Real-Time Modeling of Power Networks

ANJAN BOSE, SENIOR MEMBER, IEEE, AND KEVIN A. CLEMENTS, SENIOR MEMBER, IEEE

Invited Paper

The use of large digital computers in control centers has made it possible to track the changing conditions in the power system with a mathematical model in the computer. This real-time model can be used to assess the security of the present system as well as to check out possible control strategies. In this paper the various steps in constructing the model from the real-time measurements are described. These steps include the determination of the network topology, the estimation of the network state, and the approximate modeling of the unobservable (external) network. This paper also discusses the checks for observability and bad measurements, and the calculation of bus load forecast factors and generator penalty factors.

I. Introduction

The security of a power system can be defined as its ability to withstand contingencies. It is relatively easy to check the measurement data for present violations of security limits, but it takes several analytical steps to determine the effects on the present system of a particular contingency. Such analysis for the off-line study of steady-state and dynamic effects of contingencies became commonplace with the advent of the digital computer. The computerization of the control center made it feasible to implement such off-line analysis in the on-line environment. Since on-line analysis must be continually updated, the computation time required for dynamic contingency analysis is too long to be useful for operations. However, steady-state analysis for a large number of contingencies can be completed quickly enough to provide timely alert messages to the operator.

The contingency analysis has to be done on a model of the power system. For off-line studies, this model is specified by the user through the input data. For on-line contingency analysis, the model must reflect the present conditions of the power system. Thus the model must be built from the real-time measurements before contingencies can be analyzed. This building of the real-time model of the power network is the subject of this paper. Although on-line contingency analysis was the major motivation to develop methods for real-time modeling, there are several other uses of the real-time model. It can be used for better monitoring of the system by detecting erroneous as well as estimating the incorrect or missing measurements. It can provide updated penalty factors to the economic dispatch program for more economic operation. It can be used to study possible control strategies like switching operations, volt-VAR coordination, economic operation within security constraints, and many others.

The power system model needed for contingency analysis is a solved power network described in terms of buses and branches. The model is built in two parts, one representing the internal system from which the control center receives telemetered data and the other representing the external system which consists of the rest of the interconnected system. Each part is built in two steps, the first being the determination of the network topology and the second being the solution for the complex bus voltages (or states) of the network. The step by step process is shown in Fig. 1.

These programs are supported by a database that contains the description of the network in terms of its parameters such as branch impedances and connectivity. These data are combined with real-time measurements to construct the real-time model. The topology processor picks out the status of circuit breakers and switches from the real-time data and, using the connectivity data from the database, determines the present network topology. This topology and all the other measurements are then used by the state estimator to solve for the bus voltages. Since the availability of real-time measurements can change because of failures in the telemetering equipment, an observability check is usually made before the state estimator solution is executed. Those parts of the network from which measurements are normally not received are always unobservable. The observability check only examines the normally observable portions and identifies those buses that may have become temporarily unobservable. These buses can then be made observable by adding pseudo-measurements or taken out of the state estimator calculation and lumped with the external model.

After the state estimator has solved for the observable network, a check for bad measurements is made. If bad data are detected and identified, they can be removed and the estimation updated. The unobservable portion of the interconnected network is then modeled. This portion can

Manuscript received September 20, 1986; revised September 24, 1987.

A. Bose is with the Department of Electrical and Computer Engineering, Arizona State University, Tempe, AZ 85287-5706, USA.

K. A. Clements is with the Worcester Polytechnic Institute, Worcester, MA 01609, USA.

IEEE Log Number 8717970.

0018-9219/87/1200-1607$01.00

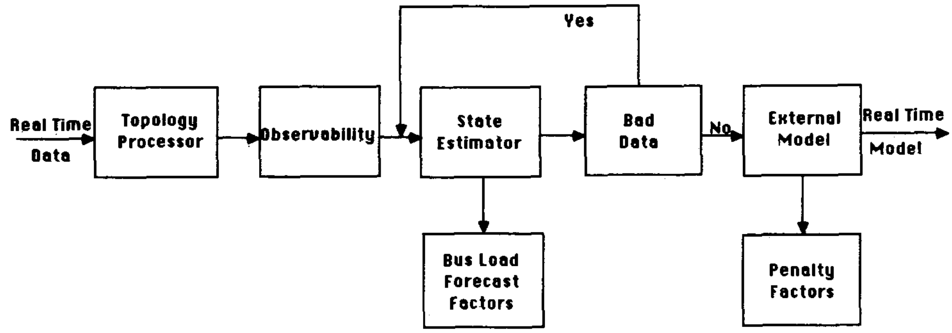

Fig. 1. Steps for building real-time model.

be quite large and the only reason to model it is to obtain accurate results for the subsequent contingency analysis or other study functions. Since the remote parts of the interconnected network have very little effect on the internal system, a reduced model may be adequate. Even for a reduced model, certain assumptions have to be made about the status of equipment and loading levels to obtain a solved external model.

The external model is needed mainly for the contingency analysis and is thus calculated just before the contingency analysis is done. A typical period of execution for these two functions is between 15 and 30 min in modern control centers. The transmission loss penalty factors for generators and tie-lines are also updated from the real-time model and the above periodicity is usually more than adequate for this calculation. The real-time model can also be used for automatic or operator-initiated determination of control strategies.

The state estimator constructs the real-time model of the internal system. Since the state estimator is a filter for the real-time measurements, the resulting model can be useful to the operator as a check for missing or suspicious data. For this reason, the internal model is usually updated more frequently than the external model and a typical periodicity for the network topology and state estimator calculations is 5 to 10 min. Since the state estimator calculates the bus loads every few minutes, it is possible to track the relationship of each load to the system load over time. These factors can be used to forecast bus loads when they are needed as pseudo-measurements or in other study functions.

Each of the functions shown in Fig. 1 are discussed in the following sections. Section II describes the network topology processor and Section III the state estimator. Section IV is on observability and Section V is on bad data detection and identification. Calculation of bus load forecast factors is discussed in Section VI, the modeling of the external system in Section VII, and the calculation of penalty factors is described in Section VIII. Section IX presents some of the other considerations such as software design, implementation, database, and man-machine interface which play major roles in the efficiency and utility of these functions. Some conclusions and references are presented at the end of the paper.

Schweppe and Handschin [1] had reviewed the state of the art in state estimation in 1974 and this present review explores the advances since then. Since much of the advancement has been in the wide-spread implementation of state estimators as well as in the new development of algorithms, this paper describes not only the state estimator but also the associated functions that are needed to completely construct the real-time model. This review is largely directed to the present general practice in today's control centers instead of a complete survey of the published literature.

II. Network Topology Processor

The function of the network topology processor is to determine the present topology of the network from the telemetered status of circuit breakers. The database describes the network connectivity in terms of bus-sections and circuit breakers. All equipment, such as generators, load feeders, shunt reactors, transformers, transmission lines, etc., are connected to bus-sections. Bus-sections within one voltage level at a substation may be connected together by circuit breakers. A simple power system with this level of detail is shown in Fig. 2 and its associated data

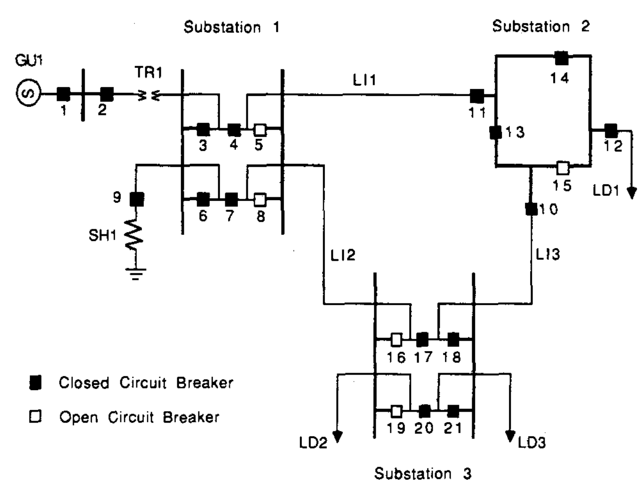

Fig. 2. Bus section/circuit breaker network model.

are shown in Table 1. The circuit breaker status data shown in the table are not part of the base data but are telemetered and are subject to change.

These status data are used to determine the present topology described in terms of buses and branches. For the circuit breaker status in Table 1, the power system of Fig. 2 can be represented as Fig. 3 in its bus-branch form. As the status of circuit breakers changes in real time, the bus-

Table 1 Network Topology Input Arrays

Substation Number	Bus Sections			Circuit Breakers			
	No.	Type	Equipment Ident.	No.	From B. Sec.	To B. Sec.	Status
	1	gen. unit	GU1	1	1	2	closed
	2	connection		2	2	3	closed
	3	transformer	TR1	3	4	4	closed
	4	connection		4	6	6	closed
1	5	connection		5	7	5	open
	6	transformer	TR1	6	4	8	closed
	7	line	LT1	7	8	9	closed
	8	connection		8	9	5	open
	9	line	LT2	9	10	8	closed
	10	shunt	SH1				
	11	connection		10	12	15	closed
	12	connection		11	14	11	closed
2	13	connection		12	13	16	closed
	14	line	LT1	13	11	12	closed
	15	line	LT3	14	11	11	closed
	16	load	LD1	15	12	13	open
	17	connection		16	17	19	open
	18	connection		17	19	20	closed
3	19	line	LT2	18	20	18	closed
	20	line	LT3	19	17	21	open
	21	load	LD2	20	21	22	closed
	22	load	LD3	21	22	18	closed

branch topology is expected to change, and the network topology processor must determine the new topology

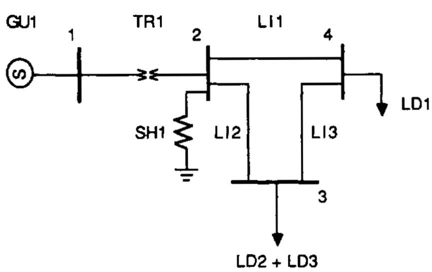

Fig. 3. Bus/branch network model.

whenever there is a change. Thus this program needs to run only if there is a status change. For no status change, this program can be skipped in a periodic cycle as the topology does not change.

The output of the network topology processor is the traditional data that describe a bus-branch oriented network. Thus each of the buses must be identified together with the generation, loads, and shunts at these buses. Also the connectivity between the buses due to the transmission lines and transformers has to be described. In addition, the topology processor must identify network islands and discard those that are not energized, that is, have no generation. Isolated buses and branches are trivial de-energized islands and will be discarded by this process.

There are several methods to convert bus-section–circuit breaker topology into bus–branch topology, some including the use of logic tables and incidence matrices. However, only one basic method [2], [3] using a tree search algorithm has been widely used in practice and is briefly described here. It consists of three sequential steps described in the following subsectons.

A. Substation Configuration

In the first step, the bus-sections at each substation voltage level are processed to determine if they are connected together by closed circuit breakers. At the beginning of the step, each bus-section is considered a potential separate bus. At the end of this tree search process, all bus-sections connected by closed circuit breakers become part of one bus. Thus each separate bus in the topology is uniquely identified in this step together with its constituent bus-sections.

It is not convenient to sequentially number the buses in this step as some of them may be found to be de-energized in the next steps. It is sufficient for them to be uniquely identified. When initializing the program, every bus-section and circuit breaker has to be processed in this step. In the tracking mode, however, only those substations in which circuit breaker status changes have taken place need to be processed. This step is usually very fast as the number of status changes during each cycle is normally quite small.

B. Network Configuration

In this step, all the energized network islands are identified. The tree search process used here is identical to that in the first step. Instead of bus-sections being combined by closed circuit breakers into buses, the buses are combined by branches into islands. Starting from a generator bus guarantees that the island is energized. As buses are added to the island, they can be sequentially numbered. Also the number of connections to a bus is counted for use in optimal ordering later. When no more buses can be found to be added to the first island, a search is made for an unprocessed generator bus. If one is found, it is used to start the search for the next island. Otherwise, this step is completed and all energized islands identified.

It should be pointed out that the optimal ordering of the buses can be done at this stage as the number of connections to each bus is already known. This has the advantage of carrying forward to subsequent calculations buses that are numbered optimally. Otherwise, the later calculations have to keep track of the mapping between sequential numbers and optimal numbers, a more common practice.

C. *Equipment-Bus Tables*

In this final step, all the equipment connected to the buses is tabulated. Since the equipment connected to each bus section is known and the bus-sections constituting a bus are known from the first step, the connectivity of the equipment to the buses can be established. The tables produced should be structured for easy use by subsequent programs. For example, each equipment type can be processed into separate tables to accommodate generator control, transformer control, or shunt switching algorithms in later programs. Generator and load buses are identified in this step and slack buses may be identified for each island.

If there is a status change, the three steps outlined above are executed to obtain the new topology. The new topology has all new bus numbers which have no relation to the old numbers. The solution matrices have to be recorded and refactorized for all subsequent programs such as the state estimator. Also, the iterative solutions have to be started from a flat start.

Many times, however, a status change may cause very little or no change in topology. For small changes, there are ways to obtain network solutions without reordering and refactorization of the matrices and by starting the iterative process from the previous solution. This is, of course, computationally much faster but the change in topology must be tracked, that is, the new topology must be compared to the topology of the previous cycle. To do this, Prais and Bose [4] have proposed a bus-numbering scheme that instead of renumbering all the buses, gives new numbers to newly formed buses and deletes those bus numbers that are eliminated. Admittedly, such changes will move the ordering away from optimal but for small changes in topology this does not affect the efficiency very much. Of course, large changes or an accumulation of small changes over time will require optimal renumbering of the buses. However, a tracking topology processor is not yet in common use.

III. State Estimation

The static state of an electric power network is described by the vector of bus voltage magnitudes and angles. An estimate of the state can be computed from system data consisting of network structural information, transmission system parameter values, and a sufficient set of power and voltage measurements. For a network with N buses, the state vector contains N bus voltage magnitudes and $N - 1$ bus voltage angles. One of the buses is chosen as the reference bus and is assigned a voltage angle of zero degrees. The state vector, denoted $\boldsymbol{x}$, thus has dimension $n = 2N - 1$. Computation of the state estimate provides complete real-time information on the current condition of the electric power network.

If there are m measurements, then these measurements can be written as an m-dimensional vector $\boldsymbol{z}$, which is related to the state vector through the measurement equation

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{w} \tag{1}$$

where $\boldsymbol{h}(\boldsymbol{x})$ is the nonlinear vector function relating the measurement vector to the state vector, and $\boldsymbol{w}$ is a vector of measurement errors.

The power system state estimation problem was first formulated as an unconstrained least squares problem. Equality constraints, such as those imposed by zero-injection buses in the network were treated as very accurate measurements rather than as equality constraints. The unconstrained least squares formulation is described first; this is followed by the constrained formulation; and then by the decoupled formulation.

A. *The Unconstrained Weighted Least Squares Formulation*

In this formulation, the state estimate is computed as a least squares solution to the overdetermined set of equations obtained from the measurements. The least squares estimate of $\boldsymbol{x}$, denoted $\hat{\boldsymbol{x}}$, minimizes the weighted least squares function

$$J(\boldsymbol{x}) = (1/2)\,[\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]^T R^{-1} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] \tag{2}$$

where R is a diagonal matrix. The diagonal entries of R, σ_{ii} are often chosen as the measurement error variances. However, perfect (zero error variance) measurements cannot be handled in this manner and must be treated as very accurate measurement rather than as equality constraints.

Several methods have been proposed for solving the above least squares problem. The first solution method proposed for power network state estimation [1] is based on the so-called normal equation. The normal equation is derived from the necessary condition for $\boldsymbol{x}$ to be a minimum of $J(\boldsymbol{x})$, namely, that

$$\left.\frac{\partial J(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = -H(\hat{\boldsymbol{x}})^T R^{-1} [\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}})] = \boldsymbol{0} \tag{3}$$

where $H(\hat{\boldsymbol{x}})$ is the measurement Jacobian matrix

$$H(\hat{\boldsymbol{x}}) = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}. \tag{4}$$

Since (3) is nonlinear, its solution generally requires the application of an iterative method. The most commonly employed method is that of Newton, which is based on the following linear approximation of $\boldsymbol{h}(\hat{\boldsymbol{x}}^{i+1})$:

$$\boldsymbol{h}(\hat{\boldsymbol{x}}^{i+1}) = \boldsymbol{h}(\hat{\boldsymbol{x}}^{i}) + H(\hat{\boldsymbol{x}}^{i})\,\Delta\hat{\boldsymbol{x}}^{i+1}. \tag{5}$$

If a further approximation is made by assuming that $H(\hat{\boldsymbol{x}}^{i+1}) = H(\hat{\boldsymbol{x}}^{i})$ then the following iterative sequence of equations results:

$$C(\hat{\boldsymbol{x}}^{i})\,\Delta\hat{\boldsymbol{x}}^{i+1} = H(\hat{\boldsymbol{x}}^{i})^T R^{-1} [\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}}^{i})] \tag{6}$$

$$\hat{\boldsymbol{x}}^{i+1} = \hat{\boldsymbol{x}}^{i} + \Delta\hat{\boldsymbol{x}}^{i+1} \tag{7}$$

where

$$C(\hat{\boldsymbol{x}}^{i}) = H(\hat{\boldsymbol{x}}^{i})^T R^{-1} H(\hat{\boldsymbol{x}}^{i}). \tag{8}$$

The iterations are initialized either at flat-start conditions or at the previous state estimate.

Equation (6) is referred to as the normal equation. Note that a unique solution for $\Delta\hat{\boldsymbol{x}}^{i+1}$ can be obtained from (6) only if $C(\hat{x}^i)$ is nonsingular. The necessary condition for $C(\hat{x}^i)$ to be nonsingular is that $H(\hat{x}^i)$ be rank n, i.e., full rank. When $H(\hat{x}^i)$ has rank n, the network is said to be observable. For the estimate to be reliable, the number of measurements needs to be greater than n, i.e., there should be redundancy in the measurements to counteract the errors in the measurements. Network observability and measurement redundancy is addressed in Section IV.

Equation (6) is similar in structure to the equation encountered in the Newton power flow. In particular, $C(\hat{x}^i)$ is a symmetric positive-definite matrix that is slightly less sparse than the Jacobian in the Newton power flow. An efficient method for solving (6) is to perform ordered triangular decomposition on $C(\hat{x}^i)$ and then solve for $\Delta\hat{x}^{i+1}$ via forward and backward solution.

The triangular decomposition of $C(\hat{x}^i)$ is written as

$$PC(\hat{x}^i)P^T = U^T DU \tag{9}$$

where P is a permutation matrix, U is an upper unit triangular matrix ($U_{ij} = 0$ for $i > j$, $U_{ii} = 1$), and D is a diagonal matrix. As in the Newton power flow problem, the Tinney II ordering scheme has been found to perform well in maintaining sparsity in U. As pointed out in the previous section, this ordering can be done when forming the network topology, but in general practice it is often done when solving the normal equation.

Although the above method is utilized in most of the state estimators in the field today, it can be ill-conditioned in certain situations; consequently, researchers have investigated other solution methods with better numerical properties. Gu *et al.* [5] proposed using the method of Peters and Wilkinson, the first step of which is to compute factorization of H

$$P_1 H(\hat{x}^i) P_2 = \tilde{L}\tilde{D}\tilde{U} \tag{10}$$

where P_1 and P_2 are permutation matrices, $\tilde{L}$ is an $m \times n$ lower unit trapezoidal matrix ($\tilde{L}_{ij} = 0$ for $i < j$, $\tilde{U}_{ii} = 1$), $\tilde{D}$ is a diagonal matrix, and $\tilde{U}$ is an $n \times n$ upper unit triangular matrix.

The rows of H are ordered as the factorization is performed. Peters and Wilkinson used completed pivoting for nonsparse cases, but for sparse matrices a good choice for the pivot element would be a nonzero element for which the product of the number of nonzero row elements and nonzero column elements is minimum. In the method of Peters and Wilkinson, rather than solving directly for $\Delta\hat{x}^{i+1}$, a new vector variable

$$\Delta\hat{y}^{i+1} = \tilde{U}\Delta\hat{x}^{i+1} \tag{11}$$

is introduced. When this is substituted into (5) one obtains

$$\tilde{L}^T R^{-1} \tilde{L}\Delta\hat{y}^{i+1} = \tilde{L}^T R^{-1}[\mathbf{z} - \mathbf{h}(\hat{x}^i)]. \tag{12}$$

Equation (12) is another normal equation but, because $\tilde{L}$ is a unit lower trapezoidal matrix, the equation tends to be better numerically conditioned than (6). The second step in the method of Peters and Wilkinson is to factor $\tilde{L}^T R^{-1}\tilde{L}$ and solve for $\Delta\hat{y}^{i+1}$ by forward and backward solution. The third step is to compute $\Delta\hat{x}^{i+1}$ from (11) via backward substitution.

The method requires additional computations, namely, the factorization of $H(\hat{x}^i)$; however, it offers the ability of tradeoff between speed and stability in the calculation of the state estimate. Gu *et al.* report substantial improvement in numerical roundoff error for the method of Peters and Wilkinson when compared with the normal equations technique.

Solution of the power system state estimation problem using orthogonal transformation methods is a third technique that has been investigated [6]–[8]. An orthogonal transformation matrix T is a square matrix such that

$$T^T T = I. \tag{13}$$

Suppose an orthogonal transformation matrix can be found such that

$$T\tilde{H}(\hat{x}^i) = \begin{bmatrix} U \\ 0 \end{bmatrix} \tag{14}$$

where

$$\tilde{H}(\hat{x}^i) = H(\hat{x}^i)R^{-1/2}. \tag{15}$$

Solution of the nonlinear least squares problem by Newton's method is equivalent to solving a sequence of linear least squares problems of the form

$$J(\Delta\hat{x}^{i+1}) = (1/2)\,[\mathbf{z} - H(\hat{x}^i)\Delta\hat{x}^{i+1}]^T R^{-1} \cdot [\mathbf{z} - H(\hat{x}^i)\Delta\hat{x}^{i+1}] \tag{16}$$

which may be rewritten as

$$J(\Delta\hat{x}^{i+1}) = (1/2)\,[T\mathbf{z} - TH(\hat{x}^i)\Delta\hat{x}^{i+1}]^T R^{-1} \cdot [T\mathbf{z} - TH(\hat{x}^i)\Delta\hat{x}^{i+1}]. \tag{17}$$

Let

$$T\mathbf{z} = \begin{bmatrix} \mathbf{y} \\ \mathbf{e} \end{bmatrix}. \tag{18}$$

Then (17) may be written as

$$J(\Delta\hat{x}^{i+1}) = (1/2)\,[\mathbf{y} - U\Delta\hat{x}^{i+1}]^T [\mathbf{y} - U\Delta\hat{x}^{i+1}] + \mathbf{e}^T\mathbf{e}. \tag{19}$$

Clearly, the value of $\Delta\hat{x}^{i+1}$ that minimizes $J(\Delta\hat{x}^{i+1})$ satisfies the equation

$$U\Delta\hat{x}^{i+1\prime} = \mathbf{y}. \tag{20}$$

U can be computed either by processing columns of $\tilde{H}$ (Householder's rotations) or by processing rows of $\tilde{\mathbf{H}}$ (Givens rotations). The same operations applied to $\mathbf{z}$ allow the computation of $\mathbf{w}$. Equation (20) is then employed to solve for $\Delta\hat{x}^{i+1}$. Although orthogonal transformation methods have good numerical properties, they are usually not as efficient as the normal equations algorithm. Orthogonal transformation methods have been widely applied to general least squares problems and are beginning to be used in control centers. The first papers proposing their application to power system state estimation were by Simoes-Costa and Quintana [6], [7].

A hybrid method combining aspects of the normal equations method and the orthogonal transformation method has also been described [9]. In this method, an orthogonal transformation method is used to compute U. Since U is the upper triangular factor of $C(\hat{x}^i)$, then (6) may be written as

$$U^T U\Delta\hat{x}^{i+1} = H(\hat{x}^i)^T R^{-1}[\mathbf{z} - \mathbf{h}(\hat{x}^i)]. \tag{21}$$

Equation (21) can then be solved by forward and backward substitution as in the normal equations approach. In this method, the ordering of the equations can be done by the Tinney II method so that U has exactly the same sparse structure as in the normal equations algorithm.

B. The Constrained Weighted Least Squares Formulation

Buses with neither load nor generation are referred to as zero injection buses. There is, in effect, a perfect measurement of bus injection available at such buses. It has been noted that the unconstrained least squares formulation cannot treat these perfect measurements properly. Although earlier implementations modeled equality constraints as accurate measurements by setting the corresponding elements of the R matrix to small values, this may lead to ill-conditioning and nonconvergence of the state estimate under certain circumstances. Aschmoneit *et al.* [10] were the first to propose inclusion of equality constraints into the least squares formulation and this method of handling zero-injection buses is common today. Suppose there are p constraints imposed by zero-injection buses in the network and let $\boldsymbol{g}(\boldsymbol{x})$ to be p-dimensional vector of constraint equations, then the constrained least squares formulation may be expressed as

minimize

$$J(\boldsymbol{x}) = (1/2)\,[\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]^T R^{-1} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] \tag{22}$$

subject to

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}. \tag{23}$$

This constrained minimization problem may be solved by the method of Lagrange multipliers. The Lagrangian $L(\boldsymbol{x})$ is formed as

$$L(\boldsymbol{x}) = (1/2)\,[\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]^T R^{-1} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] + \lambda^T \boldsymbol{g}(\boldsymbol{x}) \tag{24}$$

where λ is the Lagrange multiplier vector.

The necessary conditions for the solution of the constrained problem are then given by the following two equations:

$$\left.\frac{\partial L(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}} = -H(\hat{\boldsymbol{x}})^T R^{-1} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] + G(\hat{\boldsymbol{x}})\lambda = \boldsymbol{0} \tag{25}$$

and

$$\left.\frac{\partial L(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}} = -\boldsymbol{g}(\hat{\boldsymbol{x}}) = \boldsymbol{0} \tag{26}$$

where $G(\hat{\boldsymbol{x}})$ is the constraint equation Jacobian matrix

$$G(\boldsymbol{x}) = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}}. \tag{27}$$

Solution of (25) and (26) by Newton's method leads to the following system of equations to be solved at each iteration:

$$\begin{bmatrix} C(\hat{\boldsymbol{x}}^i) & G(\hat{\boldsymbol{x}}^i)^T \\ G(\hat{\boldsymbol{x}}^i) & 0 \end{bmatrix} \begin{bmatrix} \Delta\hat{\boldsymbol{x}}^{i+1} \\ \lambda^{i+1} \end{bmatrix} = \begin{bmatrix} H(\hat{\boldsymbol{x}}^i)^T R^{-1} \Delta \boldsymbol{z}^i \\ -\boldsymbol{g}(\hat{\boldsymbol{x}}^i) \end{bmatrix} \tag{28}$$

where

$$\Delta \boldsymbol{z}^i = \boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}}^i). \tag{29}$$

As in the case of the normal equation, the above equations can be solved by ordered triangular factorization. The matrix on the left-hand side of (28) is not positive-definite however, consequently, the ordering algorithm must take this into account by testing the value of potential pivot elements.

Another method for solving constrained least squares problems is the sparse tableau method, also known as Hachtel's method. In this technique, the problem is described by a larger albeit sparser set of equations. Gjelsvik, Aam, and Holten [11] have applied this method to power system state estimation. In this technique, the weighted least squares cost function is written as

$$J(\boldsymbol{x}) = (1/2)\, \boldsymbol{r}^T R^{-1} \boldsymbol{r} \tag{30}$$

where

$$\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}). \tag{31}$$

Equation (31) is treated as an equality constraint. The Lagrangian for this problem may then be written as

$$L(\boldsymbol{x}) = (1/2)\, \boldsymbol{r}^T R^{-1} \boldsymbol{r} - \lambda^T \boldsymbol{g}(\boldsymbol{x}) - \gamma^T [\boldsymbol{r} - \boldsymbol{z} + \boldsymbol{h}(\boldsymbol{x})] \tag{32}$$

and the solution for the least squares estimate must satisfy the following necessary conditions:

$$\left.\frac{\partial L(\boldsymbol{x})}{\partial \boldsymbol{r}}\right|_{x=\hat{x}} = R^{-1}\boldsymbol{r} - \gamma = \boldsymbol{0} \tag{33}$$

$$\left.\frac{\partial L(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}} = -G(\hat{\boldsymbol{x}})^T \lambda - H(\hat{\boldsymbol{x}})\gamma = \boldsymbol{0} \tag{34}$$

$$\left.\frac{\partial L(\boldsymbol{x})}{\partial \lambda}\right|_{x=\hat{x}} = -\boldsymbol{g}(\hat{\boldsymbol{x}}) = \boldsymbol{0} \tag{35}$$

$$\left.\frac{\partial L(\boldsymbol{x})}{\partial \gamma}\right|_{x=\hat{x}} = \boldsymbol{r} + \boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}}) = \boldsymbol{0}. \tag{36}$$

Application of Newton's method to the above equations results in the following system of $n + m + p$ equations to be solved at each iteration:

$$\begin{bmatrix} R & 0 & H(\hat{\boldsymbol{x}}^i) \\ 0 & 0 & G(\hat{\boldsymbol{x}}^i) \\ H(\hat{\boldsymbol{x}}^i)^T & G(\hat{\boldsymbol{x}}^i)^T & 0 \end{bmatrix} \begin{bmatrix} \gamma^{i+1} \\ \lambda^{i+1} \\ \Delta\hat{\boldsymbol{x}}^{i+1} \end{bmatrix} = \begin{bmatrix} \Delta \boldsymbol{z}^i \\ -\boldsymbol{g}(\hat{\boldsymbol{x}}^i) \\ \boldsymbol{0} \end{bmatrix}. \tag{37}$$

Gjelsvik *et al.* report good performance with respect to both efficiency and numerical stability of the sparse tableau method. Since the coefficient matrix is not positive-definite, ordering of the equations generally requires a numerical test for the candidate pivot elements. Although the number of equations is much larger than for the normal equations method, the number of floating-point operations required for solution may actually be fewer in certain cases.

One may also extend orthogonal transformation methods as well as the method of Peters and Wilkinson to treat constrained least squares problems.

C. Fast Decoupled Solution Algorithms

As with the Newton load flow, fast decoupled algorithms for power system state estimation have been developed and implemented. These methods are based on the strong coupling between active power flows and bus voltage angles (P–θ coupling) and between reactive power flows and bus voltage magnitudes (Q–V coupling) compared with the P–V and Q–θ couplings. There are many possible variations of decoupled algorithms, but the most successful of these seem to be those that are very similar to the fast decoupled load flow algorithm that has evolved over a period of time. To begin with, the state vector is partitioned into bus voltage angles and bus voltage magnitudes

$$x = \begin{bmatrix} \theta \\ V \end{bmatrix}. \quad (38)$$

The measurement vector is transformed by dividing real and reactive power measurements by the bus voltage magnitude at the measured bus. The measurements are denoted

$$z^T = [t^T, i^T, u^T, k^T, e^T] \quad (39)$$

where t contains active lineflow measurements divided by the bus voltage on the "from" side of the measurement, i contains active bus injection measurements divided by the bus voltage, u contains reactive lineflow measurements divided by the bus voltage on the "from" side of the measurement, k contains reactive bus injection measurements divided by the bus voltage, and e contains bus voltage magnitude measurements. z is partitioned into active and reactive measurements

$$z = \begin{bmatrix} z_P \\ z_Q \end{bmatrix} = \begin{bmatrix} h_P(x) \\ h_Q(x) \end{bmatrix} + \begin{bmatrix} w_P \\ w_Q \end{bmatrix} \quad (40)$$

where

$$z_P = [t^T, i^T] \quad (41)$$

and

$$z_Q = [u^T, k^T, e^T]. \quad (42)$$

The measurement Jacobian matrix and the information matrix are written in partitioned form as

$$H(x) = \begin{bmatrix} H_{P\theta}(x) & H_{PV}(x) \\ H_{Q\theta}(x) & H_{QV}(x) \end{bmatrix} \quad (43)$$

and

$$C(x) = \begin{bmatrix} C_{P\theta}(x) & C_{PV}(x) \\ C_{Q\theta}(x) & C_{QV}(x) \end{bmatrix}. \quad (44)$$

There are two variations of fast decoupled algorithms that have been proposed [12]–[14]. In the so-called algorithm decoupled estimator, the right-hand side of (6) is computed exactly, but the nondiagonal blocks of $C(x^i)$ matrix are set to zero while the diagonal blocks are approximated by their flat-start values. With these approximations, (6) is replaced by the following decoupled pair of equations:

$$C_{P\theta}\Delta\theta^{i+1} = [H_{P\theta}^T | H_{Q\theta}^T] R^{-1} \Delta z^i \quad (45)$$

and

$$C_{P\theta}\Delta V^{i+1} = [H_{PV}^T | H_{QV}^T] R^{-1} \Delta z^i. \quad (46)$$

In the second version or the model decoupled algorithm, the right-hand side of (6) is also approximated by setting the off-diagonal blocks of H to zero. When this approximation is made the estimation equations become

$$C_{P\theta}\Delta\theta^{i+1} = H_{P\theta}^T R_P^{-1} \Delta z_P^i \quad (47)$$

and

$$C_{QV}\Delta V^{i+1} = H_{QV}^T R_Q^{-1} \Delta z_Q^i \quad (48)$$

where

$$\Delta z_P^i = z_P - h_P(x^i) \quad (49)$$

and

$$\Delta z_Q^i = z_Q - h_Q(x^i). \quad (50)$$

The model decoupled estimator has been found to be more reliable than the algorithm decoupled estimator. It should be noted that the decoupling principle can be applied not only to the normal equations method but also to the other least squares solution techniques described above.

D. Some Other Considerations

The solution algorithms described above assume that the measurement data are available in the right format and simultaneously. In practice, of course, the state estimator measurement data have to be mapped from the real-time database that is continually updated by the data acquisition system. For example, a bus injection measurement can, in reality, be an accumulation of several measurements of generator outputs and distribution line flows. A bus voltage measurement is often an average of several bus section voltages. The data acquisition system normally polls the different substations in sequence and hence there can be significant time skew within one cycle of measured data. The polling sequence can usually be set up such that all the data needed by the state estimator can be gathered within a time window of a few seconds and this seems to have been satisfactory for most implementations.

The problem of time skew is a lot more severe when all the data are not collected by the same data acquisition system. In the case of a power pool, the data may come from the several control centers belonging to the member companies. If some of the member companies have state estimators, the data passed to the pool could be estimated data. A pool estimator of this type is known as a hierarchical state estimator and although various ways of handling time skew and other data problems are known no field implementation experience is available yet.

In practice, the exact configuration of the power system and the measurement system poses unique implementation problems for a particular control center. For example, transmission lines with several unmeasured taps are inherently unobservable and require special approximations. Tap changing transformers, which are often unobservable, usually require special handling. The tap measurements, if available, can be used like any other measurement or can be considered perfect and utilized in the transformer model. Most state estimators today consider the tap another state to be estimated. Special cases, like three-winding transformers, require special handling.

It should be pointed out here that the field installation of all the on-line programs largely consists of installing the state estimator. The state estimator input is real-time data whereas all other network programs are fed from the state estimator output. The "cut over" of the state estimator is somewhat similar to that of AGC. It is first tested off-line on the on-line database and simulated real-time data. Then it is first tried on two or three small substations on-line. Once it starts working successfully on this, the rest of the system is cut over one or two substations at a time. In this process, bugs in the measurement data and their mapping have to be found and eliminated to successfully run the program. Parameter data in the database may have to be corrected. Some tuning of the R matrix may be needed. This shake-out can require a large effort but can be significantly helped by careful preparation of the parameter and connectivity database and proper check out of the SCADA database.

IV. Network Observability

When sufficient measurements are available so that the entire state vector of bus voltage magnitudes and angles throughout the network can be estimated, the network is said to be *observable*. As shown in the previous section, this is true when the rank of the Jacobian matrix of $\boldsymbol{h}(\boldsymbol{x})$ equals the number of unknown states. The rank of the measurement Jacobian matrix is, in turn, dependent on the locations and types of available measurements as well as on the network's topology. Normally, the metering system for the internal network, i.e., the controlled portion of the network, is designed so that the network will not only be observable, but also redundant. In fact, the placement of the meters is very important to state estimator reliability and is discussed later in this section.

Because availability of the measurements as well as network topology may vary with time, it is necessary to perform an observability test every time there is a change in the set of available measurements or the network topology. If the network is observable, state estimation may proceed. Otherwise, it is necessary to determine which buses are unobservable. These unobservable buses have to be either removed from the state estimator calculation or made observable by adding pseudo-measurements (see Section VI). It is possible to have several observable islands of buses and modern state estimators are capable of solving all the islands by providing a reference bus for each.

Two classes of observability determination algorithms have been studied: numerically based algorithms and topologically based algorithms.

A. Topological Observability

Topological algorithms which only use information about the network and measurement topology were developed in order to avoid the rather difficult task of numerical computation of the rank of the measurement Jacobian matrix. Such algorithms have been widely used in the state estimator observability programs. In [15], Clements and Wollenberg considered networks containing only lineflow and bus injection measurements. In the case of networks containing only lineflow measurements in which real and reactive measurements occur in pairs, the topological condition for observability is that there exists at least one bus voltage magnitude measurement and that a spanning tree of the entire network can be built using only measured lines. Finding such a tree can be done using one of the well-known tree search methods such as breadth-first or depth-first search. For an N-bus network with only bus injection measurements, the determination of observability is even simpler; there must be at least one bus voltage measurement and at least $N - 1$ bus injection measurements.

In the Clements–Wollenberg algorithm, these two ideas are combined to establish sufficient (though not necessary) conditions for observability. In the first phase of the algorithm, those regions of the network containing trees of flow measured lines are identified. These regions are called flow measurement observable islands. The remaining regions will, of necessity, contain only bus injection measurements. In each of these regions, two types of buses are defined: 1) *boundary buses* that are common to both measurement observable island and the non-flow-measured region and 2) the remaining buses which are called *internal buses*. The number of degrees of freedom of a non-flow-measured region is defined to be equal to the number of internal buses plus the number of adjacent flow-measured islands. A sufficient condition for observability is that the number of injection measurements in the region be at least equal to the number of degrees of freedom minus one and that no more than one boundary bus be unmeasured.

The Clements–Wollenberg algorithm is conservative in that, if a network is declared observable by it, then the network will always be observable; on the other hand, the algorithm may label certain observable networks as not observable. The algorithm assumes that real and reactive power measurements always occur in pairs and thus observability of the P-θ portion of the solution implies observability of the Q–V portion of the solution. Horton and Masiello [16] extended the Clements–Wollenberg algorithm by treating the P-θ and Q–V portions of the solution separately in a decoupled fashion.

In 1980, Krumpholz, Clements, and Davis published a graph-theoretic observability algorithm [17]. The algorithm is based on a fundamental theorem which states that the necessary and sufficient condition for a network to be observable is that it contains at least one observable spanning tree. Determination of observability of a tree is rather simple; in an observable tree, each branch is assigned to a measurement incident to it and each measurement can only be assigned to a single tree branch. Assignment of line flow measurements to branches is unique since a line flow measurement is only incident to a single branch. Bus injection measurements can be assigned to any of the branches incident to the measured bus. Rather than testing observability of all trees contained in a network, which is not practical, for a large network, the strategy of the Krumpholz–Clements–Davis algorithm is to first find a maximal observable spanning forest of flow measured branches and then use an algorithm similar to the network flow algorithm to enlarge the spanning forest to an observable spanning tree by assigning injection measurements to certain tree branches. This algorithm has been used in working state estimator observability programs.

Quintana, Simoes-Costa, and Mandel [18] proposed another graph-theoretic algorithm based on the observable spanning tree theorem of [17]. In their paper, they related the problem of finding an observable spanning tree to a problem in combinatorial mathematics called the matroid intersection problem. They applied the matroid intersection algorithm to determine whether an observable spanning tree exists. Another non-numeric algorithm was presented by Slutsker and Scudder [19], this algorithm is based on symbolic, rather than numerical, reduction of the measurement Jacobian matrix. The algorithm has been implemented in control centers, but no theoretical proof for the algorithm has been provided.

B. Numerical Observability

Monticelli and Wu [20]–[22] have proposed a numerical test for observability based on triangular decomposition of the information matrix $C(\boldsymbol{x})$. If $C(\boldsymbol{x})$ can be successfully factored without encountering any zeros in the diagonal, then the system is observable. On the other hand, if the network is not observable then one or more zeros will appear on the diagonal of the triangular factor. When this happens, this method adds a pseudo-measurement of bus voltage angle in the P–θ observability test at the bus corresponding to the zero diagonal element and then continues with the factorization process. These buses then are automatically identified as buses requiring injection measurements for observability. The algorithm also provides information about observable islands within the network if a solution is computed using an artificial measurement set of zero measurement values where actual power measurements are located and by choosing a different reference angle value for each pseudo-measurement of bus voltage angle used in the observability test. If power flows are zero within an observable island, then all bus voltage angles will equal that of the reference bus. The pseudo-measurements of angle in effect provide additional reference buses in the network in order to allow the computation to be performed. As a result, all buses in the same observable island will have the same voltage angle as the corresponding pseudo-measurement.

The numerical observability algorithms are new and are starting to be implemented at some control centers. They have the virtue of being conceptually simple and of employing numerical routines that are already needed for computation of the state estimate. There is, however, the potential of difficulty in determining whether a rather small number appearing on the diagonal is either a nonzero value or is actually zero but, because of numerical roundoff, is computed to be nonzero. Topological algorithms, on the other hand, require additional non-numerical routines that may be rather complex but generally run faster than numerical tests. No definitive studies have yet been done that indicate the superiority of either the topological or the numerical approach to observability determination.

C. Meter Placement

The observability of the network under normal network configuration and normal measurement availability determines the model desired from the state estimator. Thus it is necessary to determine that the metering system is adequate before a state estimator is implemented. Such a study usually entails the checking of observability under the available metering and then determining the placement of new meters. The algorithms used are the same as those discussed above [21], [23], [24].

However, the placement of meters has to take into account not only observability but also the redundancy required for good estimation. For an N-bus system observability requires a minimum of $2N - 1$ measurements and it is generally accepted that at least $3N$ measurements are needed for adequate redundancy. But just as the observability requires that the $2N - 1$ measurements be uniformly distributed over the buses, the redundant measurements must also be uniformly distributed to obtain the intended effect on state estimator accuracy. Also, this distributed redundancy is crucial for bad data detection, which is discussed in the next section.

V. Bad Data Detection

Bad data detection refers to the detection, identification, and elimination of measurements with large errors. Bad data detection relies on measurement redundancy and is based on analysis of the measurements residual vector

$$\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}}). \tag{51}$$

An approximate linear relationship between $\boldsymbol{r}$ and the state estimate error is obtained by linearizing $\boldsymbol{h}(\boldsymbol{x})$ about $\hat{\boldsymbol{x}}$. This relationship is given by

$$\boldsymbol{r} = H(\hat{\boldsymbol{x}})\tilde{\boldsymbol{x}} + \boldsymbol{w} \tag{52}$$

where

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}. \tag{53}$$

Using (52) and (3) one can relate the measurement residuals to the measurement errors by

$$\boldsymbol{r} = W\boldsymbol{w} \tag{54}$$

where the residual sensitivity matrix W is given by

$$W = I - H(\hat{\boldsymbol{x}})\, C(\hat{\boldsymbol{x}})^{-1}\, H(\hat{\boldsymbol{x}})^T R^{-1}. \tag{55}$$

The residual sensitivity matrix has dimension $m \times m$ and rank $m - n$. Linear dependencies among the columns of W are determined by the redundancy relationships among the measurements.

A. Bad Data Detectability and Identifiability

The degree to which bad data can be detected and identified depends on the degree of redundancy in the measurement set. Measurement redundancy relationships can be characterized by the following definitions. A *critical measurement* is one whose deletion from the measurement set results in loss of observability of the network. A *critical pair* of measurements is a pair of measurements, neither of which is critical, whose deletion from the set results in loss of observability. Similarly, one can define a *critical k-tuple* of measurements, none of which belongs to lower order critical tuples, whose deletion results in loss of observability.

A measurement error is said to be *detectable* if an error in the measurement shows up in the measurement residual vector. In order for a measurement error to affect the measurement residuals, it is necessary that the corresponding column of W be nonzero. It was shown [25] that an error in a measurement is detectable if and only if the measurement is not critical.

A single measurement error is said to be *identifiable* if the column of W corresponding to that measurement error is not colinear with any other column W. Clements and Davis show [26] that a single measurement error is identifiable if and only if the measurement is not critical and does not belong to any critical pairs. Conditions for multiple bad data detectability and identifiability are also derived in [26].

The *residual spread component* of measurement i is the set of measurements whose residuals are affected by an error in measurement i. Clearly, an error in measurement

i does not affect the residual of measurement j if W_{ji} is zero. Two measurement errors are said to be noninteracting if their residual spread components contain no common measurements. A topology-based method for determining residual spread components is presented in [25].

B. Statistical Tests for Detection and Identification of Single Bad Data

Under the assumption that the measurement error vector is Gaussian with zero mean and covariance matrix R, it can be shown that $\boldsymbol{r}$ is also a Gaussian random vector. It has zero mean and covariance

$$\Sigma = WR. \tag{56}$$

Testing for bad data can be viewed as a statistical hypothesis testing problem. Three statistical tests involving $\boldsymbol{r}$ have been used for bad data detection: one is based on testing the weighted sum of squares of residuals and the other two are based on examining the individual residuals.

The first test is a chi-squared test and is called the $J(\hat{\boldsymbol{x}})$ test. Under the hypothesis that no bad data are present, the weighted sum of squares of residuals

$$J(\hat{\boldsymbol{x}}) = \boldsymbol{r}^T R^{-1} \boldsymbol{r} \tag{57}$$

has a chi-squared distribution with $m - n$ degrees of freedom. One may test the no bad data hypothesis by checking whether or not

$$J(\hat{\boldsymbol{x}}) > X^2_{m-n}(\alpha) \tag{58}$$

where the α is a specified false alarm probability, i.e., the probability of exceeding the threshold when no bad data are present.

The $J(\hat{\boldsymbol{x}})$ test allows detection of bad data but does not identify the bad data. Furthermore, for false alarm probability levels frequently used (typically 0.01 or 0.05), it is not a very reliable test for bad data in the range of ± 3 to ± 20 standard deviations [27]. For this reason, a second test, the normalized residuals test, is often performed. The normalized residual for measurement i equals the measurement residual divided by the square root of its variance. Thus if the ith diagonal element of Σ is denoted ρ^2_{ii}, then the normalized residual for measurement i is

$$r_i^N = r_i/\rho_{ii}. \tag{59}$$

Obviously, this test can only be performed for those measurements whose residual variance is not equal to zero. The residual variance will be zero if the measurement is critical. If a measurement is critical then, in theory, both its residual value and the computed residual variance will be identically zero. In practice, due to numerical roundoff, both numbers will be small nonzero numbers and their ratio will be a meaningless number. For this reason, it is important that critical measurements be identified prior to performing the normalized residuals test.

Normalized residuals are zero-mean unity-variance Gaussian random variables when no bad data are present. One may test each r_i^N individually and if

$$|r_i^N| > \gamma \tag{60}$$

for one or more i then the no bad data hypothesis is rejected. A value of γ is chosen which determines the false alarm probability.

It can be shown [28] that if all of the measurements were error-free except for one, say measurement k, then

$$|r_k^N| \geq |r_i^N|, \quad i \neq k. \tag{61}$$

Note that the inequality in the above equation is not strict. If the inequality were strict, then one would be guaranteed that single instances of bad data could be detected and identified provided that the other measurement errors were relatively small. This will be the case if measurement k is not critical and is not a member of any critical measurement pair. If measurement k is a member of a critical pair of measurements, then the magnitude of the normalized residual of the other member of the critical pair will equal that of measurement k.

There is a computational burden associated with performing the normalized residuals test since it is necessary to compute the diagonal elements of Σ, given by

$$\rho^2_{ii} = \sigma^2_{ii} - \boldsymbol{h}_i C(x)^{-1} \boldsymbol{h}_i^T \tag{62}$$

where $\boldsymbol{h}_i$ is the ith row of $H(\hat{\boldsymbol{x}})$. To avoid the matrix inversion calculation burden, in practice, the residual was often normalized by σ_{ii}, the measurement error variance, instead of ρ_{ii}. However, even when a bad measurement is not part of a critical tuple, this calculation cannot discriminate between the bad measurement residual and some neighboring measurement residuals. This problem is known as smearing.

It was soon noticed that $\boldsymbol{h}_i$ is a sparse vector and only those elements of $C(\hat{\boldsymbol{x}})$ corresponding to pairs of nonzero entries in $\boldsymbol{h}_i$ are needed to compute the diagonal elements of Σ. These nonzero elements turn out to be the nonzero elements of $C(\hat{\boldsymbol{x}})$. It is possible to compute the needed elements of $C(\hat{\boldsymbol{x}})^{-1}$ using an efficient algorithm known as the sparse matrix inversion algorithm. Use of this algorithm for calculating normalized residuals was proposed by Brousolle [29], and has been used in control centers.

Monticelli and Garcia [27] have proposed a third statistical test, called the $\hat{b}$ test, to be done in conjunction with calculation of normalized residuals. They characterize a gross measurement error at a single measurement as an unknown bias error. The bias error for measurement i is written as $\sigma_{ii} b_i$, thus the term b_i is a normalized bias error. An estimate of the normalized bias error is given by

$$\hat{b}_i = \sigma_{ii} r_i^N / \rho_{ii}. \tag{63}$$

The $\hat{b}$ test consists of checking whether $[\hat{b}_i] > c$, for each of the measurements. As with the normalized residuals test, c is a prespecified threshold whose value can be computed to yield a given false alarm probability for the test. The performance of the $\hat{b}$ test appears to be comparable to that of the normalized residuals test.

When potential bad data have been identified, it is necessary to recompute the state estimate and then redo the bad data detection tests to assure that no further bad data remain. Typically, the measurements with the largest normalized residuals are removed, either one at a time or in groups, and the state estimate is recomputed until the $J(\hat{\boldsymbol{x}})$ and the normalized residuals tests are passed. This procedure may be prohibitively time-consuming due to the need for several state estimate calculations. An alternative procedure that is more efficient has been suggested by Aboytes and Cory [30].

Rather than recompute and refactor the $C(\hat{\boldsymbol{x}})$, Aboytes and Cory proposed correcting the measurement vector in such a way as to suppress the measurement containing the bad data. Suppressing measurement i is equivalent to setting row i of $H(\hat{x})$ to zero, thus the corrected measurement Jacobian matrix is

$$H_c(\hat{x}) = H(\hat{x}) - \mathbf{e}_i \boldsymbol{h}_i \quad (64)$$

where $\boldsymbol{h}_i$ is the ith row of $H(\hat{x})$ and $\mathbf{e}_i$ is an m-dimensional vector whose ith element is a one and whose other elements are zero. An expression for the corrected inverse of the information matrix can be found using the matrix inversion lemma, given by

$$[C_c(\hat{x})]^{-1} = [C(\hat{x})]^{-1} - [C(\hat{x})]^{-1}\boldsymbol{h}_i^T\boldsymbol{h}_i[C(\hat{x})]^{-1}/\rho_{ii}^2. \quad (65)$$

Substituting (64) and (65) into the normal equation gives the following expression for the corrected state estimate:

$$C(\hat{x})\Delta x_c = H(\hat{x})^T R^{-1}[\boldsymbol{z} - \boldsymbol{h}(\hat{x}) + \sigma_{ii}^2 \mathbf{e}_i r_i/\rho_{ii}^2] \quad (66)$$

where $\hat{x}$ is the previously computed value of the state estimate and $\Delta\hat{x}_c$ is the correction to that value.

This correction is a linearized correction but it may be iterated if necessary. The reduction in computations is very significant since refactorization of $C(\hat{x})$ is avoided.

C. Treatment of Multiple Bad Data

Provided that measurement redundancy is adequate, the normalized residuals or the $\hat{b}$ tests for bad data work well when either a single bad datum or noninteracting multiple bad data are present. When interacting bad data are present, it is no longer certain that the largest measurement residual will correspond to a large measurement error nor is it certain that a small measurement residual indicates no bad data for that measurement. Identifying multiple bad data is different and the methods described here have not yet been incorporated into current practice.

Xiang, Wang, and Yu [31] proposed a method for detection and estimation of multiple bad data. Their ideas were expanded by Mili, Van Cutsem, and Ribbens-Pavella in [32], [33]. In [31]-[33] the method of dealing with multiple bad data is to estimate the errors of the suspected measurements from the measurement residuals generated by the initial state estimate. The estimated errors are then used to correct the measurement vector prior to recalculation of the state estimate.

The measurement error estimates are computed based on the relationship between measurement residuals and measurement errors given in (4). Since the rank of W is $m - n$, it is not possible to estimate more than $m - n$ measurement errors on this basis. Furthermore, for the same reason, there are only $m - n$ independent measurement residuals; thus it would not make sense to base the estimates on more than $m - n$ residuals. Suppose that on the basis of some initial screening for bad data, the measurements are partitioned into two classes: 1) the suspected measurements of which there are $s (s \le m - n)$ and 2) the remaining $m - s$ measurements. The suspected measurements are denoted by the subscript s and the remaining measurements by the subscript t.

Suppose that the estimate of the errors is based on $p (s \le p \le m - n)$ of the residuals and that the residuals of the s suspected measurements are included among the p residuals. The subscript p will refer to these residuals and the subscript q will refer to the remaining residuals. With this notation, (5) may be written in partitioned form as

$$\begin{bmatrix} \boldsymbol{r}_p \\ \boldsymbol{r}_q \end{bmatrix} = \begin{bmatrix} W_{ps} & W_{pt} \\ W_{qs} & W_{qt} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_s \\ \boldsymbol{w}_t \end{bmatrix}. \quad (67)$$

The p residuals can thus be written in the form

$$\boldsymbol{r}_p = W_{ps}\boldsymbol{w}_s + \boldsymbol{d}_p \quad (68)$$

where

$$\boldsymbol{d}_p = W_{pt}\boldsymbol{w}_t. \quad (69)$$

Estimation of $\boldsymbol{w}_s$ can be treated as a least square problem with $\boldsymbol{r}_p$ being the "measurement." A least squares estimate for $\boldsymbol{w}_s$ can be calculated by minimizing the quadratic cost function

$$J(\boldsymbol{w}_s) = [\boldsymbol{r}_p - W_{ps}\boldsymbol{w}_s]^T P[\boldsymbol{r}_p - W_{ps}\boldsymbol{w}_s] \quad (70)$$

where P is a positive-definite weighting matrix. In order to be able to compute a unique estimate for $\boldsymbol{w}_s$ it is necessary that W_{ps} have rank s. It can be shown that a necessary and sufficient condition for W_{ps} to be of rank s is that the network remains observable after deletion of the suspected set of measurements. The expression for the resulting least squares estimate, when the rank condition is satisfied, is therefore given by

$$\hat{\boldsymbol{w}}_s = [W_{ps}^T P W_{ps}]^{-1} W_{ps}^T P \boldsymbol{r}_p. \quad (71)$$

The above estimate for $\boldsymbol{w}$ will be optimal in a statistical sense if P is chosen as the inverse of the covariance matrix of $\boldsymbol{d}_p$.

Mili *et al.* advocate the use of the estimate of $\boldsymbol{w}_s$ that results when the residuals used to form $\boldsymbol{r}_p$ coincide with the suspect set. In this case, the expression for the estimate simplifies to

$$\hat{\boldsymbol{w}}_s = W_{ss}^{-1}\boldsymbol{r}_p. \quad (72)$$

As in the single bad datum situation, the measurement vector can be corrected subtracting $\hat{\boldsymbol{w}}_s$ from $\boldsymbol{z}_s$. This has the same effect as explicitly deleting the measurement (on a linearized basis). Instead of doing that, statistical tests could be done on $\hat{\boldsymbol{w}}_s$ to determine whether bad data are present at the suspected measurements.

Monticelli, Wu, and Yen [34] have suggested that the multiple bad data identification problem be set in a decision theory framework. Their formulation takes into account measurement reliability as well as measurement accuracy. In their method, a binary decision tree is constructed and partially traversed in the process of identifying bad data. Strategies for minimizing the number of tree nodes to be examined are proposed.

Other solutions to the bad data problem have been suggested. Merrill and Schweppe [35] had suggested a non-least squares state estimator that automatically damps out the effects of bad data. Kotiuga [36], [37] had proposed the use of linear programming and the L_1 norm to detect bad data. Since nonquadratic state estimators have not been implemented in the real-time environment, most of the research on bad data detection methods has concentrated on those that complement the least squares estimator.

VI. Bus Load Forecast Factors

Since the state estimator executes every few minutes, the ratio of each bus MW load to the system MW load can be tracked over time. This ratio and the power factor at each bus are the bus load forecast factors and they are stored and updated. For each bus, several pairs of these factors can be stored corresponding to the time of day, day of the week, and even month of the year. The number of pairs stored depends on how much they vary over time.

These factors can then be used for forecasting the bus loads for a given system MW load and a given month, day, and time. These bus loads are forecasted mainly for two different purposes. If communication or RTU failure makes a normally state estimated bus unobservable, the forecasted complex load can be used as pseudo-measurements to make the bus observable. The other use is during study mode when the power system studied is for a different season, day, or time. The bus load forecast can automatically specify all the bus loads from a given system MW load.

The pair of factors for each bus for a particular period can be expected to change over a long period of time as the system characteristics change. When the state estimator first starts up, these factors are initialized to average numbers. Then the state estimator continually filters these factors to take into account the latest measurements. For example, the factors for Saturday morning are updated every Saturday morning; usually a very simple digital filtering scheme with an empirically determined parameter used.

VII. External Network Modeling

The external modeling problem is defined with respect to Fig. 4. The internal system is the observable system solved by the state estimator. It is assumed that the unobservable

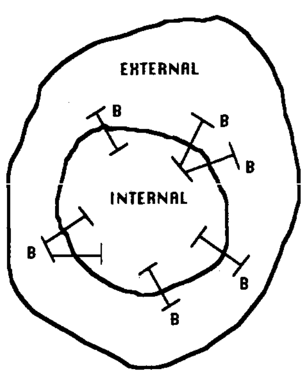

Fig. 4. The internal and external systems.

parts within the jurisdiction of the control center are either lumped into the external system or made observable by using pseudo-measurements. The boundary of the internal system is denoted by the boundary buses (marked *B*), which are the outermost buses to be observable. The boundary buses are the interface buses between the internal and external system. Thus one of the main constraints of the external model solution is that the boundary bus voltages must be the same as those calculated by the internal state estimator.

The external system is the rest of the interconnected network. The problem is how to represent it in real time in the absence of real-time data from this portion of the network. The absence of any real-time data makes this external network completely unobservable. Thus enough data will have to be assumed to obtain a solved model of the external network. The order (or size) of the external network model can also be chosen and the amount of data needed to make this network observable depends on this size.

The various methods used for the external model calculations are reviewed by Wu and Monticelli [38] and Bose [39]. The *external equivalent methods* were used in the early security analysis packages at control centers. In these methods, the external network was reduced off-line to the boundary buses. This reduced network representation had the advantage that it required no assumption of external system data to make the model observable. The reduction could be obtained by one of the many Ward or REI type methods. The variations in load and generation patterns could be accommodated on-line by changing the boundary bus injections to ensure power matching at the boundary. The problem was more difficult when the external network topology changed and the reduced network became an incorrect representation. Although the reduced network could be updated to reflect these topology changes, the occurrences of such status changes in the external network were hard to detect. In addition, since the boundary was fixed, any unobservability developing within the internal system had to be handled with pseudo-measurements as the unobservable areas could not be lumped with the external system.

The more commonly used approach for external modeling today consists of the *external solution methods.* In these methods, a significant amount of the neighboring network is explicitly represented. The status of the equipment and the loads and generation are assumed in real time to make this network observable. The equations of the network are then solved to obtain the real-time external model representation. The assumptions of real-time data are made on reasonability. Breaker statuses are considered normal, external loads are considered proportional to internal load, and generation is considered to be economically dispatched. The solution method can be either the power flow or the state estimation algorithm.

Since the external power system is not of direct interest, the criterion for the adequacy of the external model is not the exact replication of the system. Instead, the external model is considered quite adequate if it can accurately reflect the *effects* of the external system when subsequent analysis is done on the internal system. This analysis may be contingency analysis, any operator power flow study, contingency remedial action, security constrained dispatch, optimal voltage control, or other optimal power flow study.

The main source of errors in the external model is the assumption of real-time data. Since the subsequent functions, such as contingency analysis, study incremental changes to the power system, errors in the status assumptions have more of an effect on this analysis than errors in analog assumptions. That is, in general, it is more important to know the status of important transmission lines and generators than the exact values of the loads and generations. Also, if the errors in the assumptions are electrically closer to the internal system, the effect of the subsequent analysis is more profound. Although nearby status assumptions can cause the highest errors, other sources of errors are the assumptions for transactions between external companies.

Also, for these transaction errors or generation status errors, a secondary source of errors is the assumption of generation pattern.

Bose has shown [40] that the external equivalent methods produce larger errors than the external solution methods. Among the external equivalent methods the REI equivalents seem to produce larger errors than the extended Ward equivalents. Among the external solution methods the more common solution technique is the power flow solution. In practice this method often produces large mismatches at the boundary buses when wrong assumptions are made in the external system. The state estimation solution has been suggested by Geisler and Bose [41] and Geisler and Tripathi [42] to take advantage of the redundancy at the boundary. For wrong assumptions in the external system, the state estimation solution tends to disperse the errors away from the boundary buses and also provides better indications of the location of the wrong assumptions.

Another advantage of using the state estimation solution for the external model is that the same program can be used to solve the internal as well as the external networks. The internal and external systems can even be solved in one pass [43], care being taken that external errors not smear internal results. In practice, however, separate solutions have been used mainly because the external model need not be solved as frequently as the internal.

Obviously, the external modeling solution is an exercise of approximation in the absence of real-time data from the unobservable part of the system. Since the neighboring power systems are often observable to the neighboring control centers, real-time data can be made available through data links between control centers. Although such data links are now being established, the exact data needed to be exchanged for external modeling remains under study. Since a control center does not model its neighbor's system in the same detail as its own, any exchanged data have to be mapped from one model to another. The data can be in a raw or estimated form although the estimated form will have a larger time skew. The problems of handling exchanged data for external modeling is somewhat similar to the hierarchical state estimation problem. However, the present rate of establishment of data links will not alleviate the external modeling problem in the near future and the complete availability of all needed real-time data is not foreseen for even the distant future.

The output of the external model is the combined model of the internal and external systems that is ready for contingency and other analysis. A nontrivial bookkeeping problem in obtaining this final model is the handling of multiple islands. Network islands are identified by the topology processor and each such island must have one reference bus. The state estimator, however, solves for observable islands but once the external model is combined to these observable islands any discrepancies in the phase angles have to be resolved with respect to network island reference buses.

A major task in choosing the external model is the off-line determination of the external model parameters which then become a part of the control center database. In the external solution approach, the external network representation has to be determined. Since the actual external network consists of the whole interconnection outside the internal system, not all of it can be represented in the model. Usually, the external system is divided into three parts (Fig. 5):

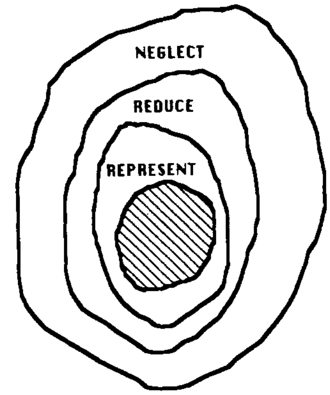

Fig. 5. The three parts of the external system.

1) The portion farthest away that does not have any effect on the internal system need not be represented at all.

2) The portion that is near enough to have some effect on the internal system but can be represented in a reduced form.

3) The portion neighboring the internal system that has significant effects and must be represented explicitly.

The exact boundaries of these three parts have to be determined from off-line studies. At present, these are mainly heuristic studies (Gahagan *et al.* [44]) that determine the sensitivities of the internal network to changes in the external network at various electrical distances. Various boundaries of the three parts are assumed and the sensitivity studies are carried out until, by trial and error, reasonable boundaries are established. Such a study requires a large number of power flow, network reduction, and contingency analysis runs but is crucial for the proper performance of the external model.

Similar studies have to be done for each internal system separately as the external model parameters are very system-specific. For example, an internal system in the middle of a large interconnection may require a much larger external network representation than an internal system on the periphery of the interconnection.

VIII. Penalty Factors

In the economic dispatching of generation, the sensitivity of the transmission losses to the individual generation levels is taken into account by penalizing the incremental cost functions of the generators. These penalty factors are given by

$$PF_i = \frac{1}{1 - (\partial L/\partial P_{gi})} \tag{73}$$

where L is the system loss and P_{gi} is the real power output of the ith generator.

The sensitivities $\partial L/\partial P_{gi}$ depend on the state of the network. Several sets of these sensitivities corresponding to several network conditions can be stored for use by the economic dispatch function. However, with the availability of the real-time network model it is possible to calculate these sensitivities right after the external model calculation is completed. These penalty factors reflect the behavior of the present network conditions more closely than the stored factors and their usage results in more economic production of electricity.

The power balance equation of a power system is

$$\sum_i P_{gi} - D = L \tag{74}$$

where D is load demand. In incremental form (74) can be written as

$$\sum_i \Delta P_{gi} - \Delta D = \Delta L. \tag{75}$$

Now, the sensitivity of the load demand to bus voltage angles can be written as

$$\frac{\partial D}{\partial \theta_i} = \sum_j \frac{\partial D}{\partial P_j} \frac{\partial P_j}{\partial \theta_i} \tag{76}$$

or in matrix form

$$\left[\frac{\partial D}{\partial \theta}\right] = H^T \left[\frac{\partial D}{\partial P}\right] \tag{77}$$

where θ_i is the bus voltage angle at bus i, P_j is the real power injection at bus j, and H is the matrix of elements $[\partial P_j/\partial \theta_i]$.

Solving (77) and defining the resulting vector as α

$$\frac{\partial D}{\partial P} = [H^T]^{-1} \frac{\partial D}{\partial \theta} \triangleq \alpha. \tag{78}$$

Substituting (78) into (75)

$$\sum_i \Delta P_{gi} - \sum_i \alpha_i \Delta P_{gi} = \Delta L \tag{79}$$

resulting in

$$\frac{\partial L}{\partial P_{gi}} = 1 - \alpha_i \quad \text{or} \quad PF_i = \frac{1}{\alpha_i}. \tag{80}$$

Thus the penalty factors are found by solving (78), in which the $\partial D/\partial \theta_i$ terms can be interpreted as the sensitivity of the slack generation in the network solution to the bus voltage angles, and H is one quadrant of the Jacobian matrix. In the actual implementation, the available Jacobian matrix is used and the needed α_i's are extracted. If a decoupled power flow is being used, H can be approximated by the constant B' matrix. Although an iterative procedure is needed to get the exact α_i's, one iteration with B' is thought to give enough accuracy. Also, the assumption of a change in demand being the same as a change in slack power produces an error shift in the α_i's [45] but this does not affect the accuracy of the economic dispatch.

In addition to obtaining the sensitivity of losses to the generators it is sometimes of interest to obtain the sensitivity of losses to interchange. For example, these sensitivities are needed when evaluating the economic desirability of an interchange transaction. One way to do this is to calculate the loss sensitivity to the net interchange of the internal area [46]. Since interchanges of interest are between the internal area company and one of the external area companies, the loss sensitivity of that external area company is sometimes desired. This can be approximated as a weighted average of the sensitivities of the losses to that external company generators.

In addition to the real-time penalty factor for economic dispatching, penalty factors are needed for various other studies. Since these studies can be for different network conditions, penalty factors for several conditions are stored. These stored penalty factors, however, can be automatically updated to reflect the slow changes in real-time penalty factors. This updating is similar to the updating of load distribution factors described in Section VI.

IX. Other Considerations

The very first implementation of these functions occurred in the early 1970s. By 1980 most new control centers were routinely implementing these functions. The capabilities have been expanded over the years and today each function is customized to take into account the special features and needs of individual utilities. Experience with the implementation and operation of these functions over these years has led not only to more efficient algorithms but also to more reliable programs that can withstand the range of conditions in a real-time environment. Despite the sophistication of these functions today, they have not reached their potential usefulness as an operating aid. Operator acceptance has come slowly as the efficiency, reliability, ease of interaction, and operator training have improved.

The usefulness of these functions has also depended on some of the support functions. For example, although much production grade interactive software had already been designed for engineering use, that level of interaction was completely inadequate for the operator. For the control center environment, the man–machine interface had been developed for the SCADA-AGC systems and the interface to these network functions had to be developed in this environment. Fortunately, the man–machine interface has also gone through dramatic improvements in this period incorporating such useful functions as display compilers and full graphics. This has led to the decoupling of the network analysis programs from the way the operator interacts with them. However, the design of the displays and the procedures to get to these displays remain a critical issue on which operator acceptance completely depends. Although training is very important in familiarizing the operator with such new functions, their use will certainly depend on how easily the operator can access the needed information in a timely manner.

Another support function that influences the efficiency and reliability of these programs is the database manager. The addition of the network functions in the control center significantly increased the already large database that was needed to support the SCADA-AGC functions. The descriptions and the parameters of the network are needed in the database to support the new programs. The off-line checking of the network data is more complicated because of all the connectivity checks. More importantly, the mapping of these data into the program data structure has to be designed very carefully to provide the highest computational efficiency. Quite often, the data structure has a greater effect on the program speed than the algorithm used. The unique requirements of a modern control center have resulted in database design and management techniques that are different from other large database handlers. One of these unique requirements is the efficient support of the network programs.

Certain procedures in the implementation of these programs have become standard in the industry. For example, the off-line checkout of the network database is done by a topology processor and power flow program. Although reasonability checks are normally built into the database generators, the ultimate test is the successful running of network solutions on the network under various topology and loading conditions. The database can then be tested on the network programs on simulated SCADA data. The simulated SCADA data can be produced by a power flow with random noise added. The tuning of the state estimator, mainly the R matrix, is also done at this stage. Although the R matrix is supposed to reflect the measurement error variance, it is often tuned to obtain good convergence.

After the network programs and database are tested offline, their implementation in the field remains a major step. The mapping of the actual SCADA data into the input of the topology processor and the state estimator requires the patient shakeout of all the measurements. Similar to the field implementation of the SCADA system, a few measurements, usually from one or two RTUs at a time, are cut into these programs. The bad data detector is used to shake out the measurements. Experience has shown that many gross errors in the measurements cannot be detected by the SCADA system but have to be corrected at this stage before the real-time model is operational. If the state estimator is capable of parameter estimation, any suspicious parameters can also be checked out at this stage. Although bad parameters are often a problem, the actual use of a parameter estimator is not common; instead, more heuristic methods are used. Once the topology processor and state estimator are operational on all the measurements, the rest of the network functions are easier to implement.

X. Conclusions

This paper describes those functions in a power system control center that track the real-time network conditions with a network model. Although the basic principles of estimating the network state from periodic measurements have been known since 1970, the actual implementation has steadily improved in the last fifteen years. The weighted least squares algorithm is still used for state estimation, but the decoupled and orthogonal versions have been developed for better efficiency and accuracy. A tracking network topology that enables the state estimator to use data from the previous cycle may add to the efficiency. Better methods to check for observability and bad data continue to be developed. The best approximations for estimating the external model are starting to be replaced with methods to handle external real-time data over data links. In addition to these better methods, the implementation on different power systems all over the world has resulted in the accumulated experience of adapting these functions to different power system equipment, to different system operational procedures, and to different functional requirements.

References

[1] F. C. Schweppe and E. J. Handschin, "Static state estimation in electric power systems," *Proc. IEEE*, vol. 62, pp. 972–983, July 1974.

[2] A. M. Sasson, S. T. Ehrmann, P. Lynch, and L. S. Van Slyck, "Automatic power system network topology determination," *IEEE Trans. Power App. Syst.*, vol. PAS-92, pp. 610–618, Mar./Apr. 1973.

[3] T. E. DyLiacco, K. A. Ramarao, and A. W. Weiner, "Network status analysis for real-time systems," in *8th PICA Conf. Proc.*, pp. 356–362, June 1973.

[4] M. Prais and A. Bose, "A topology processor that tracks network modifications over time," paper 87SM602-6, presented at the IEEE/PES Summer Meet., San Francisco, CA, July 1987.

[5] J. W. Gu, K. A. Clements, G. R. Krumpholz, and P. W. Davis, "The solution of ill-conditioned power system state estimation problems via the method of Peters and Wilkinson," in *PICA Conf. Proc.*, pp. 239–246, May 1983.

[6] A. Simoes-Costa and V. H. Quintana, "A robust numerical technique for power system state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-100, pp. 691–698, Feb. 1981.

[7] —, "An orthogonal row processing algorithm for power system sequential state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-100, pp. 3791–3800, Aug. 1981.

[8] J. W. Wang and V. H. Quintana, "A decoupled orthogonal row processing algorithm for power state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-103, pp. 2337–2344, Aug. 1984.

[9] A. Monticelli, C. A. F. Murari, and F. F. Wu, "A hybrid state estimator: Solving normal equations by orthogonal transformations," *IEEE Trans. Power App. Syst.*, vol. PAS-105, pp. 3460–3466, Dec. 1985.

[10] F. C. Aschmoneit, N. M. Peterson, and E. C. Adrian, "State estimation with equality constraints," in *10th PICA Conf. Proc.*, pp. 427–430, May 1977.

[11] A. Gjelsvik, S. Aam, and L. Holten, "Hachtel's augmented matrix method—A rapid method improving numerical stability in power system static state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 2987–2993, Nov. 1985.

[12] H. P. Horisberger, J. C. Richard, and C. Rossier, "A fast decoupled static state estimator for electric power systems," *IEEE Trans. Power App. Syst.*, vol. PAS-95, pp. 208–215, Jan./Feb. 1976.

[13] A. Garcia, A. Monticelli, and P. Abreu, "Fast decoupled state estimation and bad data processing," *IEEE Trans. Power App. Syst.*, vol. PAS-98, pp. 1645–1652, Sept. 1979.

[14] J. J. Allemong, L. Radu, and A. M. Sasson, "A fast and reliable state estimation algorithm for AEP's new control center," *IEEE Trans. Power App. Syst.*, vol. PAS-101, pp. 933–944, Apr. 1982.

[15] K. A. Clements and B. F. Wollenberg, "An algorithm for observability determination in power system state estimation," paper A75 447-3, presented at the IEEE/PES Summer Meet., July 1975.

[16] J. S. Horton and R. D. Masiello, "On-line decoupled observability processing," in *10th PICA Conf. Proc.*, pp. 420–426, May 1977.

[17] G. R. Krumpholz, K. A. Clements, and P. W. Davis, "Power system observability: A practical algorithm using network topology," *IEEE Trans. Power App. Syst.*, vol. PAS-99, pp. 1534–1542, July/Aug. 1980.

[18] V. H. Quintana, A. Simoes-Costa, and A. Mandel, "Power system observability using a direct graph-theoretic approach," *IEEE Trans. Power App. Syst.*, vol. PAS-101, pp. 617–626, Mar. 1982.

[19] I. W. Slutsker and J. M. Scudder, "Network observability analysis through measurement Jacobian matrix reduction," presented at the IEEE/PES Winter Meet., Feb. 1986.

[20] A. Monticelli and F. F. Wu, "Network observability: Theory," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 1042–1048, May 1985.

[21] —, "Network observability: Identification of observable islands and measurement placement," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 1035–1041, May 1985.

[22] —, "Observability analysis for orthogonal transformation based state estimation," *IEEE Trans. Power Syst.*, vol. PWRS-1, pp. 201–206, Feb. 1986.

[23] K. A. Clements, G. R. Krumpholz, and P. W. Davis, "Power system state estimation with measurement deficiency: An algorithm that determines the maximal observable subnetwork," *IEEE Trans. Power App. Syst.*, vol. PAS-101, pp. 3044–3052, Sept. 1982.

[24] —, "Power system state estimation with measurement deficiency: An observability/measurement placement algorithm" *IEEE Trans. Power App. Syst.*, vol. PAS-102, pp. 2012–2020, July 1983.

[25] —, "Power system state estimation residual analysis—An

algorithm using network topology," *IEEE Trans. Power App. Syst.*, vol. PAS-100, pp. 1779-1787, Apr. 1981.
[26] K. A. Clements and P. W. Davis, "Multiple bad data detectability and identifiability: A geometric approach," *IEEE Trans. Power Delivery*, vol. PWRD-1, pp. 355-360, July 1986.
[27] A. Monticelli and A. Garcia, "Reliable bad data processing for real time state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-102, pp. 1126-1139, May 1983.
[28] E. Handschin, F. C. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power systems state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-94, pp. 329-337, Mar./Apr. 1975.
[29] F. Broussolle, "State estimation in power systems: Detecting bad data through the sparse inverse matrix method," *IEEE Trans. Power App. Syst.*, vol. PAS-97, pp. 678-682, May 1978.
[30] F. Aboytes and B. J. Cory, "Identification of measurement parameter and configuration errors in static state estimation and bad data processing," in *PICA Conf. Proc.*, May 1975.
[31] N. Xiang, S. Wang, and F. Yu, "A new approach for detection and identification of multiple bad data in power system state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-101, pp. 454-462, Feb. 1982.
[32] L. Mili, Th. Van Custem, and M. Ribbens-Pavella, "Hypothesis testing identification: A new method for bad data analysis in power system state estimation, *IEEE Trans. Power App. Syst.*, vol. PAS-103, pp. 3239-3252, Nov. 1984.
[33] —, "Bad data identification methods in power system state estimation—A comparison study," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 3037-3049, Nov. 1985.
[34] A. Monticelli, F. F. Wu, and M. Yen, "Multiple bad data identification for state estimation by combinatorial optimization," *IEEE Trans. Power Delivery*, vol. PWRD-1, pp. 361-369, July 1986.
[35] H. M. Merrill and F. C. Schweppe, "Bad data suppression in power system static state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-90, pp. 2718-2725, Nov./Dec. 1971.
[36] W. W. Kotiuga and M. Vidyasagar, "Bad data rejection properties of weighted least absolute value techniques applied to static state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-101, pp. 844-853, Apr. 1982.
[37] W. W. Kotiuga, "Development of a least absolute value power system tracking state estimator," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 1160-1167, May 1985.
[38] F. F. Wu and A. Monticelli, "A critical review on external network modeling for on-line security analysis," *Elec. Power Energy Syst.*, vol. 5, 222-235, Oct. 1983.
[39] A. Bose, "Modeling of external networks for on-line security analysis," *IEEE Trans. Power App. Syst.*, vol. PAS-103, pp. 2117-2123, Aug. 1984.
[40] —, "The effects of external modeling errors on on-line security analysis," *IEEE Trans. Power Syst.*, vol. PWRS-1, pp. 227-232, Feb. 1986.
[41] K. I. Geisler and A. Bose, "State estimation based external network solution for on-line security analysis," *IEEE Trans. Power App. Syst.*, vol. PAS-102, pp. 2447-2454, Aug. 1983.
[42] K. I. Geisler and N. K. Tripathi, "State estimation based external network solution: Field testing," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 2126-2132, Aug. 1985.
[43] A. Monticelli and F. F. Wu, "A method that combines internal state estimation and external network modeling," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 91-103, Jan. 1985.
[44] M. J. Gahagan, J. R. Hunt, and A. Bose, "Power system model development for the control center security analysis functions," *IEEE Trans. Power Syst.*, vol. PWRS-1, pp. 308-313, Aug. 1986.
[45] F. L. Alvarado, "Penalty factors from Newton's method," *IEEE Trans. Power App. Syst.*, vol. PAS-97, pp. 2031-2040, Nov./Dec. 1978.
[46] F. Jamshidian, K. C. Liu, and R. Lugtu, "Penalty factor calculations incorporating interchange constraints," *IEEE Trans. Power App. Syst.*, vol. PAS-102, pp. 3667-3671, Nov. 1983.

Anjan Bose (Senior Member, IEEE) was born in Calcutta, India, in 1946. He received the B. Tech. Degree from the Indian Institute of Technology, Kharagpur, India, in 1967, the M.S. degree from the University of California, Berkeley, in 1968, and the Ph.D. degree from Iowa State University, Ames, in 1974, all in electrical engineering.

From 1968 to 1970, he worked for the Consolidated Edison Company, New York, NY, in Power System Planning. He was a Post-Doctoral Fellow at the IBM Scientific Center, Palo Alto, CA, during 1974-1975, and an Assistant Professor of Electrical Engineering at Clarkson University, Potsdam, NY, during 1975-1976. From 1976 to 1981 he was with the Energy Management Division of Control Data Corporation in Minneapolis, MN, where he was in charge of the research and development in power system control and related software. Since 1981, he has been on the faculty of Arizona State University, Tempe, where he is Professor of Electrical Engineering and Director of the Electric Power Research Laboratory. He has served as a consultant to various electric utilities, the Electric Power Research Institute, and other organizations.

Dr. Bose is an active member of the IEEE Power Engineering Society.

Kevin A. Clements (Senior Member, IEEE) received the B.S. degree in electrical engineering from Manhattan College, Bronx, NY, in 1963, and the M.S. and Ph.D. degrees from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1967 and 1970, respectively.

From 1963 to 1968 he was employed by the General Electric Company, Pittsfield, MA. He was at the General Precision division of Singer, Little Falls, NJ, from 1968 to 1970. In 1970, he joined the Electrical Engineering Department of Worcester Polytechnic Institute, Worcester, MA, where he is presently Professor and Chairman. His research interests include monitoring and control of electric power systems, power system stability, and methods for the solution of large-scale systems.