and n denotes the number of data points in each set, i.e. the dimension of the problem or size of the network. The

probability P is set equal to nearly 1 (typically 0.95). This equation allows one to determine the number of samples that must be considered to achieve a desired probability of obtaining an uncontaminated data set for a given fraction of contamination. Although the resampling equation reduces the number of measurement sets that must be considered, the number increases exponentially with the size of the system.

The number of measurement sets which must be created for the LMS may be further reduced via a system decomposition scheme which allows the LMS to be run on relatively small subsystems. Considering the fact that the number of measurement sets required grows exponentially with system size, the decomposition scheme offers a substantial reduction in the number of measurement sets that need be considered.

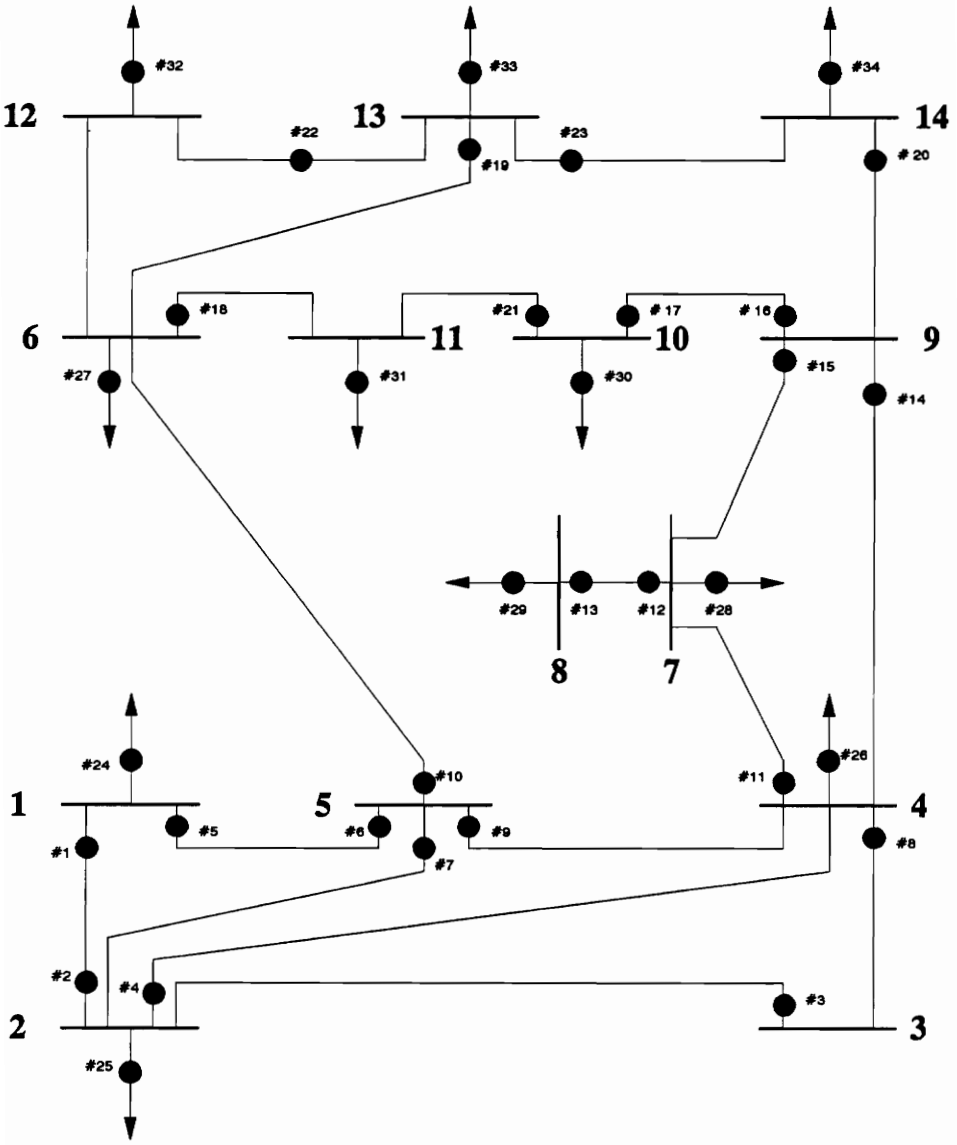
1.6 Observability and The Resampling Technique

Despite the use of techniques to reduce the number of measurement samples required for the LMS to be applied effectively to power system state estimation, the requirement for a fairly large number of measurement samples for which the system (or each of the decomposed subsystems) is observable remains. In addition, each measurement should have an equal probability of appearing in the measurement samples so that the resampling method can be used. Obtaining these sets in a rapid manner is critical to the use of the LMS in real time applications.

The method currently used to generate the measurement samples proceeds as follows. The measurements are first grouped into their *fundamental sets*. Each

fundamental set contains those measurements which are functions of the same state variable. Since there are two state variables associated with each system bus (one associated with the reference bus), it follows that the number of fundamental sets for a network is equal to the number of buses in the network. Utilizing the fact that it is a necessary condition of observability that at least one measurement from each fundamental set be contained in the sample [1], a measurement is randomly drawn from each. This procedure, instead of simply selecting n measurements from the ungrouped set, increases the probability of drawing a sample for which the system is observable. However, since including a measurement from each of the fundamental sets is only a necessary condition of system observability, the measurement set drawn must be subjected to an observability test to ensure that they actually do result in an observable system. Simple topological tests are first employed to reject samples containing power flow measurements situated on (a) both ends of a line, and (b) all lines incident to a node whose injection measurement was drawn. Measurement samples passing these simple tests are then subjected to a numerically–based observability analysis algorithm to make a final observability determination.

This method has several weaknesses. First, the percentage of measurement samples passing the simple topological tests but failing the final observability test is high. For the IEEE 14–bus system with the measurement configuration shown in figure 1, the percentage is approximately 44%. This failure rate increases for larger systems since the topological checks used are unable to detect the formation of loops by either flows or injections and the number of loops increases with system size. As a result, much time is wasted testing samples which prove to be unusable.

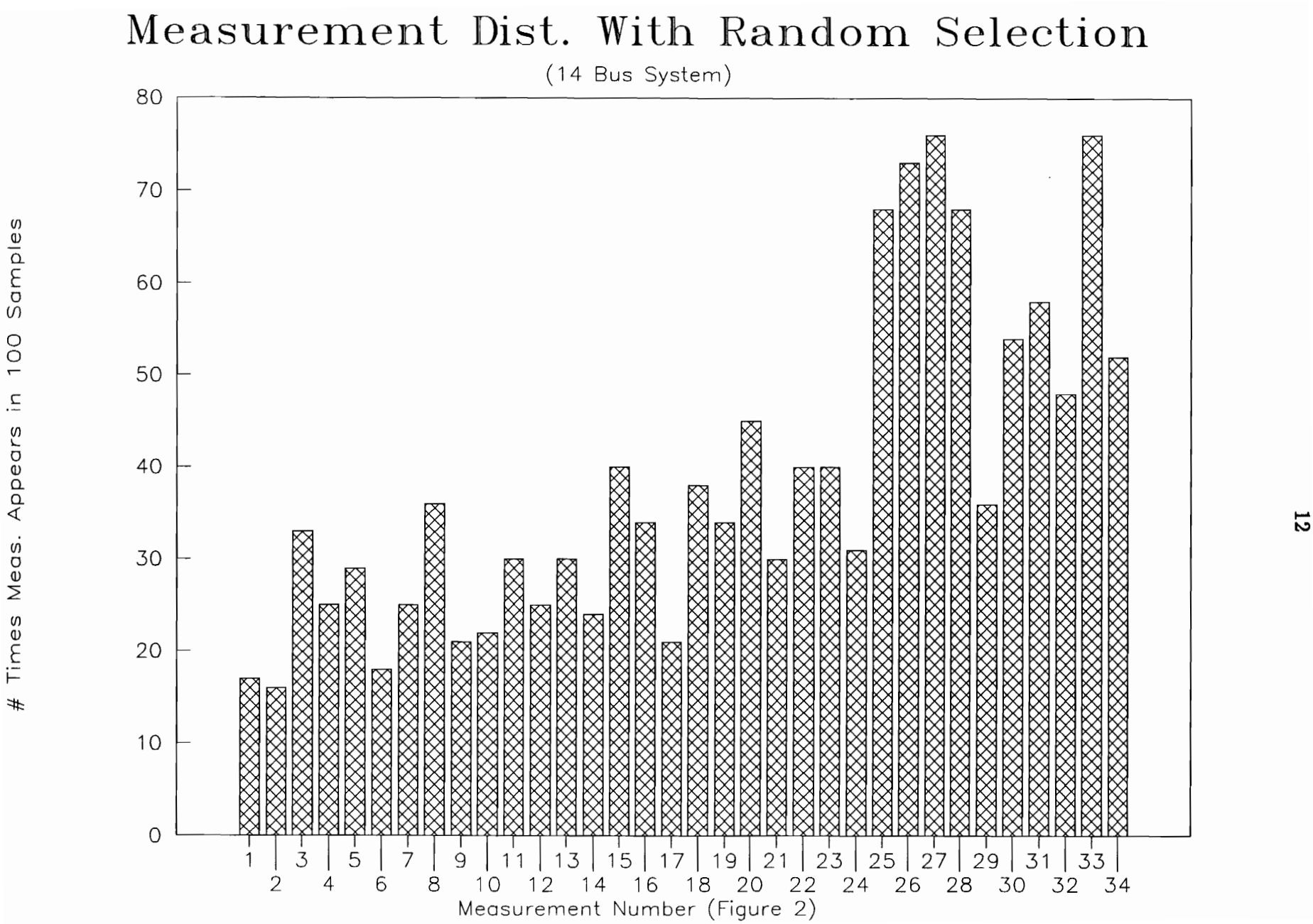


Actual Network

Figure 1

Including the samples that fail either the initial topological tests or the final observability check, the percentage of drawn samples that fail to make the system observable reaches 84% for the IEEE– 14 bus system. For the IEEE 118–bus system, this percentage approaches 100%. A second weakness is that the numerically–based observability algorithm used as the final check is relatively slow when compared to some topologically–based algorithms. Thirdly, the correct determination of system observability using this method is also dependent upon the effective zero utilized in the triangular factorization process.

The most important weakness of the above method is that it does not satisfy the assumptions behind the resampling method. Since injection measurements appear in the fundamental sets associated with all of the nodes incident to the node on which the injection is located, each injection measurement appears in several fundamental sets. In contrast, each flow measurement appears in only two fundamental sets. The result of this measurement distribution is that randomly selecting a measurement from each of the fundamental sets results in a higher probability of selecting an injection. Figure 2 shows the distribution of the measurements in 100 measurement samples generated by the above method for the IEEE 14–bus system of figure 1. It is clear from the plot that the injections (measurements 24–34) appear more frequently than do the flow measurements. This means that injections are used to a greater extent in determining the optimal solution in the LMS sense. If some of these injections correspond to the bad measurements that are to be identified by the LMS estimator, their presence in a high percentage of measurement sets means that the probability of obtaining an uncontaminated measurement sample cannot be given by equation 2.


Measurement Dist. With Random Selection
(14 Bus System)
Times Meas. Appears in 100 Samples
0
10
20
30
40
50
60
70
80
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
Measurement Number (Figure 2)

1.7 Thesis Objective

This thesis develops an algorithm to generate measurement samples for use with the LMS estimator. The final algorithm overcomes the weaknesses of the currently used method as presented above. The algorithm is efficient, correct, and ensures that each measurement has a nearly equal probability of appearing in a given measurement sample.

Chapter 2 begins with an historical overview of the development of observability analysis algorithms. The basic principles of observability are then presented and several recent algorithms claiming to be fast and correct are analyzed.

Chapter 3 describes in detail an observability analysis algorithm based upon the concept of intersecting matroids. The theory of matroids and their use in power system observability analysis are presented along with several examples which serve to make the concepts clear. This observability analysis technique is used in the final algorithm used to generate the measurement sets.

Chapter 4 describes the final algorithm in detail. Techniques utilized to increase the efficiency of the algorithm are presented, as is a method which ensures an almost uniform measurement distribution within the generated measurement sets. Comparisons between the current method and the new method are also presented in this chapter.

CHAPTER 2

OBSERVABILITY ANALYSIS

2.1 State–of–the–Art in Observability

There have been many papers published concerning power system observability theory and analysis methods since the subject was first examined by Clements and Wollenberg [9] in 1975. The major objective of each of these papers has been to develop and implement observability analysis algorithms that yield a necessary and sufficient condition for observability and are computationally efficient. This common goal has resulted in a number of different fundamental approaches to the problem, as well as a variety of implementation schemes for each approach.

Initial observability analysis algorithms were heuristic in nature [9–11] and only verified a sufficient condition for observability. These algorithms may incorrectly label observable networks as being unobservable.

A major change in the approach to the observability problem was pioneered by Krumpholz and Clements [12] in the early 1980's. Krumpholz and Clements made the connection between classical graph theory and power system observability when they introduced the concept of topological observability and proved that a necessary and sufficient condition for network observability is that a spanning tree can be formed in the system one–line diagram. Clements and his colleagues have successively refined the topologically–based algorithm [13–15] initially presented in [12].

Many of the more recent papers dealing with the subject of observability have presented additional topologically–based algorithms. Van Custem [16] attempts to form a spanning tree by treating the injections in an enumerative way, claiming that the high proportion of flow measurements typical of power systems makes such an approach competitive with non–enumerative algorithms. One way to avoid an enumerative search is by reformulating the problem as a maximum flow problem [17]. This method has been shown to be quite slow [18] as are all of the observability algorithms based upon classical graph–theory.

Another major class of observability analysis algorithms is based on factorization of the Jacobian matrix in complex form. The goal is to determine if the matrix is of full rank indicating that the overdetermined system of equations describing the power network can be solved. Numerically–based observability algorithms were first introduced by Monticelli and Wu who presented the supporting theory in [19] and associated algorithms in [20–21]. The advantages of such algorithms are that they use computer subroutines already in existence at control centers and that system line parameters are taken into consideration. The major disadvantages of such algorithms are their susceptibility to finite arithmetic errors and their requirement for the selection of an appropriate effective zero [22].

Symbolic factorization algorithms have been developed in an attempt to avoid the disadvantages associated with the factorization of the measurement Jacobian matrix. The initial algorithm [23] was subject to incorrect determination of system observability [24]. A more recent algorithm by Clements and Davis [25] avoids this problem, but the efficiency of the algorithm cannot be judged since

computing times are not presented in the paper.

Reformulating the network observability problem as a maximum–cardinality matroid intersection problem has resulted in two major papers. Quintana et al. [26] introduces the concept and presents an algorithm. The same concept is utilized by Nucera and Gilles in [27], where they develop a much more efficient algorithm based on augmenting sequences. This algorithm has been shown to be faster than an optimized numerically–based method, thereby, disproving the claim that topologically–based procedures are necessarily slower than numerically–based ones.

2.2 Definition and Principles

In the introduction it was stated that the measurement samples generated for the LMS estimator must make the system observable. In this section the concept of observability will be discussed and methods of determining network observability will be presented.

One definition of observability [19] is that a network is observable if any flow in the network can be observed by the set of measurements. Equivalently, a network is observable if, whenever all measurements are equal to zero, all flows are equal to zero. If some nonzero network flows actually do exist when all measurements read zero, the network branches in which these flows exist are called unobservable branches. A more intuitive definition of observability is as follows. A network is said to be observable if the measurements are sufficient in number and in geographic distribution such that the state of the system can be determined.

The state of the system is determined by finding a solution to the overdetermined set of equations represented by Equation (1). Since this equation is nonlinear, a direct solution is not possible and an iterative solution technique is required. The Newton–Raphson algorithm is the method most commonly used. At each iteration, Equation (1) is linearized about an operating point $\hat{\underline{x}}$ via a first–order Taylor series expansion. The result being

$$\Delta \underline{z} = \underline{H} \Delta \underline{x}, \qquad (3)$$

where $\underline{H}$ is the Jacobian matrix, $\underline{H} = d\underline{h}(\underline{x})/d\underline{x}$. Normally, an iterative solution technique would involve the evaluation of the Jacobian matrix at each new $\hat{\underline{x}}$. Since power systems are operated near a known $\hat{\underline{x}}$, the flat voltage profile ($\underline{x} = 1\angle 0°$), it is possible to converge to a solution using a Jacobian matrix whose entries are evaluated at the flat voltage profile and maintained constant throughout the iterative procedure. Making use of this approximation saves computing time. Equation (3) implies that the state of the system can be estimated if and only if $\underline{H}$ is of full rank. Observability can, therefore, be defined in terms of the rank of the Jacobian matrix. A system is said to be *algebraically observable* if the Jacobian matrix associated with the system is of full rank.

Although the Jacobian matrix may be of full rank, it does not necessarily follow that Equation (3) can be solved. The solvability of Equation (3) is also dependent upon the condition number of the Jacobian matrix. An ill–conditioned

matrix may result in the accumulation of finite arithmetic errors during the iterative process such that the solution does not converge or converges to an incorrect solution. A system for which this phenomenon occurs is numerically unobservable. Conversely, a system is said to be *numerically observable* if a solution to Equation (3) is reached via the Newton–Raphson algorithm. If a system is numerically observable, it is necessarily algebraically observable, but the converse need not be true. For most power systems under normal operating conditions, algebraic observability does imply numerical observability [12]. As a result, observability analysis algorithms can be based simply upon the determination of the rank of $\underline{H}$.

The approximate Jacobian matrix can be expressed as [22]

$$\underline{M}\,\underline{Y}\,\underline{A}^t,$$

where $\underline{M}$ is the measurement–branch incidence matrix whose entries are defined as

$$m_{ij} = \begin{cases} 1 \text{ if branch } i \text{ is incident to branch } j \text{ at the from end} \\ -1 \text{ if branch } i \text{ is incident to branch } j \text{ at the to end} \\ 0 \text{ if branch is not incident to branch } j \end{cases}$$

Here, $\underline{Y}$ is a diagonal matrix whose entries are equal to the line admittances and $\underline{A}$ is the node–branch incidence matrix whose entries are defined as

$$a_{ij} = \begin{cases} 1 \text{ if branch } j \text{ is directed towards node } i \\ -1 \text{ if branch } j \text{ is directed away from node } i \\ 0 \text{ if branch } j \text{ is not incident to node } i \end{cases}$$

The elements of $\underline{H}$ are complex values due to the complex nature of the entries of $\underline{Y}$. Expressing $\underline{H}$ in this manner has the advantage that line parameters as well as network topology are taken into account.

2.3 Comparison of Several Observability Analysis Algorithms

2.3.1 *Triangular Factorization*

Determining the rank of $\underline{H}$ expressed in complex form involves a floating point computation. One such method involves triangular factorization of the matrix [19]. This method has the advantage that it can use some of the same subroutines used by the Newton–Raphson solution algorithm. The disadvantages of this method are that it is subject to finite arithmetic errors and it is relatively slow when compared to algorithms which operate on the Jacobian matrix expressed in a form which neglects the system line parameters.

2.3.2 *Gramian*

One way to avoid some of the finite arithmetic problems encountered in the observability analysis method mentioned above is to set the line impedances of the system equal to unity. This amounts to replacing $\underline{Y}$ with the identity matrix. The rationale supporting such a simplification of the Jacobian matrix is based upon the fact that the goal of system observability analysis is essentially deciding whether or not the system measurements are sufficient in number and in geographic distribution such that they can detect any nonzero flows that exist in the network. The actual values of these flows, which are dependent upon the line parameters, is not an issue. Replacing $\underline{Y}$ with the identity matrix results in a Jacobian matrix whose entries are integers. The matrix in this form describes the measurement

distribution within the network strictly in a topological sense.

Expressing $\underline{H}$ in integer form allows the use of methods other than triangular factorization to determine the Jacobian matrix's rank. One proposed method is to calculate the Gram determinant or *Gramian* of the measurement vectors [28]. The Gramian of a set of vectors $\{\underline{v}_1, ..., \underline{v}_k\}$ is defined as

$$G = \begin{vmatrix} \underline{v}_1 \cdot \underline{v}_1 & \underline{v}_1 \cdot \underline{v}_2 & \cdots & \underline{v}_1 \cdot \underline{v}_k \\ \underline{v}_2 \cdot \underline{v}_1 & \underline{v}_2 \cdot \underline{v}_2 & \cdots & \underline{v}_2 \cdot \underline{v}_k \\ \cdots & \cdots & \cdots & \cdots \\ \underline{v}_k \cdot \underline{v}_1 & \underline{v}_k \cdot \underline{v}_2 & \cdots & \underline{v}_k \cdot \underline{v}_k \end{vmatrix}$$

The Gramian is a determinant whose entries are composed of all possible dot products of the given vectors. Using the rows of the integer Jacobian matrix as the given vectors, the Gramian can be written as

$$\text{Gramian} = \underline{H}^t \cdot \underline{H}.$$

A necessary and sufficient condition that the vectors $\{\underline{v}_1, ..., \underline{v}_k\}$ be linearly dependent is that the Gramian be equal to zero. Equivalently, the vectors $\{\underline{v}_1, ..., \underline{v}_k\}$ are linearly independent if and only if $G \neq 0$ [29].

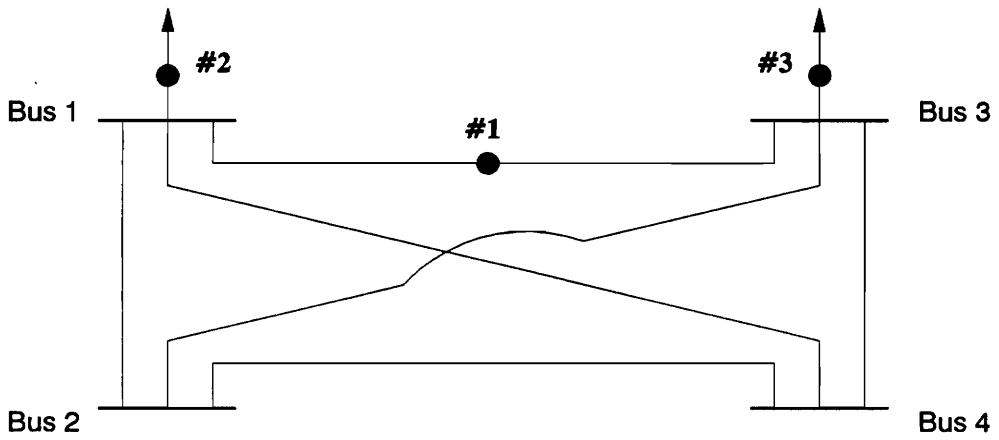
System observability requires that $\underline{H}$ be of full rank. This is equivalent to

the column vectors of $\underline{H}$ being linearly independent. Thus, a Gramian composed of measurement vectors and whose value is not equal to zero indicates an observable system. If such a Gramian is equal to zero, the system is unobservable. Efficient mixed–radix algorithms which can determine the rank of integer matrices have been developed [30–33].

There are two major disadvantages to using the value of the Gramian determinant to evaluate the observability of a system. The first weakness of this method is that it gives only a strict yes/no answer to the observability problem. For unobservable systems one does not gain any knowledge as to which measurements are dependent and, as a result, where additional measurements such as pseudo–measurements should be placed in the system.

Secondly, replacing the line impedance matrix by the identity matrix can result in labeling observable networks as unobservable. This false conclusion is known as *parametric unobservability* [12]. An example of a network which results in this phenomenon is shown in figure 3. With all of the line impedances set equal to unity, the Gramian is equal to zero indicating that the network is not observable. Using the line impedances shown in figure 3, the Gramian is equal to 0.052 indicating that the network is observable. (There are many values of line impedances, both real and integer, that can be used to make the same point). Based on these weaknesses, this method of observability analysis is rather poor.

Another method which can be used to determine the observability of a network based upon the Jacobian matrix expressed in integer form was proposed by



$X_{12} = 0.2$ pu

$X_{13} = 0.3$ pu

$X_{14} = 0.9$ pu

$X_{23} = 1.0$ pu

$X_{24} = 0.6$ pu

$X_{34} = 0.7$ pu

Example Network for Gramian Method

Figure 3

Chen [24]. A significant amount of time was spent testing this algorithm with the conclusion being twofold. First, Chen's method is not based upon the simple determination of the rank of H as stated in [22]. It is actually a topologically–based observability algorithm which attempts to build a maximum spanning tree. Secondly, the algorithm has one significant weakness. Before proving these assertions, it is necessary to develop the basic concepts of graph theory and discuss their relationship to the power system observability problem.

2.3.3 *Graph Theory*

A graph G = (N,E) is a structure consisting of a finite set N of *nodes* or *vertices* and a set E of unordered pairs of nodes called *edges* or *branches*. A *directed graph* or *digraph* is similarly defined, except that each edge is an ordered pair, giving it direction from one node to another. Figure 4a shows an example of a graph; figure 4b shows an example of a digraph.

A *path* between nodes x and y (x,y) is a sequence of edges which connects node x to node y. A graph is said to be *connected* if there exists a path between any two nodes in the graph. A *loop, cycle,* or *circuit* is a path (x,x) containing at least one edge, in which no node except x is repeated. Figure 4a contains a loop consisting of the edges {(1 ,2), (2, 5), (5, 1)}. A graph which does not contain any loops is called *acyclic.*

A *tree* is a connected acyclic subgraph. A *forest* (see figure 4c) is an unconnected acyclic subgraph. It follows that each component of a forest is a tree. A *spanning tree* is a tree that contains every node of G. A spanning tree contains

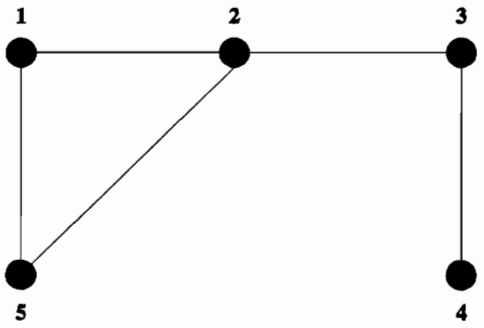
N–1 edges, where N is equal to the number of nodes of the graph (figure 4d).

These concepts can be applied to power system observability analysis as follows. A graph is constructed in which the nodes of the graph correspond to the buses of the network and the edges of the graph correspond to particular *measurement assignments.* A measurement assignment is an association of a measurement with a particular branch in the original one–line diagram of the system. The goal is to develop the largest possible forest in this *measurement assignment graph* (preferably a spanning tree) by assigning the system measurements to a set of edges in accordance with the following measurement assignment rules:

(i) Flow measurements, if assigned, must be assigned to the branch whose flow they measure;

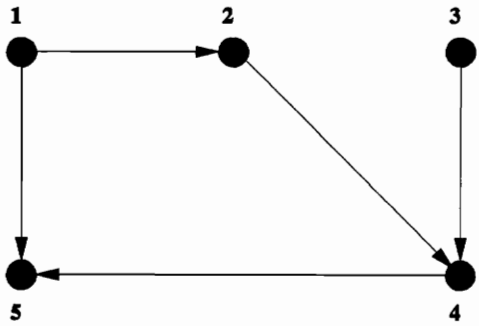
(ii) Injection measurements, if assigned, can only be assigned to one of the branches incident to the bus on which the injection is located.

If a measurement assignment function can be found such that a spanning tree can be formed in the measurement assignment graph, the system is said to be *topologically observable.* A proof showing that a spanning tree in this graph is a necessary and sufficient condition to show system observability is given in [12]. The objective is to determine to which edge each measurement should be assigned such that the maximum–sized forest is achieved. The fact that each injection measurement can be assigned to one of several edges results in a combinatorial problem. One way to solve this problem is to try all possible combinations of injection measurement assignments in an attempt to build a spanning tree. Although such an exhaustive



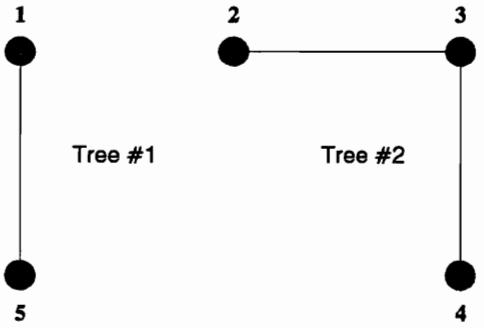
Undirected Graph

(a)



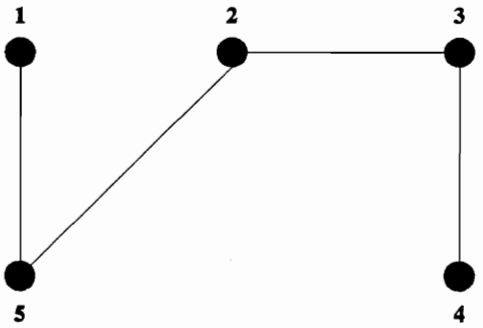
Digraph

(b)



Forest

(c)



Spanning Tree

(d)

Basic Graphs

Figure 4

search is guaranteed to result in the correct determination of system observability, it is too time–consuming for real–time applications. Finding alternative methods of solving the topological observability analysis problem has been the subject of many recent papers.

2.3.4 *Fast Integer Algorithm*

One algorithm suggested to avoid the combinatorial nature of the problem has been proposed by Chen [24]. The confusing nature of his paper has resulted in the perception that his algorithm is based upon the determination of the rank of the Jacobian matrix written in integer form and is, thus, subject to the same problems as the method based upon the Gramian [22]. Chen's algorithm is actually a topologically–based algorithm which attempts to build a spanning tree. Each manipulation of the matrix by Chen's algorithm can be equated to a step in the process of building a tree and the two theorems which Chen uses to increase the speed of his algorithm and proves via a matrix rank analysis are analogous to the graphically proven theorem presented and utilized by Krumpholz et al. [12]. An example will first be used to show the correlation between the matrix manipulations conducted during the first parts of Chen's algorithm and the forming of a tree. Next, the theorems of Chen and Krumpholz et al. will be compared. Finally, Chen's algorithm will be applied to the network of figure 3 to show that it correctly determines the observability of the network and, as a result, cannot be simply based upon the determination of the rank of the Jacobian matrix in its integer form.

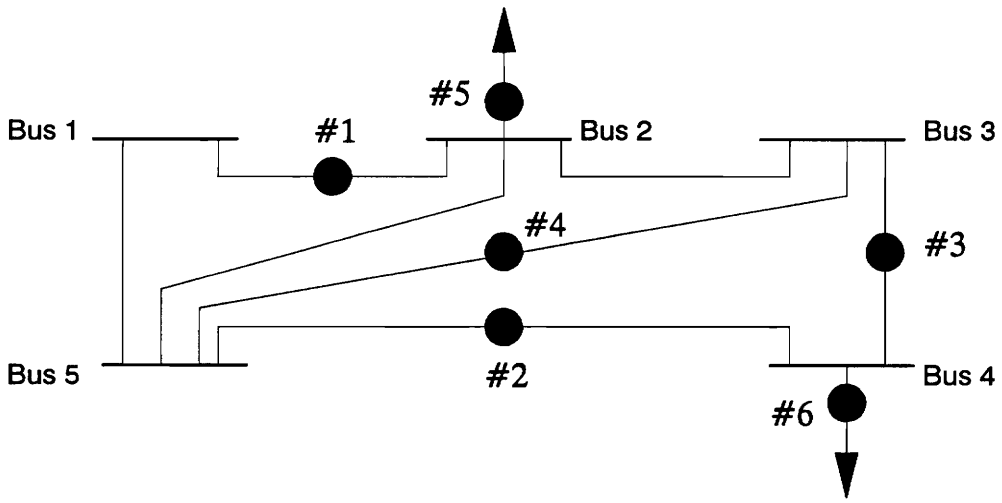
The IEEE 5–bus system with the measurement configuration shown if figure 5 will be used to show the general steps of the algorithm and show the correlation

between the matrix manipulations and the forming of a tree. The first part of Chen's algorithm determines if there are linear dependencies between any of the flow measurements or between any of the flow measurements and any of the injections. This is accomplished as follows. For each row of the Jacobian matrix corresponding to a flow measurement, the two columns containing the non–zero entries ($\pm$ 1) are added and the column being added to the other is deleted, as is the row corresponding to the flow measurement being processed. Viewed topologically, the second node is coalesced with the first node and a supernode is created. Equivalently, the flow measurement is assigned to its respective branch. This procedure is continued until all flow measurements have been processed.

Any zero rows which appear in the Jacobian matrix during this first part of the algorithm correspond to measurements which are linearly dependent with the flows previously processed. For the network in figure 5, one detects two zero rows after processing the 3rd flow measurement as indicated in the sequence below. The rows of the matrices correspond to the measurement numbers and the columns correspond to the node/supernode numbers associated with the network.

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\left[\begin{array}{ccccc}
1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 \\
0 & 0 & 1 & -1 & 0 \\
0 & 0 & 1 & 0 & -1 \\
-1 & 3 & -1 & 0 & -1 \\
0 & 0 & -1 & 2 & -1
\end{array}\right]
\end{array}
$$

Initial Jacobian Matrix



Example Network for Chen Method

Figure 5

$$\begin{array}{c} \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 2 & -1 & 0 & -1 \\ 0 & -1 & 2 & -1 \end{array}\right] \end{array}$$

Matrix after processing flow #1

$$\begin{array}{c} \\ 3 \\ 4 \\ 5 \\ 6 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 3 & 4 \end{array} \\ \left[\begin{array}{ccc} 0 & 1 & -1 \\ 0 & 1 & -1 \\ 2 & -1 & -1 \\ 0 & -1 & 1 \end{array}\right] \end{array}$$

Matrix after processing flow #2

$$\begin{array}{c} \\ 4 \\ 5 \\ 6 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 3 \end{array} \\ \left[\begin{array}{cc} 0 & 0 \\ 2 & -2 \\ 0 & 0 \end{array}\right] \end{array}$$

Matrix after processing flow #3

The zero rows corresponds to measurement #4 and #6 and indicate that (a) flow # 4 is linearly dependent with flows #2 and #3 and, (b) injection #6 is linearly dependent with flow measurements #2 and #3. These dependencies are evident from figure 5. Up to this point the correlation between the matrix manipulations and the tree–building process are readily apparent.

It should be noted that as the first part of the algorithm is conducted, the possible measurement assignments for the injections are updated. This can be seen by observing the changes in the row corresponding to injection measurement #5. Initially, the injection can be assigned to any of its three incident branches leading to nodes 1, 3, and 5. However, after processing flow #1 the injection has only two

possible assignments; to the branch leading to node #3 or to the branch leading to node #5.

The second part of the algorithm is similar to the first. Each row consisting of two non–zero entries is processed by adding the appropriate columns. Topologically, this step assigns injections with only one possible measurement assignment remaining to the appropriate branch. Each such assignment may cause other injections to have only one assignment option, therefore, the process is continued until no more two non–zero rows remain.

The reduced Jacobian matrix that remains after all two non–zero rows have been processed consists of injection measurements each having more than one possible measurement assignment. The next objective is to determine if these injections can be assigned to branches in such a manner that a spanning tree results. Determining this fact is not a matter of examining the rank of the reduced Jacobian matrix as will be shown shortly. Instead, Chen's algorithm proceeds as follows. The reduced Jacobian matrix is first examined to see if it is of a form to which one of two theorems can be applied. These theorems essentially state that (1) a network consisting of N nodes, each containing an injection, is observable, and (2) a network consisting of N nodes and N–1 injections (all located at different nodes) is observable if the injections can be shown to be independent. Chen proves the two theorems summarized above using induction and Gaussian elimination of the Jacobian matrix. It is due to these proofs that it has been perceived that his entire algorithm is based upon the determination of the rank of the Jacobian matrix. The two theorems which Chen presents are easily correlated with the topological

theorems presented by Krumpholz et al.. Chen's first theorem essentially states that a network is observable if each of the nodes not yet included in the tree has an injection measurement associated with it. This is identical to the definition of a critical tree as presented in [12], where a critical tree is defined as a tree which is incident to every every unmeasured node, and an unmeasured node is defined as a node at which there is no injection measurement.

Chen's second theorem can also be viewed from a topological perspective. This theorem essentially states that a system consisting of N nodes and N–1 injections (each associated with a different node) is observable if the injections can be assigned in such manner that the unmeasured node is included in the tree. Chen determines the ability to include the unmeasured node by the properties of the Jacobian matrix. The existence of a square submatrix in the reduced Jacobian matrix is equivalent to a group of nodes, each having an injection measurement associated with it, whose injections cannot be assigned in such a manner that they do not create a loop in the measurement assignment graph. The fact that a loop is formed indicates a dependency between the injections. Therefore, the number of independent injections is equal to (N–1)–1 or N–2. Since a spanning tree requires N–1 branches, it is impossible to build a spanning tree for such a network and the network is not observable. An example illustrating this concept will be shown.

Returning to the current example, the reduced matrix does not satisfy the requirements of either of Chen's theorems, so the algorithm proceeds as follows. The two columns containing the most nonzero entries are added. This step equates to coalescing the two largest observable islands (the two largest trees in the forest)

by assigning an injection measurement(s) to a branch(es) connecting the two islands. After making this measurement assignment(s), some of the remaining injections may have only one possible assignment remaining. This is evidenced by the formation of new two non–zero rows. Such rows are processed as explained earlier. If the result of these steps is the null matrix or a matrix which satisfies the requirements of Chen's theorems, this indicates that all nonredundant measurements have been, or can be, assigned to branches is such a manner that a spanning tree results.

Applying Chen's algorithm to the integer Jacobian matrix associated with the network of figure 3, one obtains the following sequence of matrices and trees.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & -1 & 0 \\ 3 & -1 & -1 & -1 \\ -1 & -1 & 3 & -1 \end{bmatrix} \end{array}$$

Original Jacobian Matrix

$$\begin{array}{c} \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 2 & -1 & -1 \\ 2 & -1 & -1 \end{bmatrix} \end{array}$$

Matrix after processing flow #1

$$\begin{array}{c} \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cc} 1 & 4 \end{array} \\ \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \end{array}$$

Matrix after combining islands containing nodes 1 & 2

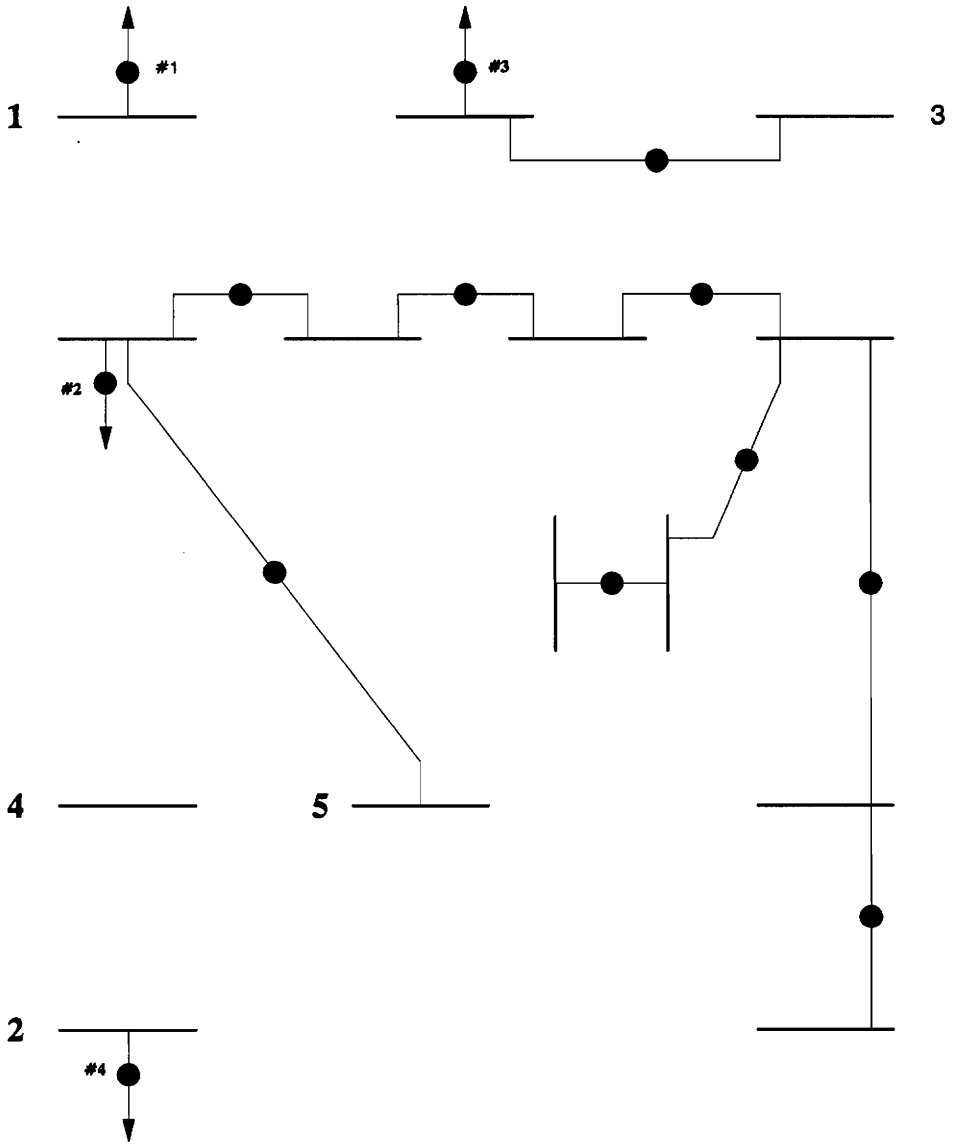
$$\begin{bmatrix} 0 \end{bmatrix}$$

Chen's algorithm concludes that the network is indeed observable. (The null matrix

indicates that all measurements have been assigned to an edge in the measurement assignment graph). This simple example proves that Chen's method is definitely not based upon the determination of the rank of the Jacobian matrix in integer form. It shows that the Chen algorithm does not result in parametric unobservability when the actual network is observable, as stated by Clements. This is because Chen uses the integer form of the Jacobian matrix as a means to keep track of measurement assignments as he attempts to build a tree. He does not simple evaluate the rank of the matrix.

Although Chen's algorithm has been shown to give the correct answer, it has one significant weakness. The weakness relates to the second theorem. The theorem relates the rank of a matrix to the observability of the system, as was previously discussed. To make use of this theorem it must be shown that one cannot find a square submatrix in the reduced Jacobian matrix via row and column permutations. The fact that no such submatrix exists must be proven prior to utilizing the theorem. There is not an efficient way to search for these square submatrices, and as the matrix size increases, the variety of dimensions that the submatrices can assume increases. The IEEE 14–bus system with the measurement configuration shown in figure 6 will be used to show the difficulty in applying theorem two. The reduced measurement Jacobian associated with this network is shown below where the node numbers represent the supernodes formed during the earlier phases of the algorithm.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 2 & 0 & -1 & 0 & -1 \\ -1 & 0 & -1 & 0 & 2 \\ -1 & 0 & 2 & 0 & -1 \\ 0 & 4 & -3 & -1 & 0 \end{bmatrix} \end{array}$$



Measurement Assignment Graph

Figure 6

Rearranging the columns results in

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 5 & 3 & 4 & 2 \end{array} \\ \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & -3 & -1 & 4 \end{bmatrix} \end{array}$$

The rows of the square submatrix of dimension three corresponds to injection measurements 1–3. The rank of this submatrix is equal to two, therefore, the rank of the original Jacobian matrix is deficient by 1 and the system is not observable. Locating this submatrix is the difficult part. Viewed graphically, the submatrix indicates that injections 1–3 can only be assigned in such a manner that a loop is formed. Determining the linear dependence between the injection measurements can be accomplished much more efficiently via graphical algorithms than by searching for submatrices as is recommended by Chen.

The disadvantages of numerically–based algorithms, integer Jacobian rank determination–based algorithms, and Chen's algorithm can be avoided by using strictly graphically–based observability algorithms. The most efficient, and recent, observability analysis algorithm which is based on graph theory is presented in [27]. The algorithm makes use of matroid theory to reformulate the power system observability analysis problem into a maximum cardinality matroid intersection problem. This particular algorithm proves to be ideal for use with the LMS. Since matroid theory is not a common field of study, the concepts and uses of matroids will be summarized.

CHAPTER 3

APPLICATION OF MATROIDS TO OBSERVABILITY ANALYSIS

3.1 Definitions and Principles

3.1.1 *Problem Statement:*

The topological observability problem is one of finding the maximum–sized forest in a graph derived from the one–line diagram of the power system being examined. This graph is called the measurement assignment graph and is composed as follows. The nodes of the graph correspond to the buses of the network. The edges of the graph correspond to *measurement assignments.* A measurement assignment is an association of a measurement with a particular branch in the original one–line diagram of the system. These associations are made in accordance with the following rules:

(i) Flow measurements, if assigned, must be assigned to the branch whose flow they measure.

(ii) Injection measurements, if assigned, can only be assigned to one of the branches incident to the bus on which the injection is located.

The objective is to determine to which edge each measurement should be assigned such that the maximum–sized forest is achieved. In this sense, it is a standard graphical optimization problem. As will be seen, there are certain properties of this problem which can be exploited to avoid a classical combinatorial solution technique.

The concept of matroids dates back to a paper published by Whitney in 1935 entitled "On the Abstract Properties of Linear Dependence". In this paper, Whitney considered algebraic systems which generalize the independence/ dependence properties of the rows/columns of matrices. He called such systems matroids.

Definition: A matroid $M = (E,\zeta)$ is a structure in which E is a finite set of elements and ζ is a family of subsets of E, such that

1) $0 \in \zeta$ and all proper subsets of a set I in ζ are in ζ.

2) If I_p and I_{p+1} are sets in ζ containing p and p+1 elements respectively, then there exists an element $e \in I_{p+1} - I_p$ such that $I_p + e \in \zeta$.

3.1.2 *Examples:*

1) The system $M = (E,\zeta)$ is said to be the *matroid of the matrix* A if E corresponds to the set of columns of A, and ζ contains all of the linearly independent subsets of columns.

2) The system $M = (E,\zeta)$ is said to be the *matroid of the graph* G if E corresponds to the set of edges of G and a subset $I \subseteq E$ is in ζ if and only if I is a loop–free subset of edges.

3) Let π be a partition which separates the finite set E into m disjoint blocks $\{B_1, B_2, ..., B_m\}$ and let d_i, $i = 1, ..., m$ be integers. For any E, π, and d_i, $i = 1, 2, ..., m$, $M = (E, \zeta)$ is a *partition matroid* if

$$\zeta = \{I | I \subseteq E, |I \cap B_i| \leq d_i, i = 1, 2, ..., m\}.$$

As in linear algebra, a maximal independent set is called a *base* of the matroid, and the *rank* r(A) of a subset $A \subseteq E$ is the cardinality of a maximal independent subset of A. The *span* sp(A) of a set $A \subseteq E$ is the maximal superset of A having the same rank of A. A definition drawn from graph theory states that a minimal dependent set is called a *circuit.*

The essence of the matroidal structure is that a *maximal* independent subset is also *maximum.* This structure drastically simplifies the solution of "maximizing" problems since finding the solution simply requires finding *a* maximal independent subset. Finding *a* maximal independent subset is an easy task; elements are added until no further additions are possible. For example, finding a base for a matroid of a matrix simply requires selecting a column and checking it for dependence with the columns previously determined to be linearly independent. If the selected column is independent of the set of independent columns being formed, it is included in the set or basis. If it is determined to be dependent with the set of column vectors which will form the basis, the selected column is not included in the set and another column is selected. This process continues until a basis is formed.

3.2 Greedy Algorithm

An algorithm which illustrates the above idea for graphic matroids is the so–called *greedy algorithm* [34]. This algorithm is used to solve the following problem. Given a matroid $M = (E,\zeta)$ whose elements e_i have been assigned weights

$w(e_i) \geq 0$, find an independent set for which the sum of the element weights is maximal. Graphically, this problem amounts to finding the tree whose edge weights sum to a maximal value. The solution is to sequentially try to include the weightiest edges in the forest (graphic matroid), rejecting only those edges whose inclusion would form a loop with the edges previously included in the forest. The algorithm obtains its name due to the fact that it always attempts to include the weightiest element in the independent set being constructed.

Setting all the element weights equal to unity, $w(e_i) = 1$, allows one to use the greedy algorithm to solve the topological observability problem for a power system containing only flow measurements. Since flow measurements can only be assigned to the branch whose flow they measure, the problem is a classic graphical one of finding a spanning tree. Since all the edge weights are equal, one arbitrarily selects an edge and attempts to include it in the forest. It is rejected only if its inclusion in the forest would form a loop. Once a spanning tree is formed, it is determined that the network is observable and any elements remaining need not be examined. This algorithm can be implemented very efficiently.

3.3 Branching Problem and Matroid Intersections

The greedy algorithm is one example of the power of matroid theory. An enumerative search for an optimal solution was avoided and no backtracking or forward–looking was needed as both depth–first and breadth–first methods require. The algorithm can be employed to efficiently process flow measurements in any power network topological observability test. However, the real difficulty in determining the topological observability of a power system is not presented by the

flow measurements, but by the injection measurements. The multiplicity of their potential measurement assignments accompanied with the constraint that only one of the assignments can be chosen results in a complicated problem. Fortunately, extending the concepts upon which the greedy algorithm is based, one can similarly avoid a combinatorial solution technique to this more complex problem. The greedy algorithm was an efficient method for finding a solution involving a single matroid. The flow and injection measurement observability problem can be solved by finding a solution involving the *intersection of two matroids.* In other words, finding a solution that must meet two independence criteria instead of only one. For power system observability analysis, the two independence criteria are:

> (i) independence among the measurement assignments as determined by the absence of loops in the measurement assignment graph;
>
> (ii) independence in the measurement–to–edge assignment function, i.e., each injection is assigned, at most, to one incident edge.

A graphical optimization problem that is very similar to the power system observability one is the *branching* problem. This problem clearly illustrates the principle behind reformulating the power system observability problem as a matroid–intersection problem. A digraph D = (V,A) as in figure 7a is used to represent the influence structure within a group of people. Each node corresponds to a person, and an edge (x,y) means that x "influences" y. It is desirable to arrange the group structure in such a manner that there is an orderly flow of responsibility and communication. Each person should only be influenced by a single person, except a leader who should not be influenced by anyone. In addition, there should not be any influence paths which short–circuit or bypass other influence paths and noone should be left out of the structure. Such a structure will

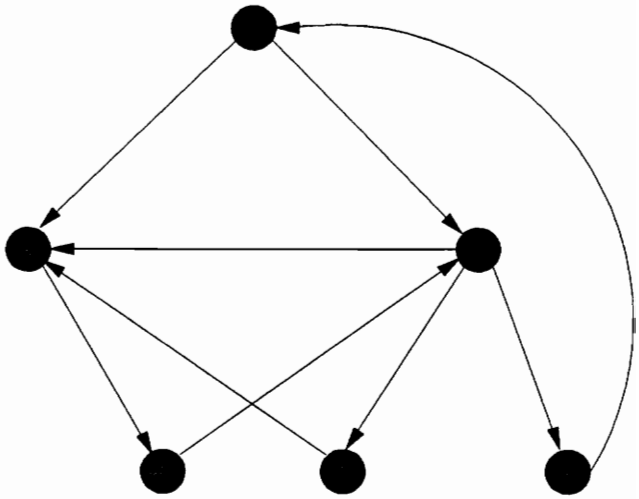
avoid chaotic and conflicting communication flows.

The digraph shows many influence paths. The objective is to organize the people into a branching as described above. This problem can be solved efficiently using the matroid intersection concept. In this case, one of the matroids is the graphic matroid of the digraph D with directions disregarded. The second matroid is the head–partition matroid of the digraph D. To avoid short–circuits, loops must be avoided in the first matroid. To ensure that each person is influenced by only one person, the edges must also be independent in the second matroid. An appropriate branching is depicted in figure 7b by the heavy lines.

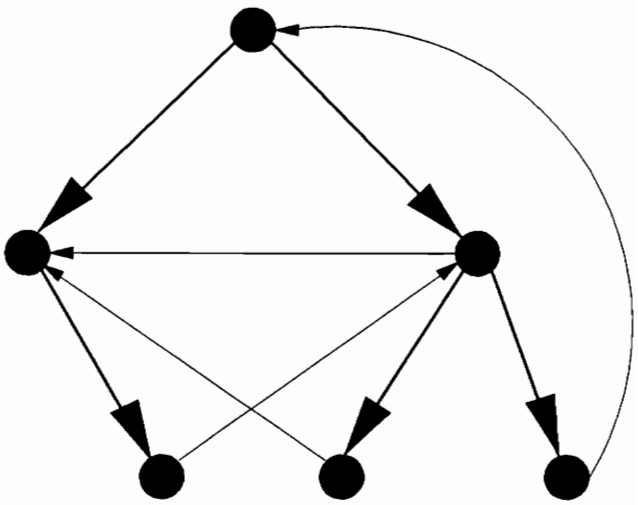
Power system topological observability is almost analogous to the branching problem. Loops must be avoided in the measurement assignment graph – the first matroid. Just as independence had to be maintained in the head–partition matroid in the branching problem, independence must be maintained in the partition matroid associated with the injection measurement assignments (i.e., a node on which an injection is located may only have one branch associated with the injection emanating from it). Finding an efficient algorithm to solve the general intersection of two matroids is the next step.

3.4 Border Graphs and Augmenting Sequences

One method of solving the matroid intersection problem is via *augmenting sequences*. Let I be any intersection of two general matroids, M_1 and M_2. An augmenting sequence is an alternating sequence of element additions and deletions (between M_1 and M_2) which increases the number of elements in the matroid



(a) Influence Digraph



(b) Branching

Branching Problem Graphs

Figure 7

intersection while maintaining independence in both matroids. For example, let $I + e_1$ be independent in M_1. If $I + e_1$ is also independent in M_2, then the addition of e_1 constitutes an augmenting sequence. Element e_1 can be added to I without affecting the independence of either M_1 or M_2. However, if $I + e_1$ is not independent in M_2 then M_2 contains a unique loop or circuit containing e_1. This circuit may be broken by removing one of the elements contained in the circuit, say e_2. Clearly, $I + e_1 - e_2$ is independent in M_1 and M_2, but the addition of e_1 and the deletion of e_2 has not increased the number of elements in the intersection. Since the goal is to form an intersection of maximum size, the sequence cannot stop at this point. Instead, an element e_3 is searched for such that $I + e_1 - e_2 + e_3$ is independent in M_1 and M_2. If not, the search continues as outlined above until an augmenting sequence is located or it is determined that one allowing the addition of e_1 to I does not exist. It should be noted that the addition to I of the 1st, 3rd, 5th, ... (odd) elements preserves independence in M_1, but creates dependence in M_2. The removal of the 2nd, 4th, 6th, ... (even) elements restores independence in M_2. Since the purpose of an augmenting sequence is to increase the size of the intersection set, the sequence must involve an odd number of operations because an even number of operations does not increase the size of the intersection set. A sequence of operations that does not increase the size of the intersecting set is called an *alternating sequence.*

An efficient method of determining the existence and exact nature of augmenting sequences is through the use of *border graphs.* A border graph is a directed bipartite graph whose nodes correspond to the elements whose addition and removal comprise an alternating or augmenting sequence. The edges of the border graph indicate the relationship between the various edge additions and deletions, i.e.

which edges are required to be removed due to the addition of another edge. The result is that a border graph allows one to trace the sequence of element operations that give an augmenting sequence if one exists. An example illustrates the use of border graphs.

Example:

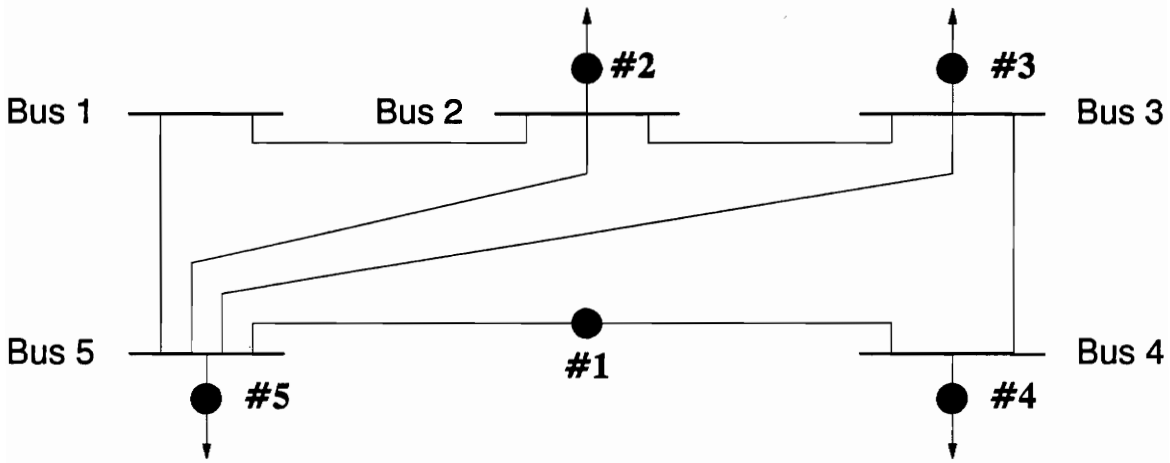
Since the matroid intersection concept is to be used to solve the power system topological observability problem, the example used to illustrate the use of border graphs will be based upon determining the observability of a power network. In addition to showing the use of border graphs, the example will serve as a general case showing the steps involved in building a spanning tree given a network and a set of measurements using the method of augmenting sequences.

The example will make use of the IEEE 5–bus system with the measurement configuration depicted in figure 8. The two matroids in this problem are (1) the measurement assignment graph and (2) the partition matroid associated with the injection assignments. The partition matroid for this example is:

$$\{\{(e_{21}, e_{23}, e_{25})\} ; \{(e_{32}, e_{34}, e_{35})\} ; \{(e_{43}, e_{45})\} ; \{(e_{51}, e_{52}, e_{53}, e_{54})\}\}$$

No more than one element may be chosen from any one set.

An initial measurement assignment graph is shown in figure 9a. The flow measurement is assigned to its respective edge and injection measurements #2 and #3 are arbitrarily assigned to the indicated incident edges. Next injection #4 is



Network

Figure 8

selected and an attempt is made to include it in the measurement assignment graph. Assigning injection #4 to edge e_{43} forms a loop with e_{34}. Assigning the injection to e_{45} forms a loop with e_1. It would appear that injection #4 cannot be included in the measurement graph. However, reassigning injections #2 and #3 to other incident edges will allow injection #4 to be included. The appropriate reassignments (given by an augmenting sequence) are found via a border graph as follows.

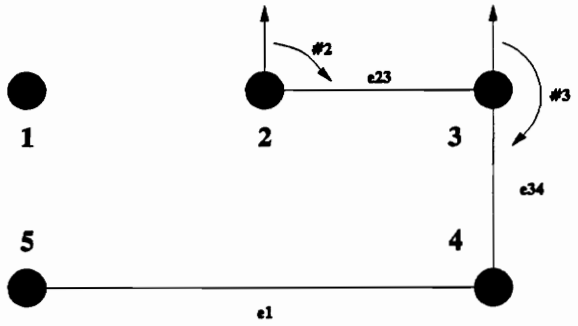
Since the flow edge e_1 cannot be reassigned, injection #4 cannot be assigned to e_{45} without removing e_1. Taking such action will not increase the size of the forest, so nothing will be gained. This exchange of one edge for another is called an alternating sequence, not an augmenting one. Assigning injection #4 to e_{43} will form a loop with e_{34}. This destroys the independence in the first matroid. Independence may be restored by removing e_{34} but this does not increase the size of the forest. However, injection #3 may be reassigned to either e_{32} or e_{35}. These options are indicated in the border graph in figure 10. Assigning injection #3 to e_{35} forms a loop with edges e_1 and e_{43}. This loop may be broken by removing either of the latter two edges. Since e_1 is associated with a flow measurement and cannot be reassigned, removing this edge is not an option. Removing e_{43} is not an option since this was the first edge added in the sequence and removing it means deleting injection #4 from the measurement set; gaining nothing. This fact is evident from the border graph.

Reassigning injection #3 to e_{32} is a valid option since it does not represent a backtrack. This, however, forms a loop involving edges e_{32} and e_{23}. This loop may

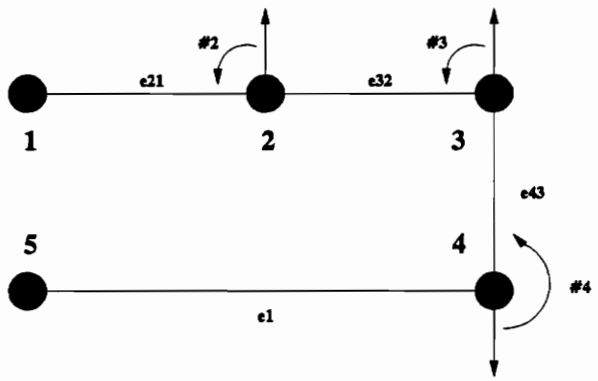
be broken by reassigning injection #2 to e_{21} or e_{25}. Assigning the injection to e_{25} would form a loop requiring the removal of either e_{43} or e_{32}, both of which represent a backtrack. Assignment of injection #2 to e_{21} does not form a loop and the measurement assignment graph shown in figure 9b results. All of the elements of the first matroid, the edges of the measurement assignment graph, are independent. Likewise, each injection has only been assigned to one edge, maintaining independence in the injection measurement partition matroid. As can be seen from the border graph, the number of edge operations involved in the reassignment (augmenting) sequence is odd and the sequence is given by

$$\{e_{43}, e_{34}, e_{32}, e_{23}, e_{21}\}$$

The algorithm explained above concludes that a measurement is redundant if an augmenting sequence cannot be found such that a measurement assignment can be made which maintains independence in the measurement assignment graph and the injection measurement partition matroids. Since power systems are relatively sparse, the border graphs associated with the search for augmenting sequences are not prohibitively large. Each injection has few possible assignments (4 to 7 as indicated in [24]) and the low level of connectivity typical of power systems means that the number of loops capable of being formed by measurement assignments are limited. Simulation shows that the longest augmenting sequence typically encountered for a 118–bus system consists of nine edge operations. The length of the augmenting sequences encountered when using this method with the resampling technique are longer than those encountered in a standard observability analysis algorithm where the flow measurements are processed first. This is because the flow



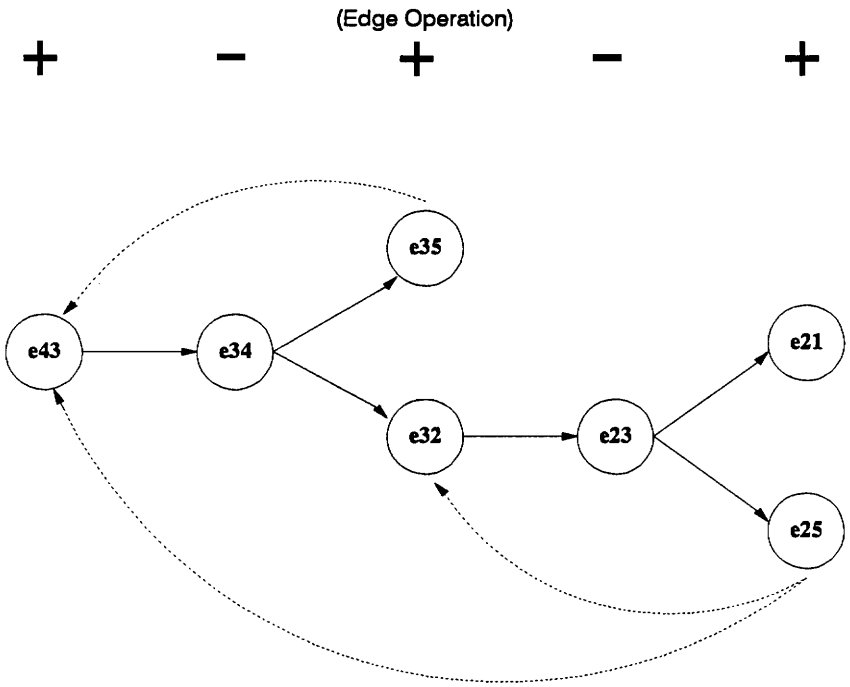
Initial Measurement Assignment Graph



Revised Measurement Assignment Graph

Measurement Assignment Graphs

Figure 9



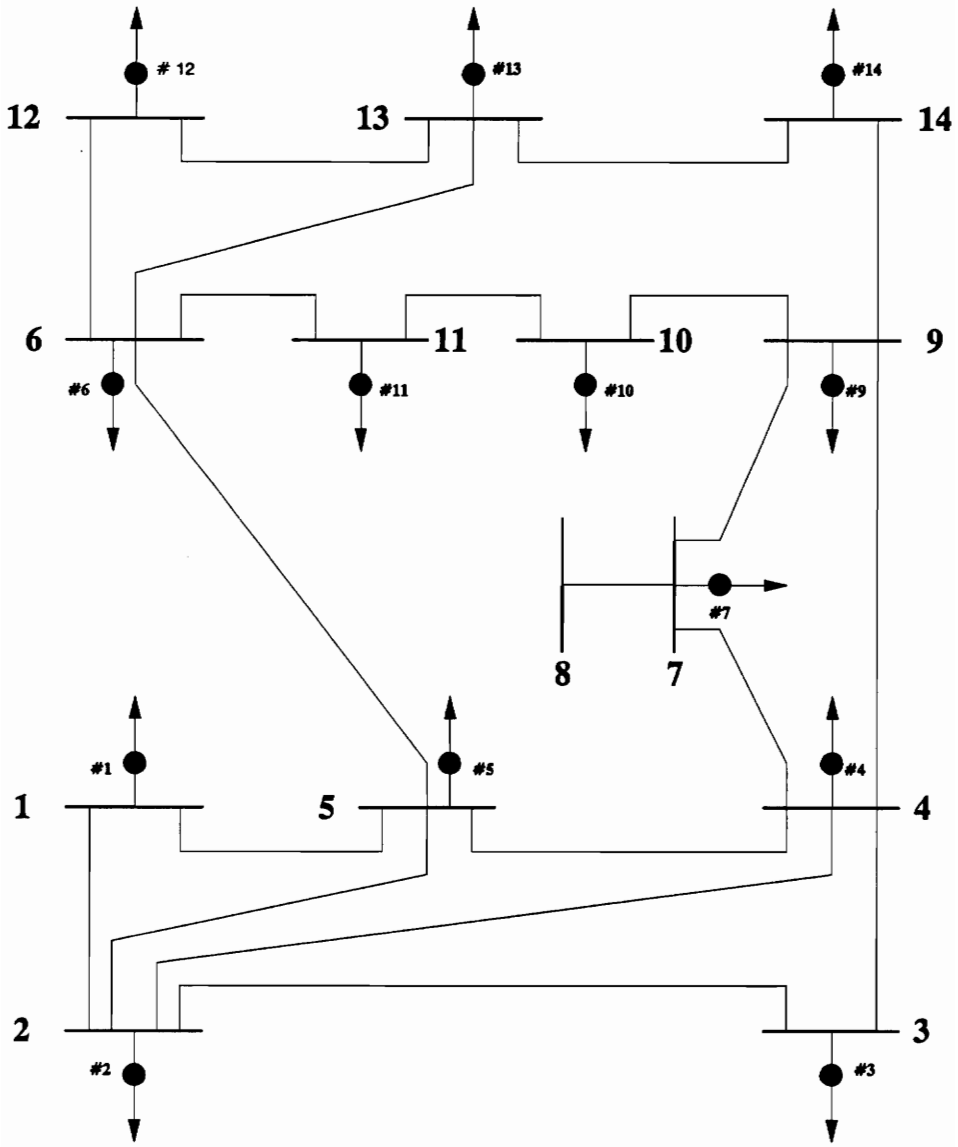
Border Graph

Figure 10

measurements once included in the measurement assignment graph cannot be reassigned and, therefore, cannot appear in an augmenting sequence. Since the resampling technique requires random measurement sets, flow measurements are not processed first. There are several consequences which result from this difference. First, every time an edge associated with a flow measurement is determined to form a loop in the measurement assignment graph and the current measurement set contains at least one injection measurement, a search for an augmenting sequence must be conducted to determine if the flow can be included in the measurement set. Such searches are not necessary if all flow measurements are processed prior to any injection measurements. An additional benefit of processing all flow measurements first results from the fact that the greater the number of flow measurements included in the measurement assignment graph prior to processing any injections means (a) fewer injections will need to be processed to complete the measurement sample and (b) those injections processed have a lower probability of forming loops with other injection edges and, thus, do not cause injection measurement reassignments. Several examples illustrate this fact.

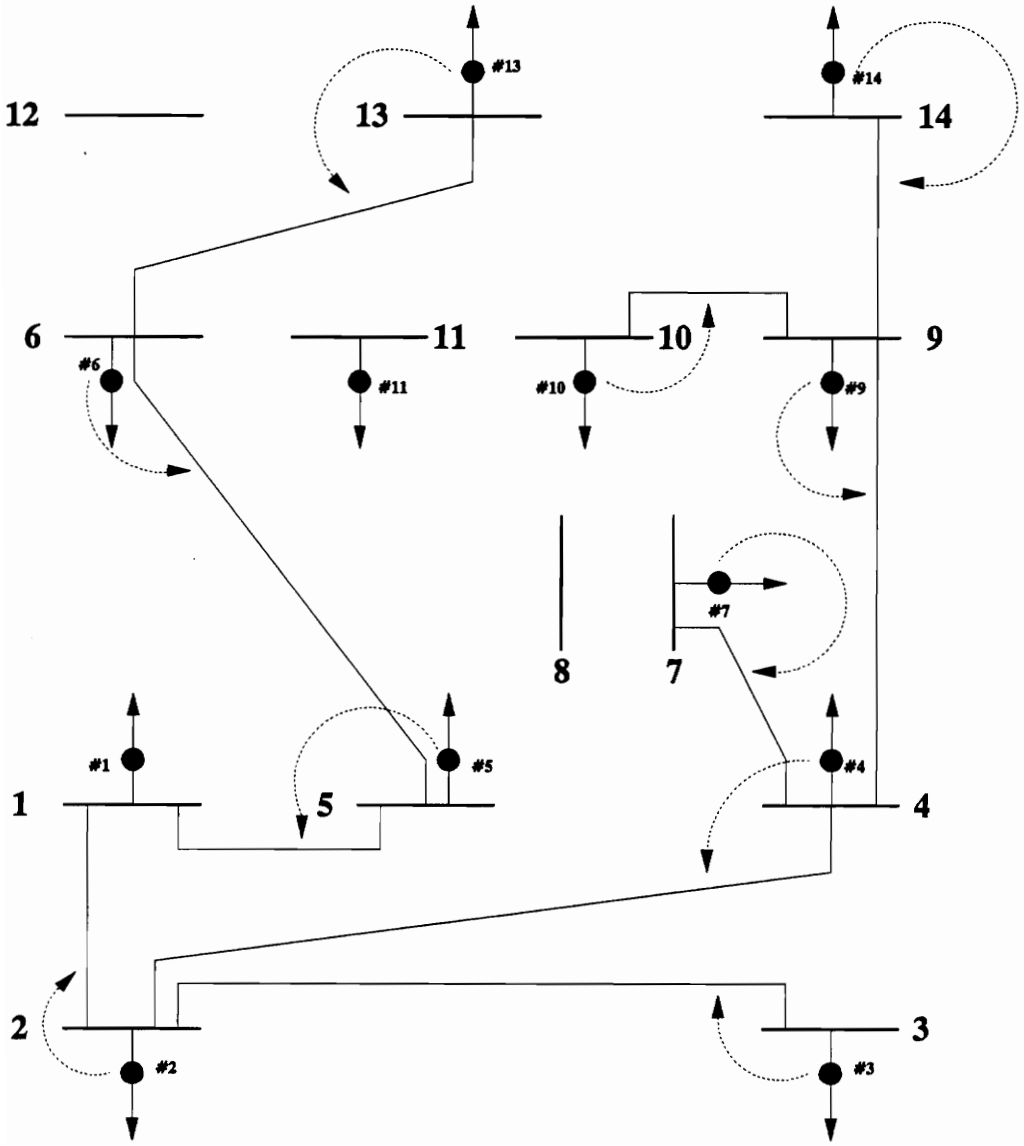
Examples:

The first example consists of the IEEE 14–bus system with only injections as shown in figure 11. The augmenting sequence required to arrive at each successive measurement assignment graph is indicated below the appropriate graph in figures 12–15. In contrast, the IEEE 14–bus system with the measurement configuration shown in figure 16 (containing flow measurements in addition to injections) results in the measurement assignment graph shown in figure 17 without a single augmenting sequence required to obtain a spanning tree. Although this



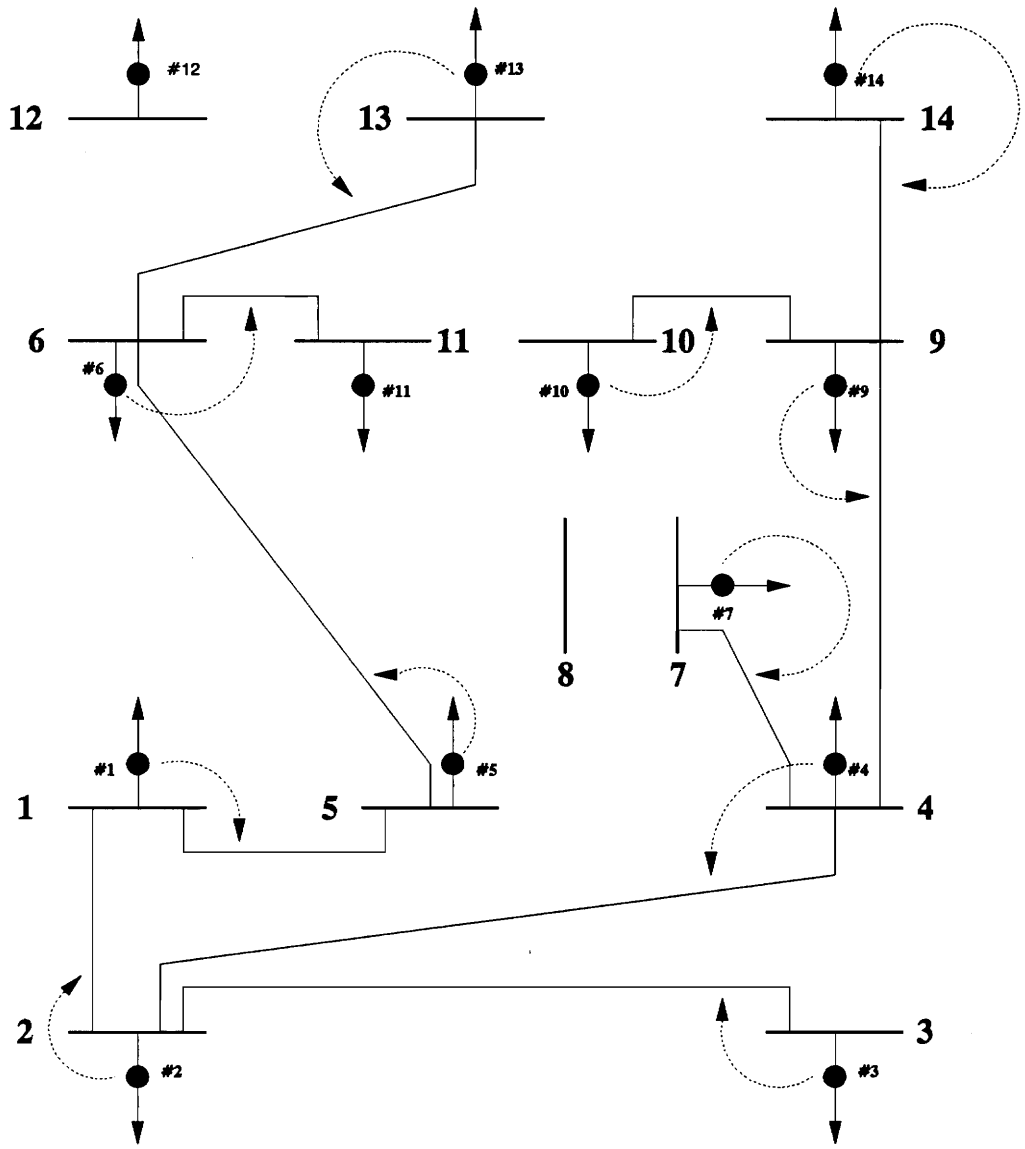
Actual Network

Figure 11



Initial Measurement Assignment Graph

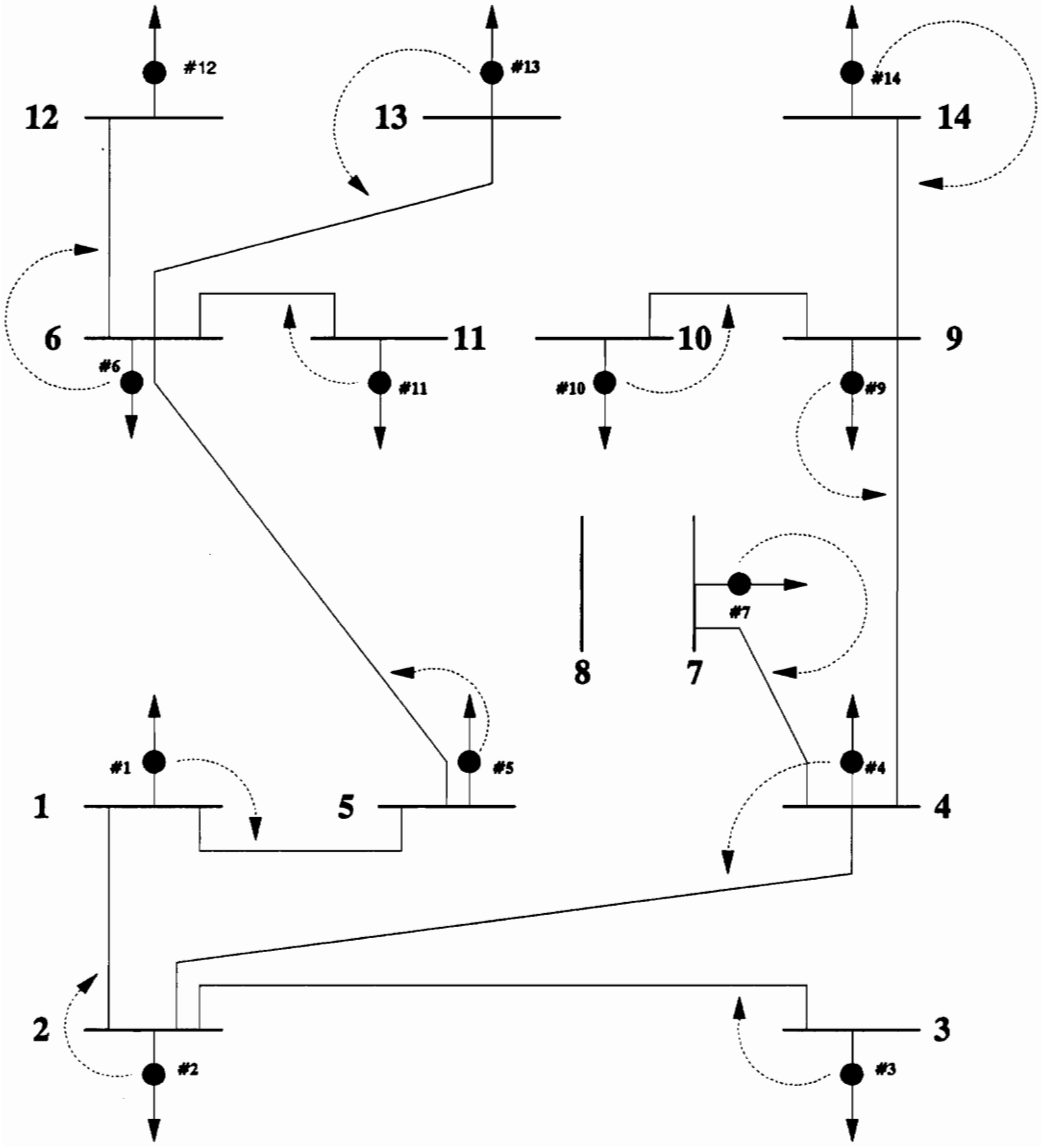
Figure 12



Measurement Assignment Graph
After First Augmenting Sequence

{E:1-5, E:5-1, E:5-6, E:6-5, E:6-11}

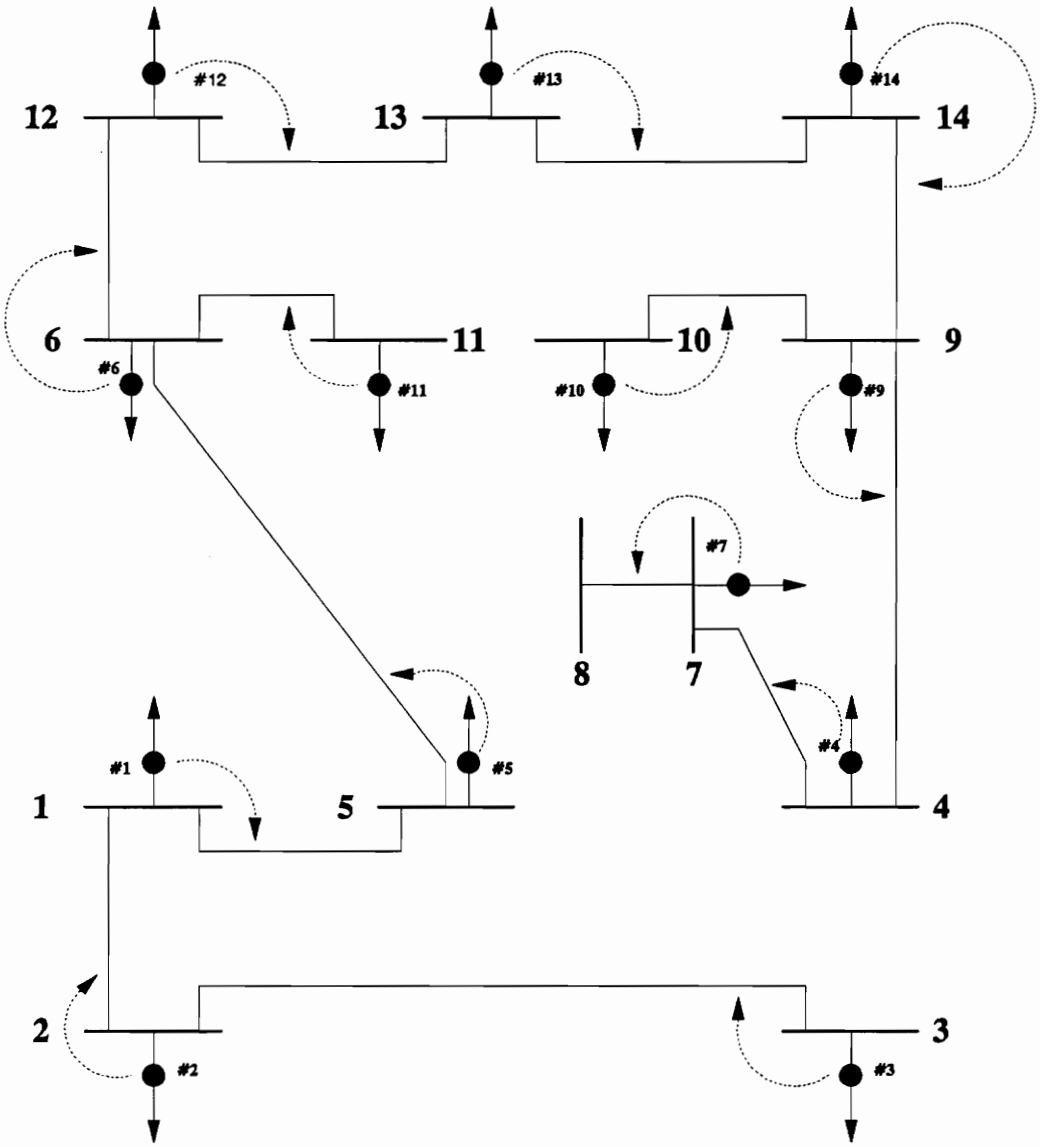
Figure 13



Measurement Assignment Graph
After Second Augmenting Sequence

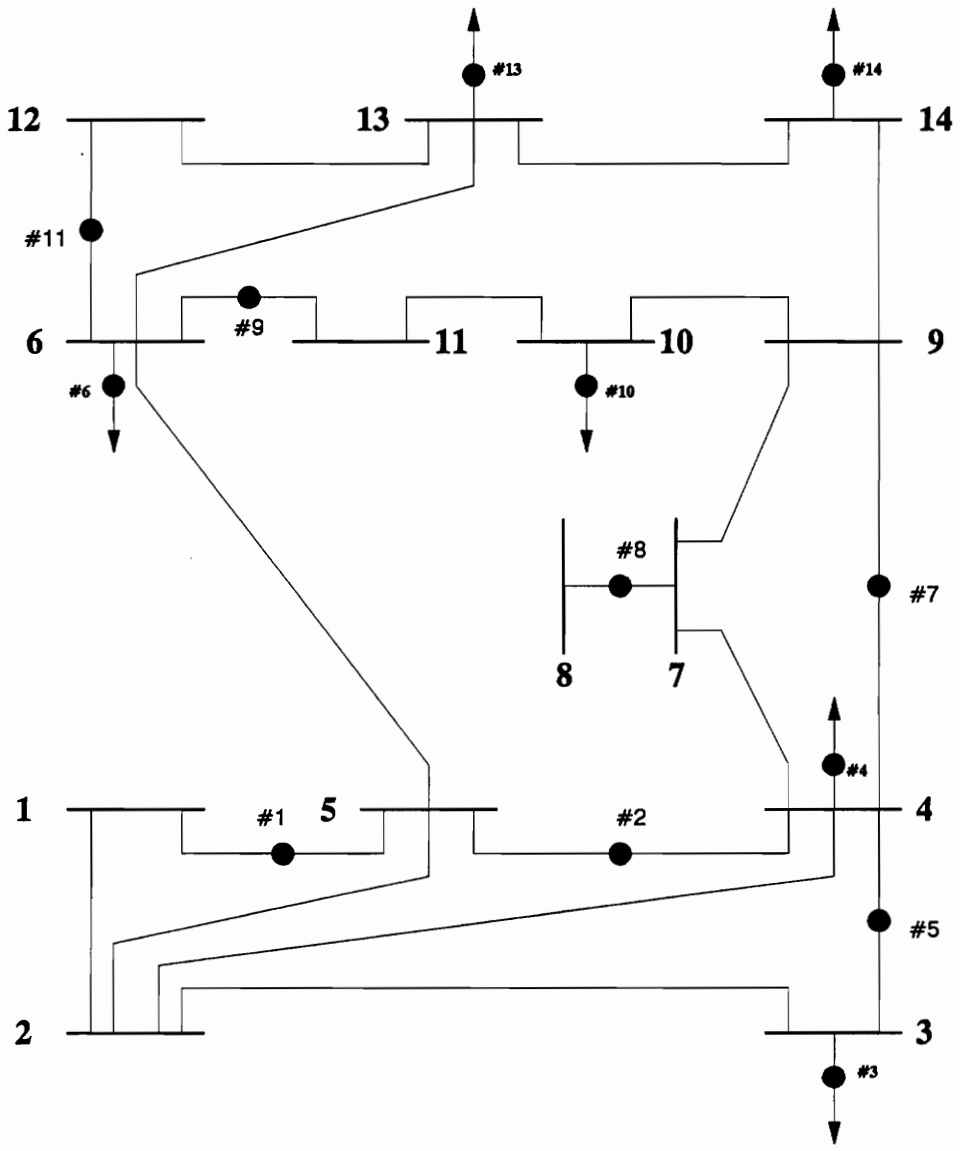
{E:11-6, E:6-11, E:6-12}

Figure 14



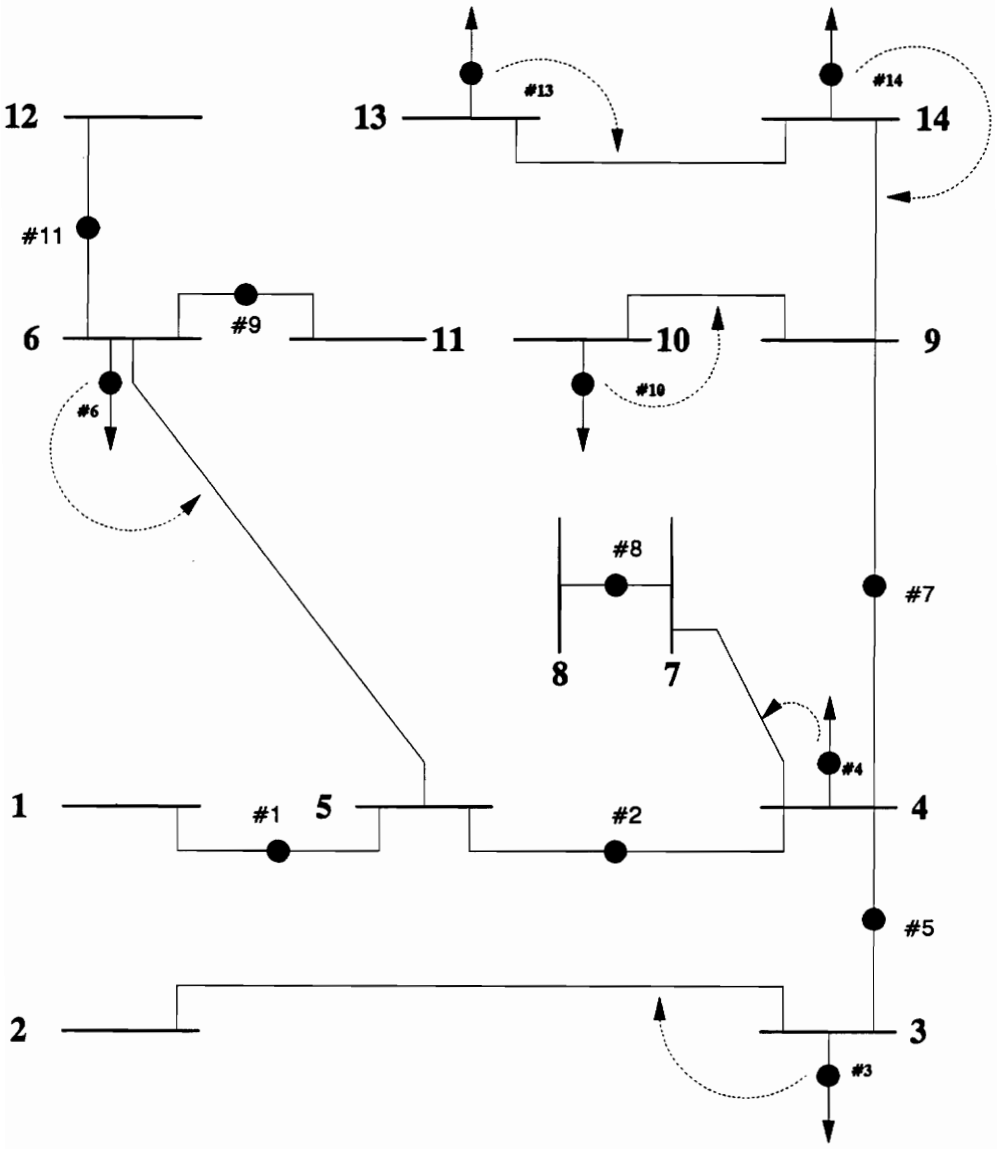
Final
Measurement Assignment Graph
{E:12-13, E:13-6, E:13-14, E:4-2, E:4-7, E:7-4, E:7-8}

Figure 15



Actual Network

Figure 16



Final Measurement Assignment Graph

Figure 17

measurement configuration was intentionally chosen such that augmenting sequences were not required, the example illustrates the advantages of processing flow measurements prior to processing any injection measurements.

CHAPTER 4

IMPLEMENTATION OF THE RESAMPLING TECHNIQUE

4.1 Optimizing Efficiency of the Algorithm

The matroid intersection observability algorithm serves as the basis for the algorithm used to generate the measurement samples required by the resampling technique. This matroid–based test for system observability, based on a general, theoretically supported algorithm, is fast and free of arithmetic errors while sparse storage techniques are easily employed.

The matroid intersection algorithm as described in [27] is used to analyze the observability of a system. In [27], flow measurements are processed prior to injection measurements. This procedure has several advantages. Most importantly, the number of injection measurements that must be processed in order to complete a spanning tree is minimized. Ideally, a spanning tree is formed with flow measurements alone; the fewer the number of injection measurements that are processed, the fewer the required number of augmenting sequence searches. Additionally, the length of the augmenting sequences is shorter for a measurement assignment graph containing many flow measurements since there is less likelihood of a branch associated with an injection forming a loop with a branch associated with another injection. Both of these facts make it advantageous to process flow measurements first.

The matroid intersection algorithm has been modified to randomly generate spanning trees and, hence, measurement samples which make the system observable.

Generating measurement samples for use with the LMS estimator does not allow the processing of flow measurements prior to processing injections. Utilizing this technique would result in few injections appearing in the generated measurement sets. Since the objective is to produce an uncontaminated measurement sample, one cannot favor the inclusion of flow measurements over injections since the flow measurements being favored may be bad. Not being able to first process the flow measurements necessarily increases the computation time of the matroid intersection algorithm. Once an injection is included in the measurement set being generated, an augmenting sequence search must be conducted for each flow measurement which forms a loop in the measurement assignment graph. Such searches are not required if all flow measurements are processed first.

As mentioned above, the manner in which measurements are selected for possible inclusion in the measurement set being generated is critical. No measurement should be favored over another. One way to select measurements in order to avoid this problem is to randomly choose each measurement from the set consisting of all system measurements. As a measurement is selected, it is checked for redundancy with the measurements already included in the measurement sample being generated. If the most recently selected measurement is determined to be independent, it is included in the sample; if the selected measurement is determined to be dependent, it is rejected and another measurement is randomly selected. This process continues until a spanning tree is formed or until it is determined that the system measurement configuration does not make the system observable. This procedure, although correct, is not efficient. The random selection of measurements as described above can result in a high percentage of the selected measurements

being redundant. Since each of the redundancy checks requires the search for an augmenting sequence (once an injection is included in the measurement set), unnecessary computational time is spent determining measurement redundancy. Reducing the probability of selecting redundant measurements can minimize these time–consuming computations.

The method used to reduce the probability of selecting redundant measurements entails grouping the measurements into their *fundamental sets.* A fundamental set is defined as a set associated with a state variable which contains those measurements which have nonzero terms in the corresponding column of the Jacobian submatrices $H_{P\theta}$ and H_{QV}. For example, the network shown in figure 5 has the integer Jacobian submatrix $H_{P\theta}$ and the fundamental sets shown below.

$$H_{P\theta} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 \end{bmatrix}$$

Fundamental Sets

1	2	3	4	5
1	1	3	2	2
5	5	4	3	4
		5	6	5
		6		6

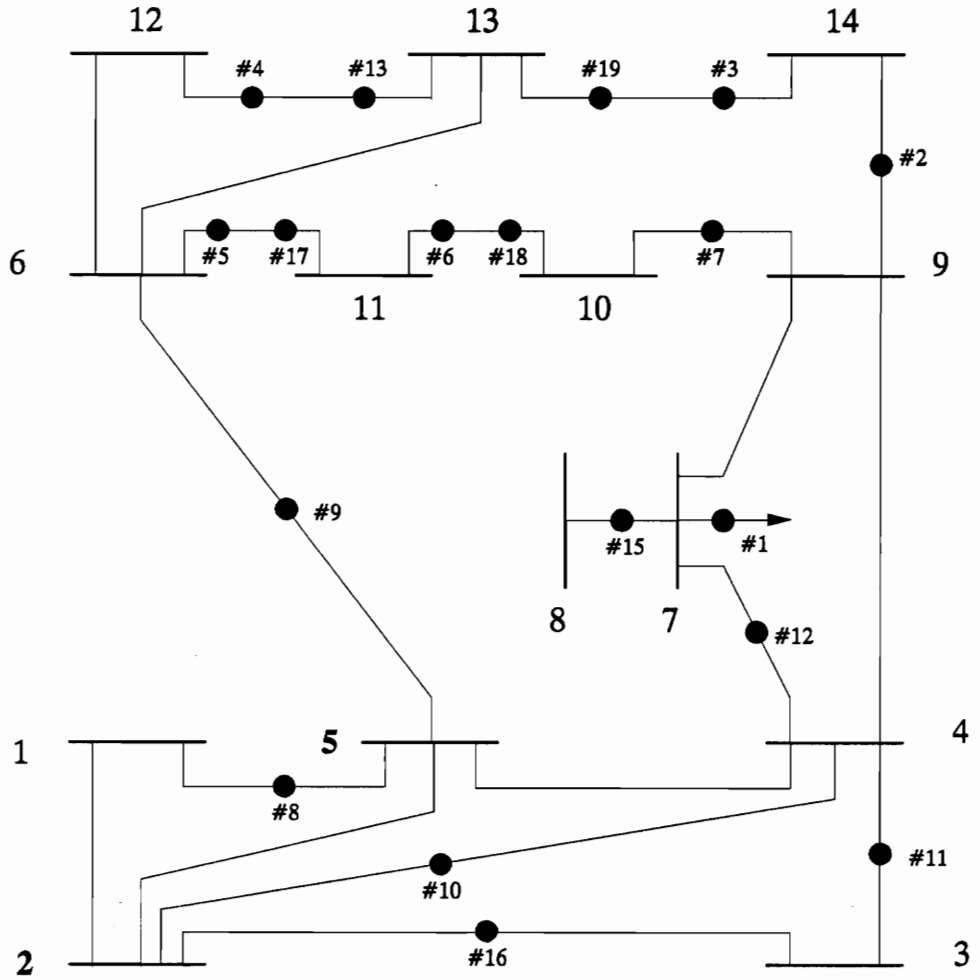
Utilizing fundamental sets, one can select measurements in the following manner. Each fundamental set is sequentially processed. Once a measurement from the fundamental set being processed is included in the measurement sample, one proceeds to the next fundamental set. The remainder of the measurements in the processed fundamental set are essentially removed from the group from which measurements can be chosen (unless they appear in other fundamental sets which have not been processed). An illustration of the resultant reduction in redundant

measurements encountered while building a spanning tree is incorporated in an example problem associated with figures 18–20.

The use of fundamental sets increases the efficiency of the measurement set generation algorithm, but also presents two problems: (1) there is no guarantee that a spanning tree will result and (2) injection measurements appear in a high percentage of the final measurement sets. Indeed, selecting measurements from fundamental sets does not guarantee that a spanning tree will result. A node associated with a fundamental set from which a measurement is included in the sample being generated may not be initially included in or may be subsequently removed from the measurement assignment graph via an augmenting sequence. The fundamental sets associated with such nodes must be designated as sets which must be processed again. If after all fundamental sets have been processed once, a spanning tree has not been formed, the appropriate fundamental sets are processed again. The additional processing of a fundamental set may change the measurement assignment graph such that the fundamental sets requiring reprocessing may change with each iteration.

Example:

An example illustrating the inability to build a spanning tree through the simple use of fundamental sets and the use of the revisiting scheme is illustrated using the network shown in figure 18 along with its associated fundamental sets. The fundamental sets are placed in order of increasing length to minimize the probability of not being able to select a non–redundant measurement from a set. The circled measurement numbers indicate the measurement selected from each



Network

Fundamental Sets

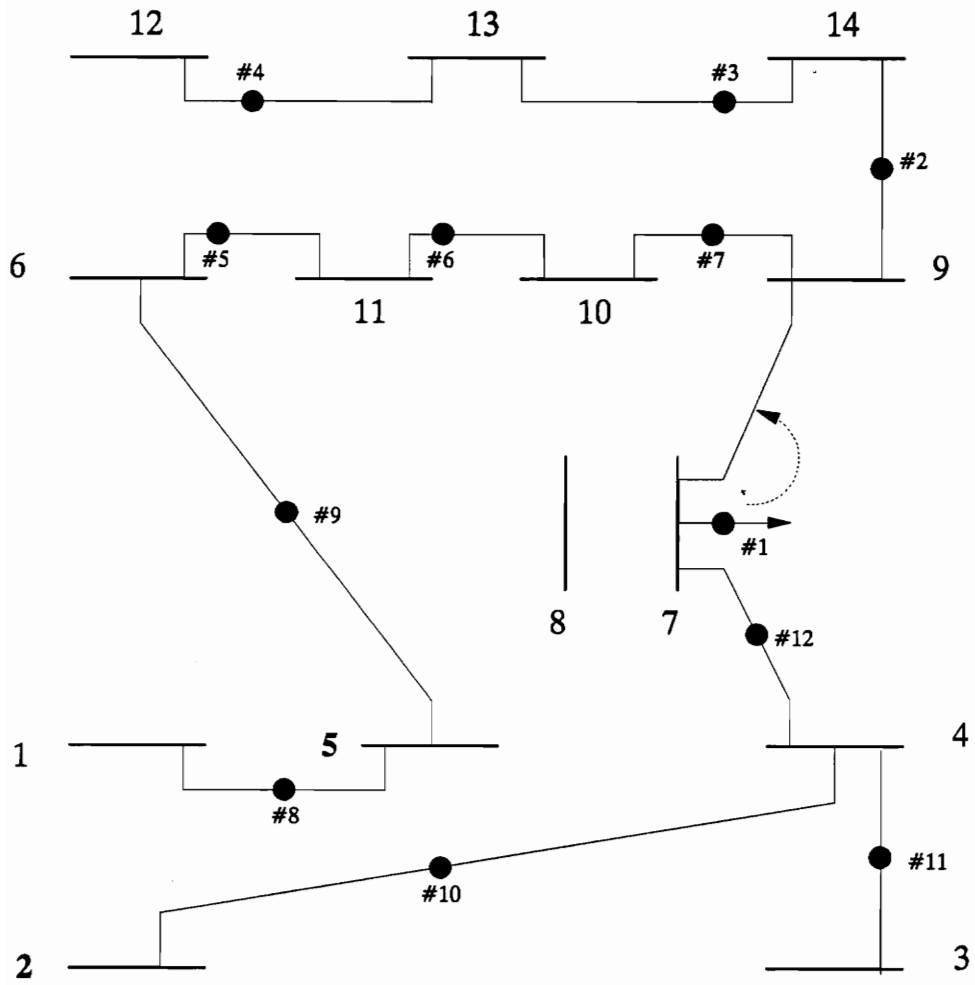
1	2	4	3	5	8	7	9	10	12	14	6	11	13
⑧	⑩	⑪	11	8	①	1	1	6	④	2	⑤	5	3
	16	12	16	⑨	15	⑫	②	⑦	13	③	9	⑥	4
						15	7	18	14	19	14	17	13
											17	18	19

Figure 18

fundamental set. Fundamental sets 3 and 13 do not contain any circled measurements since all of the measurements in these sets are determined to be redundant with those included in the measurement set generated prior to processing these particular fundamental sets. Note that injection measurement #1 is selected from the fundamental set associated with node 8 but is assigned to the branch connecting nodes 7 and 9. Processing all of the fundamental sets results in the measurement assignment graph shown in figure 19. The graph requires another branch incident to node 8 to complete a spanning tree. By noting that node 8 was not included in the measurement assignment graph when a measurement was selected from its associated fundamental set one can go directly to this fundamental set and select measurement #15 to complete the spanning tree as depicted in figure 20. This method is more efficient than attempting to complete the spanning tree by processing all of the measurements which have not yet been determined to be redundant. In this example, one would have a set consisting of measurements #14 to 18 from which to choose the measurement which completes the spanning tree. As system size increases and topologies become more complex, the computational savings experienced using this revisiting technique also increase.

4.2 Maximizing the Selection of an Uncontaminated Sample

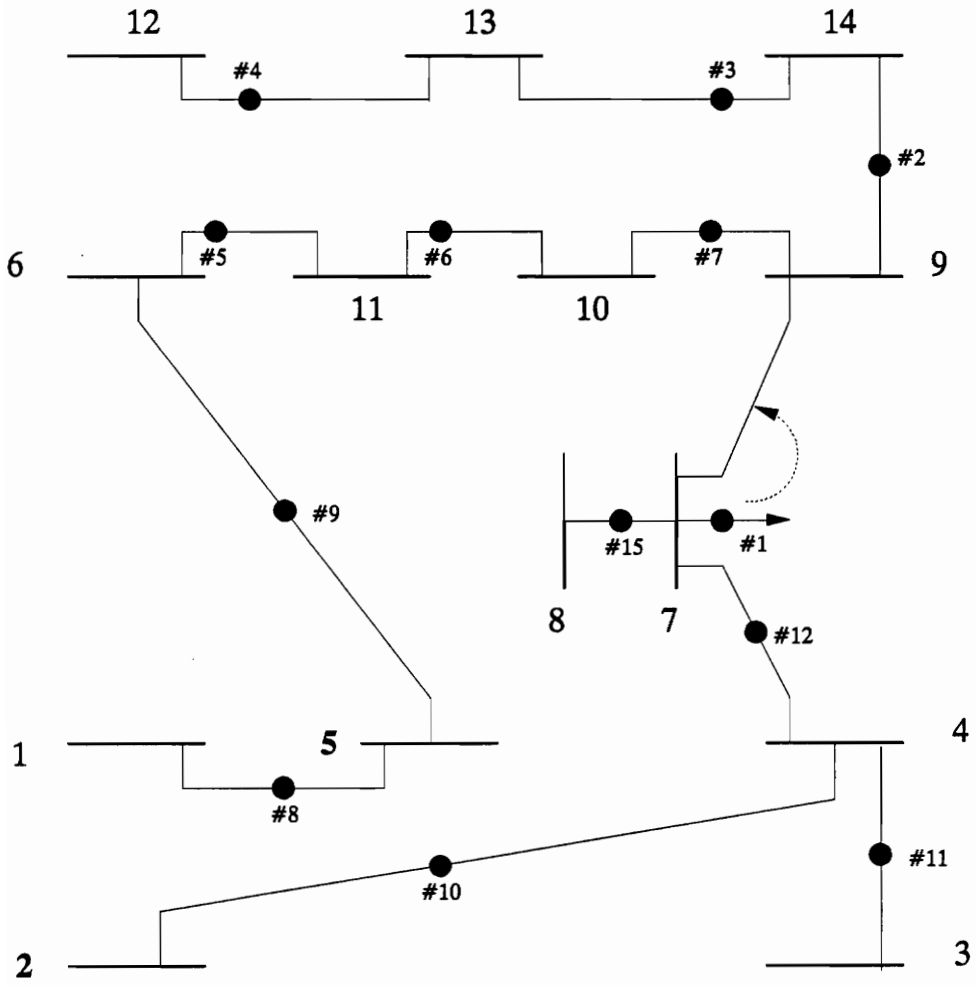
The second problem encountered when making use of fundamental sets to reduce the number of redundant measurements encountered in the tree–building process is that injection measurements appear in a high percentage of the final measurement sets. This problem is due to the fact that injections appear in all of the fundamental sets associated with the nodes incident to the node on which the injection is located. Thus, randomly selecting a measurement from a group of



Measurement Assignment Graph

After Processing Fundamental Sets

Figure 19

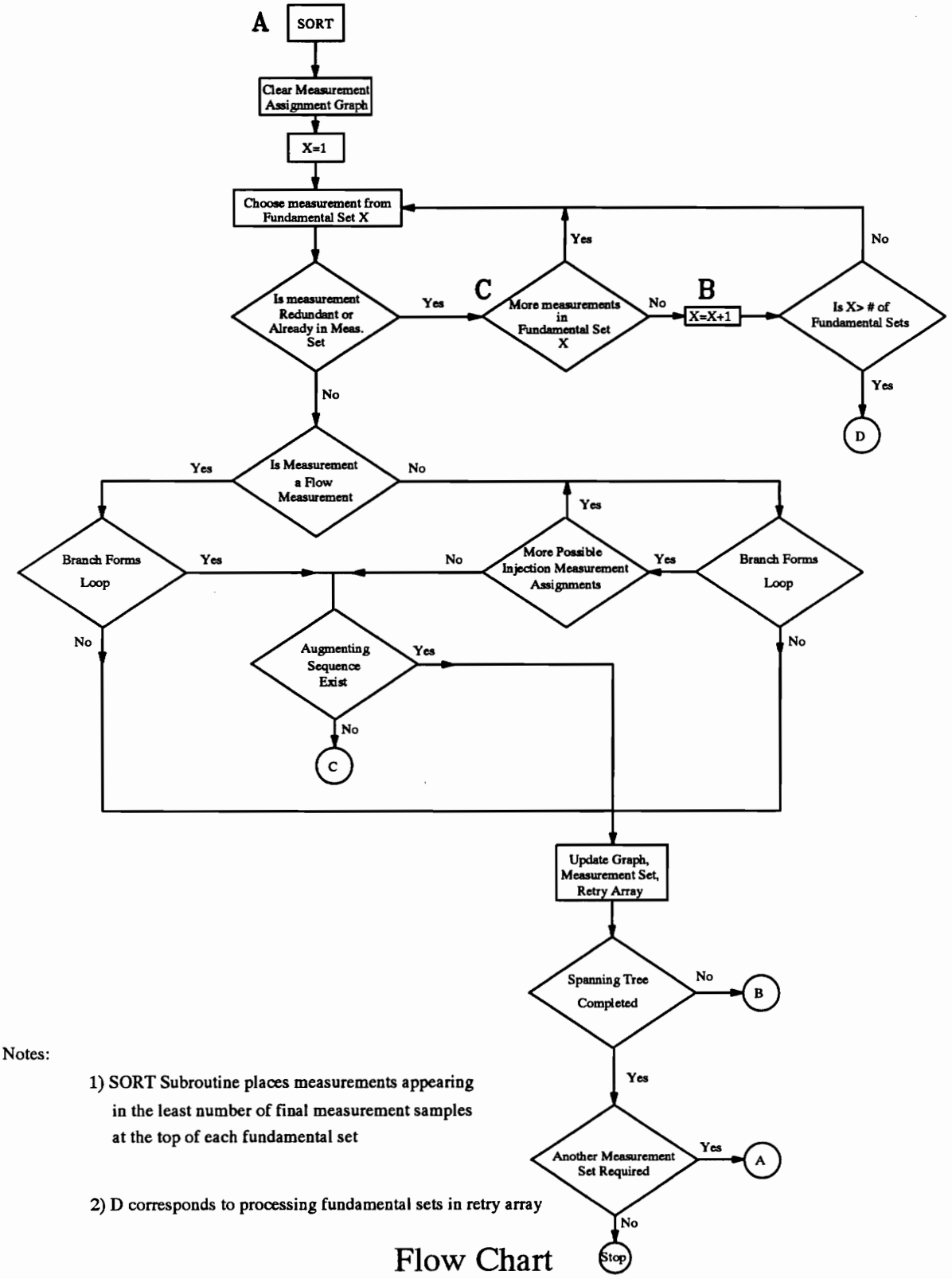


Final
Measurement Assignment Graph

Figure 20

fundamental sets each containing the same injection measurement results in a higher probability of selecting the injection than drawing the other measurements. This point was made in the introduction and is illustrated in figure 2 which is associated with the network in figure 1. Including injection measurements in a preponderance of the generated measurement samples does not maximize the probability of obtaining an uncontaminated measurement sample if some of the injections are bad measurements. The solution to the problem of favoring injection measurements over flow measurements is simple and straightforward. A counter assigned to each measurement keeps track of the number of times its associated measurement appears in a drawn measurement sample. Prior to each sample generating run, the measurements which have been included in the smallest number of samples generated thus far (the lowest counter values) are placed at the top of their associated fundamental set(s) using a simple sorting algorithm (subroutine SORT in the attached flow chart). These measurements, therefore, have the highest probability of appearing in the next sample generated. The result of this simple technique is to equalize, under the observability constraints, the probability of each measurement appearing in a final measurement sample. The extent to which the measurement distribution in the samples can be equalized is dependent upon the system topology and measurement configuration. A critical measurement will necessarily appear in all of the samples. For example, a node which is connected to the system by a single line which has only one measurement (say a flow measurement on the line) associated with that line can only be included in a spanning tree if that measurement appears in the measurement sample defining the tree. Adding another flow measurement to the line will result in each of these measurements appearing in half of the generated samples. The addition of other

measurements to this area of the network, thereby increasing the size of the fundamental set associated with the local state variables, will result in a proportional drop in the number of times each of the measurements appears in a sample. An extension of this reasoning concludes that those measurements contained in relatively small fundamental sets will be selected more often in the drawn samples than those measurements which are located in relatively large fundamental sets. In terms of maximizing the probability of obtaining an uncontaminated sample, it follows that it is desirable to equalize the sizes of the fundamental sets. This will allow the most uniform measurement distribution within the samples and, as a result, decrease the probability that one or several bad measurements will contaminate all of the samples. A flow chart outlining the overall sample generating algorithm is presented in figure 21. Each of the sections of the flowchart have been discussed in previous sections.



Notes:

1) SORT Subroutine places measurements appearing in the least number of final measurement samples at the top of each fundamental set

2) D corresponds to processing fundamental sets in retry array

Flow Chart
Figure 21

4.3 Simulation Results

The matroid–based observability analysis algorithm coupled with the necessary condition of observability of including a measurement from each of the fundamental sets and the tree completion scheme outlined above results in a fast and correct method of generating the measurement samples required by the resampling technique and, ultimately, the LMS estimator. The results listed on the following page indicate the improvement in computing time using this method versus the method currently in use. All times are in seconds unless specified otherwise. For a 14–bus system it takes 10.06 seconds (on a VAX 3200) to generate 100 samples using the old method whereas it takes only 1.11 seconds using the new method. For the 118–bus system, the old method does not succeed in generating 10 samples which make the system observable, whereas the new method succeeds in 11.04 seconds. In addition to being faster, the program implemented achieves a much more uniform measurement distribution within the measurement samples. Comparing figure 22 to figure 2 indicates the improvement in measurement selection uniformity. Figure 22 indicates that the number of times a measurement appears in 100 samples ranges from 35 to 40. This compares to a range of 16 to 76 as indicated in figure 2. The reason the measurement distribution resulting from the new method is not perfectly uniform was discussed in the previous section.

COMPUTATIONAL RESULTS

14–Bus System

Generation of 100 Measurement Sets

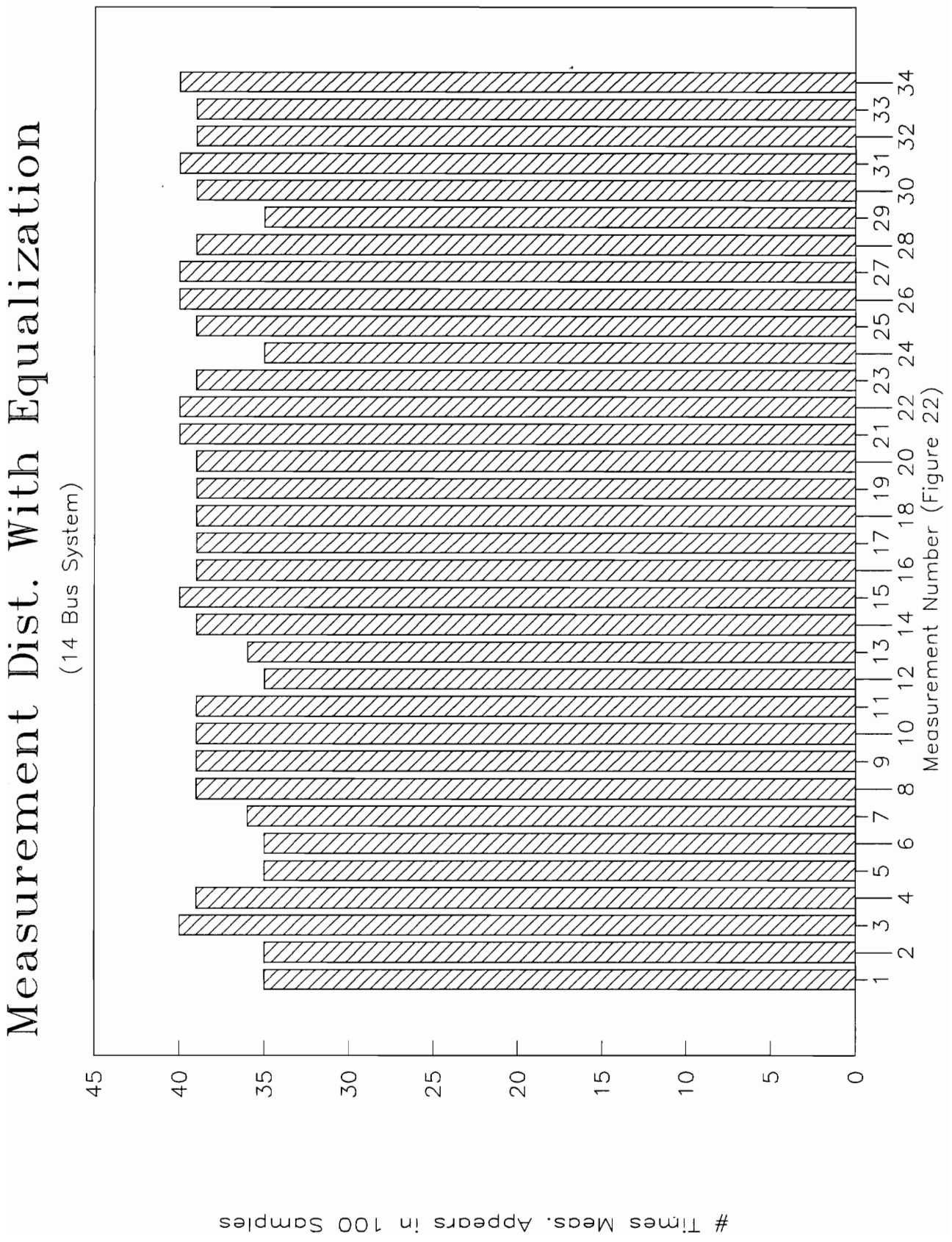
Method	CPU Time (secs)
Old Method	10.06
New Method	1.11

118–Bus System

Generation of 10 Measurement Sets

Method	CPU Time (secs)
Old Method	Days
New Method	11.04

Note: All CPU times based on VAX 3200 system.


Measurement Dist. With Equalization
(14 Bus System)
Times Meas. Appears in 100 Samples
0
5
10
15
20
25
30
35
40
45
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
Measurement Number (Figure 22)

CHAPTER 5

CONCLUSION

The topological observability analysis method based on the principle of maximum cardinality matroid intersections is a fast algorithm whose mathematical basis ensures correct results. The method is applicable to large systems since the length of the augmenting sequences encountered is relatively independent of system size.

A new method, which makes use of fundamental sets and an additional measurement selection scheme has been developed. It results in a dramatic improvement in computing times required to generate the measurement samples ultimately used by the LMS estimator. The proposed measurement selection algorithm gives a greater probability of obtaining an uncontaminated measurement set. This will not only decrease the computing time spent generating the measurement samples but will, more significantly, result in computational savings due to a reduction in the number of times the Newton–Raphson algorithm must be executed. Together, these techniques result in a program that brings the LMS estimator one step closer to real–time application in a control center.

REFERENCES

[1] L. Mili, V. Phaniraj, P.J. Rousseeuw, "Least Median of Squares Estimation in Power Systems", *IEEE Trans. Power Syst.* Vol. 6, No. 2, May 1991, pp. 511–523

[2] D.M. Falcao, S.M. de Assis, "Linear Programming State Estimation: Error Analysis and Gross Error Identification", *IEEE Trans. Power* Syst., Vol. 5, No. 5, August 1988, pp. 809–815

[3] Ali Abur, Mehmet K. Celik, "A Fast Algorithm for the Weighted Least Absolute Value State Estimation", *IEEE Trans. Power Syst.*, Vol. 6, No. 2, February 1991

[4] W.W. Kotiuga, "Development of a Least Absolute Value Power System Tracking State Estimator", *IEEE Trans. Power Appar. & Syst.*, Vol PAS–104, No. 5, May 1985, pp. 1160–1166

[5] W.W. Kotiuga, M. Vidyasagar, "Bad Data Rejection Properties of Weighted Least Absolute Value Techniques Applied to Static State Estimation", *IEEE Trans. Power Appar. & Syst.*, Vol. PAS–101, No. 4, April 1982

[6] Y. Hamam, J. Trecat, "Advanced Linear Programming Techniques Applied to Power System State Estimation and Bad Data Identification", 7th PSCC Conference, Lausanne, 12–17 July, 1981

[7] M.R. Irving, R.C. Owen, M. Sterling, "Power System State Estimation Using Linear Programming", *Proceedings of the IEE,* Vol. 125, No. 9, Sept. 1978, pp. 879–885

[8] P.J. Rousseeuw, A.M. Leroy, *Robust Regression and Outlier Detection,*

John Wiley, 1987

[9] K.A. Clements and B.F. Wollenberg, "An Algorithm for Observability Determination in Power System State Estimation", *IEEE PES Summer Meeting,* Paper A 75 447–3, San Francisco, July 1975

[10] J.S. Hortan, R.D. Masiello, "On–line Decoupled Observability Processing", *Proc. of PICA,* pp. 420–426, Toronto, May 1977

[11] J.J. Allemong, "An Examination of Solvability for State Estimation Algorithms", *IEEE PES Winter Meeting,* Paper a 80 008–3, New York, Feb. 1980

[12] G.R. Krumpholtz, K.A. Clements, P.W. Davis, "Power System Observability: A Practical Algorithm Using Network Topology", *IEEE Trans. Power Appar. & Syst.*, Vol. PAS–99, No. 4, July/August 1980

[13] K.A. Clements, et al., "Power System State Estimation Residual Analysis: An Algorithm Using Network Topology", *IEEE Trans. Power Appar. & Syst.*, Vol. PAS–100, No. 4, pp. 1779–1787, April 1981

[14] K.A. Clements, et al., "State Estimator Measurement System Reliability Evaluation — An Efficient Algorithm Based on Topological Observability Theory," *IEEE Trans. Power Appar. & Syst.*, Vol. PAS–101, No. 4, pp. 997–1004, April 1982

[15] K.A. Clements, et al., "Power System State Estimation with Measurement Deficiency: An Algorithm That Determines the Maximal Observable Subnetwork", *IEEE Trans. Power Appar. & Syst.*, Vol. PAS–101, No. 9, pp. 3044–3052, Sept. 1982

[16] Th. Van Custem, P.J. Gailly, "A Simple Algorithm for Power System Observability Analysis and Related Functions", *Proc. IFAC–CEGRE*

Symp. Control Applic. Power Syst. Secur., Paper 101–05, Florence, Italy, Sep. 1983

[17] A. Bargiela, et al., "Observability Determination in Power System State Estimation", *IEEE Trans. Power Syst.*, Vol. PWRS–1, No. 2, pp.108–114, May 1986

[18] H. Mori, S. Tsuzuki, "A Fast Method for Topological Observability Analysis Using a Minimum Spanning Tree Technique", *IEEE/PES 1990 Summer Meeting,* Minneapolis, Minnesota, July 15–19, 1990

[19] A. Monticelli, F.F. Wu, "Network Observability: Theory", *IEEE Trans. Power Appar. & Syst.*, Vol. PAS–104, May 1985, pp. 1042–1048

[20] A. Monticelli, F.F. Wu, "Network Observability: Identification of Observable Islands and Measurement Placement", *IEEE Trans. Power Appar. & Syst.*, Vol. PAS–104, May 1985, pp. 1035–1041

[21] F.F. Wu, A. Monticelli, "Observability Analysis and Bad Data Processing for State Estimation Using Hachtel's Augmented Matrix Method", *IEEE Trans. Power Appar. & Syst.*, Vol. PWRS–3, May 1988, pp. 604–611

[22] K.A. Clements, "Observability Methods and Optimal Meter Placement", *Electrical Power & Energy Systems,* Vol. 12, No. 2, April 1990

[23] I. Slutsker, J. Scudder, "Network Observability Analysis Through Measurement Jacobian Matrix Reduction", *IEEE Trans. Power Appar. & Syst.*, Vol. PWRS–2, No. 2, May 1987

[24] R.L. Chen, "A Fast Integer Algorithm for Observability Analysis Using Network Topology", *IEEE Trans. Power Syst.*, Vol. 5, No. 3, August 1990

[25] K.A. Clements, P. Davis, "Power Network Observability: Symbolic Factorization and Spanning Trees", *1989 International Symp. on Circuits and Systems,* Portland, Oregon, May 8–11

[26] V.H. Quintana, A. Simoes–Costa, A. Mandel, "Power System Observability Using a Direct Graph–Theoretic Approach", *IEEE Trans. Power Appar. & Syst.*, Vol. PAS–101, March 1982, pp. 617–626

[27] R. Nucera, M. L. Gilles, "Observability Analysis: A New Topological Algorithm", *IEEE/PES 1990 Summer Meeting,* Minneapolis, Minnesota, July 15–19

[28] E.D. Crainic, H.P. Horisberger, X.D. Do, D. Mukhedkar, "Power Network Observability: The Assessment of the Measurement System Strength", *IEEE Trans. Pwr. Syst.*, Vol. 5, No. 4, November 1990

[29] A.E. Taylor, W.R. Mann, Advanced Calculus,Third Edition, *John Wiley & Sons, Inc.*, 1983

[30] E.D. Crainic, E.D. Do, P.J. Lagace, D. Mukhedkar, "Power System Observability: On–line Measurement Set Contingency Analysis", *Proceedings of the IFAC Symposium on Power System Modelling and Control,* Brussels, September 5–8, 1988, Pergamon Press, May 1989

[31] J.A. Howell, "Exact Solution of Linear Equations Using Residue Arithmetic", *Commun. Acm.*, Vol. 14, pp.180–184, 1971

[32] S. Cabay, T.P. Lam, "ESOLVE: Congruence Technique for the Exact Solution of Integer Systems of Linear Equations", *ACM Trans. on Mathematical Software,* Vol. 3, No. 4, pp. 404–410

[33] J. Springer, "Exact Solution of General Integer Systems of Linear Equations", *ACM Trans. on Mathematical Software,* Vol. 12, No. 1,

pp.51–61, March 1986

[34] E.S. Page, L.B. Wilson, An Introduction to Computational Combinatorics, *Cambridge University Press,* 1979

[35] H.T. Lau, "Combinatorial Heuristic Algorithms With Fortran", *Lecture Notes in Economics and Mathematical Systems*, Springer–Verlag, 1986

[36] C.H. Papadimitrious, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, *Prentice–Hall, Inc.*, 1982

[37] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, *Holt, Rinehart and Winston*, 1976

[38] D.J.A. Welsh, Matroid Theory, *Academic Press,* 1976

[39] N. Christofides, Graph Theory: An Algorithmic Approach, *Academic Press,* 1975

[40] W.L. Price, Graphs and Networks, *Butterworth & Co.*, 1971

[41] D.E. Knuth, The Art of Computer Programming, Volume 1, Second Edition, *Addison–Wesley*, 1968

VITA

Michael G. Cheniae was born in Baltimore, Maryland on October 17, 1960. He received his B.S. degree in Electrical Engineering from the United States Naval Academy in 1982. He served in the U.S. Nuclear Submarine Force for 7 years, qualifying as Chief Engineer of Navy Nuclear–powered Ships. He is currently in the process of completing the requirements for an M.S. degree in Electrical Engineering at Virginia Polytechnic Institute and State University, Blacksburg, VA, his area of research being state estimation and observability analysis of power systems.

Michael G. Cheniae