5-31-1991

Bad data identification in power systems

Shichun Zou
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

Title of Thesis: BAD DATA IDENTIFICATION IN POWER SYSTEMS

Shichun Zou, Master of Science in Electrical Engineering, May,1991

Thesis directed by: Dr. W. Hubbi, Associate Professor

Two classes of bad data (BD) identification methods in power system state estimation are studied.

The first class is the perturbation method. In this method, the obtained measurements are perturbed a few times. Each time a state is obtained from which measurement corrections are calculated. The statistics of the corrections corresponding to a BD are different from those corresponding to a healthy data. The method uses those differences for identification purposes.

The effectiveness of different indicators is studied. These include the normalized residual, the weighted residual, and residuals incorporating the statistics of the corrections.

The success rate of using the different indicators is defined and calculated. The tests are done using the IEEE 14-bus and the 30-bus systems. Different measurements configurations are used. The effects of other factors are also tested and results are presented.

The presented results show that the new indicator R_{np} is the most effective indicator. Its success rate is a few percent higher than the widely accepted normalized residuals. The indicator R_{np} is better than normalized residual indicator not only on an average bases, but also in every case individually. The computational requirements for this method is little higher than those of the normalized residual method.

The second class includes three schemes. These are based on the independent equations method in that a sensitivity matrix is defined. The matrix relates the suspected BD to the healthy ones. Scheme I and III are tested using two systems respectively and single and multiple bad data cases are studied. They are successful in identifying bad data.

BAD DATA IDENTIFICATION IN POWER SYSTEMS

by

Shichun Zou

Thesis submitted to the Faculty of the Graduate School of
the New Jersey Institute of Technology
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering

1991

APPROVAL SHEET

Title of Thesis: BAD DATA IDENTIFICATION IN POWER SYSTEMS

Name of Candidate: Shichun Zou
Master of Science in Electrical Engineering, May,1991

Thesis and Abstract Approved: _____ _____

Dr. W. Hubbi — Date
Associate Professor
ECE Department

Signatures of other members of the thesis committee: _____ _____

Dr. B. Friedland — Date
Professor
ECE Department

_____

Y. Bar-Ness — Date
Professor
ECE Department

VITA

Name: Shichun Zou

Permanent address:

Degree and date to be conferred: MSEE, May,1991

Date of birth:

Place of birth:

Secondary education: Wenhau High School, Wahan, 1971-1975.

Collegiate institutions attended	Dates	Degree	Date of Degree
Hauzhong University of Science And Technology, Wuhan, PRC	1978-1982	BSEE	Febrary 1982
New Jersey Institute of Technology	1989-1991	MSEE	MAY 1991

Major: Electrical Engineering

Minor: Computer And Information Science

To My

Father, Mother

and

Sister

for all their support, faith and patience

Acknowledgement

I would like to express my sincere appreciation and gratitude to Dr. Walid Hubbi, my thesis advisor, for his invaluable guidance. Without the advice and time he provided, I would have had no chance to complete this thesis. Special thanks go to Dr. B. Friedland and Dr. Y. Bar-Ness for being members of this thesis committee reading and suggesting modifications to this thesis.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

The objective of this thesis is to develop methods for bad data identification in power systems. Four methods are presented.

In section 1.1, the problem of power system state estimation(PSSE) is presented. The bad data detection and identification problem is introduced in section 1.2. The existing methods of bad data (BD) identification are presented in section 1.3, 1.4 and 1.5 respectively. In section 1.6, the rest of the thesis is outlined.

1.1 Power System State Estimation

This section is descriptive and is taken from $[1-6]$

In power networks, control centers are equipped with a supervisory control and data acquisition (SCADA) system to monitor the network to insure a reliable and optimal operation. The central computer of the SCADA system collects real-time data from the system by the remote terminal units (RTUs). These data are processed by the Power System State Estimation (PSSE) program to provide a best estimate of the state of the system to be used in various application programs. The information transmitted to the control center is inaccurate due to one or more of

the following:

- Power transducer and instrument transformer errors,
- A/D (Analog to Digital) conversion errors,
- Analog or digital data transmission errors,
- Delayed measurements that reflect a prior system state,
- Damaged meters.

The information transmitted to the control center are not form the complete data base of the system because of the following:

- Meters and communication equipment are expensive, so it is necessary to reduce the number of meters as much as possible,
- Some variables, like voltage angles, are difficult and almost impossible to measure economically,
- The unavailable measurements can be calculated using mathematical models.

Because the available measurements contain errors and because the data are not complete, power system state estimation (PSSE) programs are used to process the available measurements to provide the control personnel with a complete, reliable and accurate data base of the system under control. The data transmitted from RTUs are processed based on a mathematical model which assumes the existence of random errors, bad data, modeling errors, and parameter errors. The system variables are calculated (or estimated) using that model.

PSSE then is a data processing algorithm for converting redundant meter readings and other available information, such as the mathematical model of the system, past behavior (or values) of the system variables (known as pseudomeasurements), etc., into an estimate of the state variables. Pseudomeasurements is a term used to indicate values of system variable, obtained from past measurements or estimates of those variables.

The state variables of an electric power system are usually defined as voltage magnitudes and phase angles at all network buses. These are sufficient to define uniquely the state of the system. and, moreover, from which all other values can be explicitly calculated.

Normally PSSE includes the following basic operations:

- Modelling of the system,
- Prefiltering raw data,
- State estimation,
- Detection of bad data,
- Identification of bad data,
- Removal of bad data.

The output of the PSSE program is not the true state of the system. Besides the measurement errors mentioned above, the reasons for the existence of this discrepancy are the following:

- Error in the mathematical model,

- Inaccuracy of system parameters,

- Use of pesudomeasurements.

Pseudomeasurements are old measurements or values of variables calculated using an old state, when these are used in place of measurements. The Weighted Least Squares Method is used to solve PSSE problem. In the WLS method [1 – 4], the static model of an electrical system (or network) is given by its admittance matrix. All measurements (and other information) are modeled in terms of

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \tag{1.1}$$

where $\mathbf{z}$ is a vector of measurements, $\mathbf{x}$ is the state vector of the static structure model, and $\mathbf{h}(\mathbf{x})$ is a nonlinear function of $\mathbf{x}$ which is determined by the admittance matrix and Kirchhoff's laws, relating the real-time measurements to the state vector of the system. The dimension of $\mathbf{z}$ is m, the number of measurements. To take into account the errors in the measurements, the error vector $\mathbf{v}$ is introduced in the above model. Note that unless otherwise stated, boldface uppercase symbols denote matrix quantities and boldface lowercase ones denote vectors.

The estimated state of the system is defined as the value of $\mathbf{x}$ which minimizes the performance index

$$J(\mathbf{x}) = [\mathbf{z} - \mathbf{h}(\mathbf{x})]^T \mathbf{R}^{-1} [\mathbf{z} - \mathbf{h}(\mathbf{x})] \tag{1.2}$$

where the superscript $(^T)$ indicates the transposition of a matrix. $\mathbf{R}^{-1}$ is a diagonal weighting matrix having

$$R_{ii} = {\sigma_i}^2 \tag{1.3}$$

where σ_i is the standard deviation of noise on the ith measurement.

In PSSE, standard deviation of a measurement is not the standard deviation stated in statistics because in the dynamic power system each measurement set represents a snapshot of the system that will never occur again. The characteristics of a meter determines the bounds of ERR and FERR, where, ERR_i and FERR_i indicate the percentage error, and the fixed error of the ith measurement, respectively. The sum of the absolute values of the errors due to ERR_i and FERR_i is taken as the standard deviation of measurement i.

From the above discussion, the weighting matrix usually used in the literature is a diagonal matrix $\mathbf{R}$, where

$$R^{-1}(i,i) = \frac{1}{(|(ERR \times M)_i| + |FERR_i|)^2} \tag{1.4}$$

It should be noted that equation (1.4) is valid only when measurement errors are correlate. If measurement errors are not correlate, the weighting matrix is not a diagonal matrix, it is the inverse covariance matrix of the noise in the measurements [4]. In the study conducted in this thesis, it is assumed that the errors are correlate and thus the weighting matrices used are all diagonal matrices.

It should also be noted that a weighting factor pertaining to a certain measurement is given according to the accuracy of that measurement, but the importance of a measurement is determined not only by its accuracy but also by its location. Thus increasing the accuracy of sensor A , for example, may not have the same effect on the quality of the estimated state, as increasing the accuracy of sensor B.

In order to estimate $\mathbf{x}$, an initial value $\mathbf{x}_0$ is assumed and a Taylor expansion

approximates it near this point:

$$\mathbf{h(x)} = \mathbf{h(x_0)} + \mathbf{h_x(x_0)\Delta x} + \mathbf{h_{xx}(x_0)}\frac{\mathbf{\Delta x}^2}{2} + \cdots$$

Disregarding higher than linear terms, we have

$$\mathbf{h(x)} = \mathbf{h(x_0)} + \mathbf{H\Delta x}$$

where $\mathbf{H}$ indicates the Jacobian (derivative with respective to $\mathbf{x}$ at point $\mathbf{x_0}$).

Substituting $\mathbf{h(x)} = \mathbf{h(x_0)} + \mathbf{H\Delta x}$ into equation (1.2) and letting $\mathbf{\Delta z} = \mathbf{z} - \mathbf{h(x_0)}$ yields

$$J(\mathbf{x}) \;=\; [\mathbf{\Delta z} - \mathbf{H\Delta x}]^T \mathbf{R}^{-1} [\mathbf{\Delta z} - \mathbf{H\Delta x}]$$

The estimate of the state vector $\mathbf{x}$ is obtained by minimizing the performance index

$$\frac{\partial J}{\partial \mathbf{x}} = 0$$

then we have

$$\mathbf{G(x_0)\Delta x} = \mathbf{H}^T \mathbf{R}^{-1} [\mathbf{z} - \mathbf{h(x_0)}] \tag{1.5}$$

$\mathbf{G(x_0)} = \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$ is the gain matrix and $\mathbf{\Delta z} = \mathbf{z} - \mathbf{h(x_0)}$, $\mathbf{\Delta x} = \mathbf{x_1} - \mathbf{x_0}$. This equation is derived in [1].

Equation (1.5) represents a set of linear equations, in which the solution is based on the initial guess $\mathbf{x_0}$. To compute $\mathbf{x}$ to a certain accuracy, equation (1.4) can be rewritten as

$$\mathbf{G}(\mathbf{x}_k)\mathbf{\Delta x_k} = \mathbf{H}^T \mathbf{R}^{-1} [\mathbf{z} - \mathbf{h}(\mathbf{x}_k)] \tag{1.6}$$

$$\mathbf{\Delta x_k} = \mathbf{x}_{k+1} - \mathbf{x}_k \tag{1.7}$$

In the above iterative equations, $k = 0, 1, 2, \ldots$ until the required convergence is achieved.

1.2 Bad Data Detection And Identification

This section is descriptive and is taken from [7, 8, 24, 28, 30]

Occasionally bad data will occur. That is, a data point will be very erroneous rather than just sightly inaccurate. This can occur for a number of reasons. Perhaps there is a momentary failure of a communication link. Perhaps a meter has an intermittent fault. Perhaps, if pseudo-measurements are used, the state of the system has changed and the pseudo-measurements are very far off. Any number of events can occur which will cause a given reading to be very inaccurate[7].

As mentioned before, the presence of bad data has to be detected, which data are bad has to be identified, and finally the bad measurement has to be removed.

It is well known that the residual equation can be obtained by Weighted Least Squares Method of state estimation in power system as follows:

$$\mathbf{r} = \mathbf{W}\mathbf{v} \tag{1.8}$$

and

$$\mathbf{W} = \mathbf{I} - \mathbf{H}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1} \tag{1.9}$$

where $\mathbf{r}$ is an m×1 residual vector of dimension m, which is defined as $\mathbf{r}$=$\mathbf{z}$-$\tilde{\mathbf{z}}$, where $\tilde{\mathbf{z}}$ is the vector of the calculated values of the measured variables. $\mathbf{R}^{-1}$ is a diagonal weighting matrix. $\mathbf{v}$ is the error vector, $\mathbf{H}$ is the Jacobian matrix[28].

The presence of BD is currently detected through one of the variables below:[24]

- The quadratic cost function: $J(\tilde{\mathbf{x}})$=$\mathbf{r}^T\mathbf{R}^{-1}\mathbf{r}$
- The normalized residual vector $\mathbf{r}_n$=$\sqrt{\mathbf{D}^{-1}}\mathbf{r}$ where

$$\mathbf{D} = diag[\mathbf{R} - \mathbf{H}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T] \tag{1.10}$$

- The weighted residual vector $\mathbf{r}_w = \sqrt{\mathbf{R}^{-1}}\mathbf{r}$

The detection of bad data is based on a hypotheses testing with the two hypothesis H_0 and H_1. Where

H_0: no bad data are present;

H_1: bad data are present.

Denoting by P_e the probability of rejecting H_0 when H_0 is actually true (probability of false alarm) and P_d the probability of accepting H_1 when H_1 is true (probability of detection), the hypothesis testing consists in comparing J(x), r_{wi} or r_{ni} with a "detection threshold" λ which depends on P_e. For example, considering the normalized residuals, one is led to :

- Accept H_0 if $r_{ni} < \lambda$, i=1,2,...,m,
- Reject H_0 (and hence accept H_1) otherwise.

The identification techniques available today are classified into three broad classes[24].

- Identification by elimination (IBE),
- Non-quadratic criteria (NQC),
- Hypothesis testing identification (HTI).

The above methods will be introduced in the following sections.

1.3 Identification By Elimination (IBE)

This section is descriptive and is taken from [2, 3, 7, 8, 20, 22, 24, 28]

The methodology that has gained wide acceptance is based on a two step procedure: the quadratic cost function J(x) test to detect whether bad data are present or not, and the weighted residuals $\mathbf{r}_w$ or normalized residuals $\mathbf{r}_n$ identification test to flag which measurements (if any) are affected by gross errors. After the detection of bad data, the usual procedure is to delete the measurement presenting the largest normalized residual and reestimation the system state[22].

The J(x) test is an indirect approach which detects bad data using a statistical test. When there is no bad data the index J(x) follows a Chi-square distribution. The computed value of J(x) is compared with a constant calculated from the Chi-square distribution (detection threshold). If J(x) exceeds this value, bad data is assumed to be present and identification is required. The weighted or normalized residuals vector is then calculated with its elements arranged in descending order of magnitude. Any element whose value exceeds that of a constant obtained from a normal distribution is suspected to be a bad measurement. Theoretically it is possible to remove all these suspected measurements simultaneously. However, experience has shown that it is safer to remove only the measurement corresponding to the largest residual [20]. The reduced measurement vector is subjectd to a further J(x) test. If it fails the test, the measurement corresponding to the the next largest residual is also removed. This process of detection and identification is repeated until all the suspected measurements are removed.

The main advantage of this method is that it is easy to use and simple to implement, since the only computation it needs besides estimation is that of residuals [24]. However, the method suffers from the following drawbacks.

- The computational requirements are high, since it requires a series of reestimation-

detection after each elimination; this may lead to computer time requirements which are too high for an implementation.

- it may lead to a degradation of the measurement configuration and a subsequent drop of the power of the detection test; this in turn may make it impossible to detect the remaining BD.

1.4 Identification By NQC

This section is descriptive and is taken from [7, 8, 20, 30]

This method is based on the observation that the weighted least-squares estimation is sensitive to bad data because it weights larger residue terms much heavier than small terms [7]. An estimator based on a cost function which somehow assigned less weight to large elements in the residuals would be less sensitive to bad data. One of such cost function is:

$$J(x) = \sum_{i=1}^{n} T_i$$

where

$$T_i = \begin{cases} \frac{r_i^2(x)}{\sigma_i^2} & \text{if} \left|\frac{r_i(x)}{a\sigma_i}\right| \leq 1 \\ a^2(4\sqrt{\left|\frac{r_i(x)}{a\sigma_i}\right|} - 3) & \text{if} \left|\frac{r_i(x)}{a\sigma_i}\right| > 1 \end{cases}$$

This function is a combination of a least-squares and minimum-square-root weighting and is called the bad data suppression(BDS) cost function.

The details of this estimator can be found in [7, 8, 20].

The BDS estimator is very similar in form to the least squares estimator. Modification of existing programs is almost trivially easy. Complexity of the BDS and least squares estimators is about the same [24]. It should be emphasized that if all

residuals are "small" (no bad data), the BDS estimator is essentially identical to the least squares estimator, which is the optimum estimator under a wide variety of circumstances.

The main advantage of the NQC method lies in its simplicity. Indeed, on one hand, it can be implemented through a simple transformation of the basic WLS algorithm; on the other hand , the estimation and identification steps are carried out in a single procedure, which avoids successive.

However, the method suffers from the following serious drawbacks [24].

- Strong tendency to slow convergence or even to diverge: the NQC exhibit a slower convergence than the corresponding quadratic criterion. This can be explained as follow:
  - the shape of cost function is more intricate,
  - the rejection of many measurements may lead to numerically unobservable situation, especially in cases of poor local redundancy and/or multiple interacting BD.
- High risk of wrong identification: Schematically, the NQC rely on measurement having small residuals (with respect to) and tend to reject the others. Now, since there is no one-to-one correspondence between large residual and large measurement errors, it may happen that valid measurements are rejected whereas erroneous ones are kept. In such case the estimation is much less reliable than that given by the quadratic estimation without any BD processing.
- No recognition of topologically unidentifiable BD situation: in this case, results are unpredictable. Moreover, the rate of convergence is generally af-

fected, since the NQC tend to reject too many suspected measurements.

1.5 Hypothesis Testing Identification

This section is descriptive and is taken from [24, 25, 30]

The Hypothesis Testing Identification (HTI) is based on the computation of measurement error estimates, on the use of a hypothesis testing and on the definition of decision rules. The HTI method comprises the following steps [25]:

After a standard detection test has shown presence of bad data and measurement residuals have been computed:

- Select the suspected measurements on the basis of their normalized residuals,
- Estimate the corresponding measurement errors using a linear estimation procedure and the knowledge of the residuals,
- Decide whether each measurement is false or valid through a hypothesis testing applied to its estimated error,
- Refine (if necessary) the hypothesis testing procedure,
- Correct the state estimation.

This method result in an one-shot procedure where all erroneous measurements are identified altogether.

The advantages of HTI are as follows: [24]

- The HTI is method generally able to identify all BD within a single step,
- This method is able to identify strongly interacting BD,

- The method treats properly topologically unidentifiable BD.

The disadvantages of this method are :

- There is a risk of poor identification, corresponding to the case where one or several BD are not selected,
- The method requires the computation of the W_{ss} whereas other procedures merely need the diagonal of the W matrix.

1.6 Thesis Outline

The main goal of this thesis is to study bad data identification methods. In Chapter 2, three schemes for bad data identification are presented. Chapter 3 is devoted to studying the perturbation method for BD identification.

Chapter 2

Bad Data Identification Methods

The objective of this chapter is to study a method of bad data identification in power systems. Three new schemes for bad data identification are discussed.

In section 2.1, The independent equations method is introduced. In section 2.2, scheme I for bad data identification is presented, Also, a new sensitivity matrix is developed in this section. Scheme II and scheme III are presented in section 2.3 and 2.4 respectively.

2.1 The Independent Equations Method

This section is descriptive and is taken from [5, 6, 15, 16].

The IEM is based on the node injections as input data. but utilizes independent line flows as redundant information for correction of erroneous or missing data. The method does not make use of nodal voltage measurements. The pertinent equations are:

$$\begin{bmatrix} \Delta \mathbf{P_{pq}} \\ \Delta \mathbf{Q_{pq}} \end{bmatrix} = \begin{bmatrix} \mathbf{JK_{pq\text{-}j}} \end{bmatrix} \begin{bmatrix} \Delta \delta_{\mathbf{j}} \\ \Delta \mathbf{V_j} \end{bmatrix} \tag{2.1}$$

where $\Delta \mathbf{P_{pq}}$ is the vector of changes of the active line flows, $\Delta \mathbf{Q_{pq}}$ is the vector of changes of the reactive line flows. $\Delta \delta_{\mathbf{j}}$ is the vector of changes of voltage angles,

ΔV_j is the vector of changes of voltage magnitudes. JK_{pq-j} is a sensitivity matrix relating incremental changes in line flows to incremental changes in the state variables.

$$\begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} = \begin{bmatrix} JK_{ij} \end{bmatrix} \begin{bmatrix} \Delta \delta_j \\ \Delta V_j \end{bmatrix} \tag{2.2}$$

where JK_{ij} is the Jacobian matrix, ΔP_N is the vector of changes of the active injections, ΔQ_N is the vector of changes of the reactive injections.

The state variables vector is eliminated from (2.1) and (2.2) to get:

$$\begin{bmatrix} \Delta P_{pq} \\ \Delta Q_{pq} \end{bmatrix} = \begin{bmatrix} \text{Spq-i} \end{bmatrix} \begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} \tag{2.3}$$

where $\begin{bmatrix} \text{Spq-i} \end{bmatrix} = \begin{bmatrix} \text{JKpq-j} \end{bmatrix} \begin{bmatrix} JK_{ij} \end{bmatrix}^{-1}$.

The above equation can be partitioned as:

$$\begin{bmatrix} \Delta P_{pq} \\ \Delta Q_{pq} \end{bmatrix} = \begin{bmatrix} \text{Spq, } \mu \end{bmatrix} \begin{bmatrix} \Delta P_\mu \\ \Delta Q_\mu \end{bmatrix} + \begin{bmatrix} \text{Spq-i} \end{bmatrix} \begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} \tag{2.4}$$

or

$$\begin{bmatrix} \Delta P_\mu \\ \Delta Q_\mu \end{bmatrix} = \begin{bmatrix} \text{Spq, } \mu \end{bmatrix}^{-1} \left(\begin{bmatrix} \Delta P_{PQ} \\ \Delta Q_{PQ} \end{bmatrix} - \begin{bmatrix} \text{Spq-i} \end{bmatrix} \begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} \right) \tag{2.5}$$

where $i \neq \mu$ and μ is the index corresponding to the lost or bad nodal input data. Subscript i, j in $[JK_{ij}]$ is to indicate that the Jacobian entries are derivatives of nodal with respect to nodal variables, while the subscript pq-i is to indicate derivatives of line flows with respect to nodal variables.

The algorithm of the method is as follows:

- Perform AC load flow (LF) analysis and calculate line flows.

- By comparing the calculated line flows with the measured ones, determine if BD exist.

- If BD exist, assume the nodal data pertaining to node μ is bad; use dummy values(zero or previously read values) for P_μ and Q_μ to perform an LF and to construct the sensitivity matrix; use (2.5) to solve for ΔP_μ and ΔQ_μ.

- Once the correction of the lost nodal points is obtained, the inputs are corrected according to

$$\mathbf{P}_\mu^{\mathbf{k+1}} = \mathbf{P}_\mu^{\mathbf{k}} + \mathbf{\Delta P}_\mu, \mathbf{Q}_\mu^{\mathbf{k+1}} = \mathbf{Q}_\mu^{\mathbf{k}} + \mathbf{\Delta Q}_\mu$$

to find improved values for P_μ and Q_μ. This step is iterative as indicated by the iteration count k. Once convergence is attained, LF is performed and a decision is made , based on the new value of J, whether node μ is indeed the BD node or not; If not, the process is repeated assuming a different μ.

2.2 Scheme I for Bad Data Identification

In this section, a scheme for bad data identification based on independent equations method is presented. Firstly, a new sensitivity matrix is developed. Its dimensions is based on the number of suspected bad data so that it is a comparatively small matrix, (since the number of suspected bad data is small in practice). Therefore, the difficulty of inverting the sensitivity matrix is grately reduced. Secondly, the existent normalized residuals technique is used to specify a set of suspected bad data. Thus all the bad data can be identified in a single step.

2.2.1 The Scheme

Assume there are s suspected bad data, i.e. PQ_{N1}, PQ_{N2}, ... PQ_{Ns}.
where PQ_{Ni} may stand for P_{Ni}(i-th active injection), Q_{Ni} (i-th reactive injection) or both.

From m-s reliable data, s power flows PQ_{L1}, PQ_{L2},...PQ_{Ls} are chosen as redundant information for corrections, where PQ_{Li} may denote P_{Li} (i-th active load flow), Q_{Li} (i-th reactive load flow) or both.

All injections P_N and Q_N are used as inputs. For lost data, use 0 or previously read values. The buses whose measurements are not avaliable can be treated as lost data, the only drawback for this is that it may increase the dimension of sensitivity matrix M which will be discussed in details later on. According to the above treatment, load flow can be performed since the values of all necessary inputs are now available.

The new sensitivity matrix M is formed as follows:

Perform load flow by using all the injections and find:

$$PQ^0_{L1}, PQ^0_{L2}, ...PQ^0_{Ls}.$$

Give P_{N1} a small incremental change ΔP and keep P_{N2}, P_{N3} ... P_{Ns} unchanged. Then, use $P_{N1}+\Delta P$, P_{N2}, ... P_{Ns} as inputs to perform LF and find:

$$PQ^1_{L1}, PQ^1_{L2}, ...PQ^1_{Ls}.$$

Calculate the mismatch:

$$\Delta PQ_{L1} = PQ^1_{L1} - PQ^0_{L1}$$

$$\Delta PQ_{L2} = PQ^1_{L2} - PQ^0_{L2}$$

$$\ldots$$

$$\Delta PQ_{Ls} = PQ^1_{Ls} - PQ^0_{Ls}$$

Calculate the 1st column of sensitivity matrix M.

$$M_{11} = \frac{\Delta PQ^1_{L1}}{\Delta P}$$

$$M_{21} = \frac{\Delta PQ^1_{L2}}{\Delta P}$$

$$\ldots$$

$$M_{s1} = \frac{\Delta PQ^1_{Ls}}{\Delta P}$$

Similarly, give P_{N2} a small incremental change ΔP and keep P_{N1}, P_{N3} ... P_{Ns} unchanged. Then, use P_{N1}, $P_{N2}+\Delta P$, P_{N3},... P_{Ns} as inputs to perform LF and find:

$$PQ^2_{l1}, PQ^2_{l2}, \ldots PQ^2_{ls}.$$

Calculate the mismatch:

$$\Delta PQ_{L1} = PQ^2_{L1} - PQ^0_{L1}$$

$$\Delta PQ_{L2} = PQ^2_{L2} - PQ^0_{L2}$$

$$\ldots$$

$$\Delta PQ_{Ls} = PQ^2_{Ls} - PQ^0_{Ls}$$

Calculate the 2nd column of sensitivity matrix M.

$$M_{12} = \frac{\Delta PQ^2_{L1}}{\Delta P}$$

$$M_{22} = \frac{\Delta PQ^2_{L2}}{\Delta P}$$

where k is an iteration count.

The load flow is again solved and the new M is again constructed using the corrected inputs. The process is repeated until the mismatches $\mathbf{\Delta PQ}_{\mathbf{N}}^{\mathbf{s}}$ are less than a predetermined tolerance. Then the corrected values of all the suspected bad data are obtained. The nodal point identification concept is based on comparing the corrected values with the corresponding measurements. If the difference between the corrected values and the measurements exceeds a predetermined tolerance δ, the bad data points are identified. i.e. the measurements presenting the difference exceeding δ are bad data, but the others are not bad data.

It must be pointed out that the sensitivity matrix M can be repeatedly used since the elements of M does not change much from the iteration to the next.

The check can be made by using s corrected injections and other injections as inputs, perform LF to find $PQ_{L,s+1}$, $PQ_{L,s+2}$, ...$PQ_{L,s+m}$. If the differences between the calculated line flows, from s+1 to m, and their corresponding measured values ($PQ_{L,s+1}$, $PQ_{L,s+2}$, ...$PQ_{L,m}$ are the m-s assumed reliable data, so that PQ_{L1}, PQ_{L2}, ...PQ_{Ls} are not included.) are less than a δ, then the above results are correct.

2.2.2 Tests and Results

To test scheme I, the well known 5-bus system of [62] is used. The cases studied here include single and multiple bad data. The simulation of measurements is done by introducing a 2% error to the true values.

$$M_{s2} = \frac{\Delta PQ_{Ls}^2}{\Delta P}$$

The remaining columns of M can be obtained by the same way. Then we have:

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} & \cdots & M_{1s} \\ M_{21} & M_{22} & M_{23} & \cdots & M_{2s} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ M_{s1} & M_{s2} & M_{s3} & \cdots & M_{ss} \end{bmatrix}$$

where [M] is sensitivity matrix which relates the changes of injections to the changes of line flows.

From the definition of [M], the following equation can be established:

$$\left[\mathbf{M} \right] \left[\mathbf{\Delta PQ_N^s} \right] = \left[\mathbf{\Delta PQ_L^c} \right] \tag{2.6}$$

where M is an s×s square matrix, $\mathbf{\Delta PQ_N^s}$ is the the vector $[\Delta PQ_{N1}, \Delta PQ_{N2}, ...\Delta PQ_{Ns}]^T$ which are corrections for the suspected bad data, and $\mathbf{\Delta PQ_L^c}$ is the the vector $[\Delta PQ_{L1}, \Delta PQ_{L2}, ...\Delta PQ_{Ls}]^T$ which are redundant independent line flows for corrections of bad data.

It must be noted that $\mathbf{\Delta PQ_L^c}$=$\mathbf{\Delta PQ_L^m}$- $\mathbf{\Delta PQ_L^0}$, where $\mathbf{\Delta PQ_L^m}$ are the measurements of the line flows, $\mathbf{\Delta PQ_L^0}$ are the calculated line flows without changing any inputs.

(2.6) can be written as follows:

$$\left[\mathbf{\Delta PQ_N^s} \right] = \left[\mathbf{M} \right]^{-1} \left[\mathbf{\Delta PQ_L^c} \right]$$

Once the corrections of the suspected bad data nodal points are obtained, the inputs are corrected according to

$$\mathbf{PQ_N^{k+1}} = \mathbf{PQ_N^k} + \mathbf{\Delta PQ_N^s}$$

case #	*BD*	*True Value*	*MFCLF*	*Corrected value*	*ERR-T*
1	P_{N2}=20	P_{N2}=40	P_{L12}	P_{N2}=40.83	0.21%
1	P_{N3}=25	P_{N2}=45	P_{L34}	P_{N2}=45.87	1.91%
2	Q_{N2}=5	Q_{N2}=25	Q_{L12}	Q_{N2}=25.10	3.92%
2	Q_{N3}=10	Q_{N3}=15	Q_{L34}	P_{N2}=14.99	0.006%
3	P_{N2}=20	P_{N2}=40	P_{L12}	P_{N2}=38.31	1.69%
3	Q_{N2}=15	Q_{N2}=30	Q_{L12}	Q_{N2}=30.13	0.42%
4	Q_{N4}=0	Q_{N4}=5	Q_{L43}	Q_{N4}=5.337	6.74%
4	Q_{N5}=5	Q_{N5}=10	Q_{L52}	Q_{N5}=10.32	3.22%
5	P_{N2}=20	P_{N2}=40	P_{L12}	P_{N2}=37.38	6.56%
5	P_{N3}=0	P_{N3}=45	P_{L34}	P_{N3}=44.53	1.04%

Table 2.1: The second column is the simulated value of bad data, the third column gives the true values, the forth column is the measurements used for corrections, the fifth column is the corrected values and the last column is the percentage error of the corrected data with respect to its true value.

The results obtained after one iteration only, are shown in Table 2.1. For case 1, two bad data are simulated and both are active injections at nodes 2 and 3. Their true values are 40 and 45 respectively. The measurements used for corrections are line flow P_{l2} and P_{l3}. The considerations for selecting line flows are as follows: Firstly, in power system the relationship between active power and reactive is very weak, in other words, the changes of the active power are not sensitive to the changes of the reactive power. Secondly, variables in the lines directly connected to the bad data points are more sensitive to changes in those bad data. After the first iteration, the corrected values corresponding to bad data are found. The percentage errors of the corrected values with respect to their true values are 2.1 % and 1.91 %. In case 2, two reactive bad data are tested. Case 3 tests one bad active injection and one bad reactive injection. Case 4 and 5 test the case of lost data.

In general, the results are satisfactory and convergence is fast.

2.2.3 Conclusion

The proposed new scheme first specify the set of suspected bad data based on a lower detection threshold. This scheme allows the identification of bad data in one single procedure. In contrast, the conventional IEM needs exhausted computation even for a single bad data. Let us take 100-bus system for example, the conventional IEM assumes the bad data is at node 1 first, once the convergence is obtained, a decision on node 1 can be made. If node 1 is not the bad data point, the next node is assumed, this process is repeated until the bad data is found. If the bad data is the last node, then the procedure must be repeated 100 times. For multiple bad data, it is even worse since various possible combinations have to be calculated.

The conventional IEM needs to invert a high order (2×N, 2×N) matrix (where N is the number of nodes) and a low order (S, S) (where S is the number of the suspected bad data, which is usually small) inversion. However, the new scheme only needs to calculate the low order inversion since the new sensitivity matrix M is constructed. Furthermore, difficulties may be encountered in finding the inversion of the high order (2×N, 2×N) matrix.

2.3 The Second Scheme

The methods presented in section 2.1 and 2.2 have a fatal drawback, i.e. the method can only be used to identify bad data in injections, also, the measurements used as correction information are limited to line flows. However, in practice, the measurements can be any combinations of injections, line flows and magnitudes of bus voltages. Therefore, the bad data might be any possible combinations of those measurements. All these factors limit the use of the above methods and render them

impractical.

To solve this problem, the power system state estimation(PSSE) based on the weighted least square method is used for detection and identification in the proposed scheme II.

2.3.1 The Scheme

The measurements used in this new scheme are active injections P_N, reactive injections Q_N, active line flows P_L, reactive line flows Q_L, and magnitude of voltages V_N.

All the measurements are divided into two sets by a low normalized residuals detection threshold. The first set that has high residuals is the suspected set. The other set is called the correction set. As mentioned in [24] and will discussed in greater detail in the next chapter, the normalized residuals test is not reliable enough, particularly for multiple bad data cases. Therefore, a low detection threshold used here can ensure that all bad data are included in the suspected set.

Assume that the number of the suspected bad data is s. The pertinent equations are:

$$\begin{bmatrix} \mathbf{\Delta P_N^s} \\ \mathbf{\Delta Q_N^s} \\ \mathbf{\Delta P_L^s} \\ \mathbf{\Delta Q_L^s} \\ \mathbf{\Delta V^s} \end{bmatrix} = \begin{bmatrix} \mathbf{JS} \end{bmatrix} \begin{bmatrix} \mathbf{\Delta \delta_2} \\ \mathbf{\Delta \delta_3} \\ \mathbf{\Delta \delta_{2n-1}} \\ \mathbf{\Delta V_1} \\ \mathbf{\Delta V_{2n-1}} \end{bmatrix} \tag{2.8}$$

where **JS** is the Jacobian matrix. $\mathbf{\Delta P_N^s}$, $\mathbf{\Delta Q_N^s}$, $\mathbf{\Delta P_L^s}$, $\mathbf{\Delta Q_L^s}$, $\mathbf{\Delta V^s}$ are the changes of active and reactive injections, active and reactive of line flows, and magnitudes of voltages respectively. Superscript s indicates that they are temporarily suspected to be bad data. Since bus 1 is chosen as slack bus, i.e. $\Delta\delta_1$ is assumed to be 0.

Therefore, the number of the state variables is 2N-1, here N is the number of nodes of the system.

Equation (2.8) can be written as follows:

$$\left[\Delta PQV^s \right] = \left[JS \right] \left[\Delta X \right] \tag{2.9}$$

From m-s reliable data, the following equations can be established:

$$\begin{bmatrix} \Delta P_N^c \\ \Delta Q_N^c \\ \Delta P_L^c \\ \Delta Q_L^c \\ \Delta V_N^c \end{bmatrix} = \left[JC \right] \begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta\delta_{2n-1} \\ \Delta V_1 \\ \Delta V_{2n-1} \end{bmatrix} \tag{2.10}$$

where JC is the Jacobian matrix. ΔP_N^c, ΔQ_N^c,ΔP_L^c,ΔQ_L^c, ΔV_N^c are changes of active and reactive injections, active and reactive of line flows, and magnitudes of voltages respectively. Superscript c indicates that those measurements will be used to corrected the suspected set.

Equation (2.10) can be written as follows:

$$\left[\Delta PQV^c \right] = \left[JC \right] \left[\Delta X \right] \tag{2.11}$$

The state variables vector is eliminated from (2.9) and (2.11), similarly to what is done in the IEM, to obtain:

$$\left[\Delta PQV^s \right] = \left[JS \right] \left[JC \right]^{-1} \left[\Delta PQV_c \right] \tag{2.12}$$

or

$$\left[\Delta PQV^s \right] = \left[JSC \right] \left[\Delta PQV^c \right] \tag{2.13}$$

where

$$\left[\ \mathbf{JSC}\ \right] = \left[\ \mathbf{JS}\ \right]\left[\ \mathbf{JC}\ \right]^{-1}$$

The algorithm of the method is as follows:

- Perform state estimation (PSSE).
- Calculate performance index J($\mathbf{X}$) to determine if BD exist.
- If BD exist, calculate normalized residuals and divide all the measurements into two parts based on a low detection threshold.
- Assume that the measurements with their normalized residuals values higher than the detection threshold are suspected to be bad data, $\mathbf{PQV^s}$, and that the measurements with their normalized residuals lower than the detection threshold are used as correction information , $\mathbf{PQV^c}$.
- Perform PSSE and construct the sensitivity matrix $\mathbf{JSC}$; use (2.13) to solve for $\mathbf{\Delta PQV^s}$.
- Once the corrections of the suspected bad data points are obtained, the suspected bad data $\mathbf{PQV^s}$ are corrected according to

$$\mathbf{PQV^{s,k+1}} = \mathbf{PQV^{s,k}} + \mathbf{\Delta PQV^s}$$

to find improved values for $\mathbf{PQV^s}$. This step is iterative as indicated by the iteration count k. Once convergence is obtained, a decision is made based on comparing the measurements with the corresponding calculated values. Only the items of $\mathbf{PQV^s}$ which are larger than a predetermined tolerance are indeed the bad data, others are not bad data.

2.3.2 Summary

This scheme uses the idea of the independent equations method to construct a sensitivity matrix relating the changes of suspected bad data $\mathbf{\Delta PQV^s}$ to the changes of a set of reliable data $\mathbf{\Delta PQV^c}$ in PSSE. This scheme overcomes the limitations of the IEM. This scheme also is thought to be more reliable than the identification by elimination (IBE) method.

The IBE which "has received wide acceptance is based on a two step procedure: the J(x) test to detect whether bad data are present or not, and the weighted or normalized residuals identification test to flag which measurements (if any) are affected by gross errors," [22]. However, "J(x) test conventionally used in power system static state estimation has poor reliability for detecting the presence of measurement errors in the range 3 to 20 standard deviations," [22]. "r_n test is not reliable enough either, ... since there is no one to one correspondence between larger residual magnitudes and larger measurement errors, since each residual is a linear combination of the various errors and the detection is taken on a global basis, and is able to give a rough indication," [25]. This new scheme first specify the scope of bad data based on a low detection threshold. The value of detection threshold in this scheme can be taken much lower than usual since the iterative correction procedure used in this scheme can re-identify all the suspected bad data, therefore the probability of bad data to be detected is expected to be higher than the method of identification by elimination (IBE).

The new scheme can identify bad data in one single procedure, i.e. all the bad data will be identified at the same time. However, the conventional IEM needs exhaustive computation even for a single bad data. The computational requirement

of this scheme is much lower than that of identification by elimination (IBE) for multiple bad data.

The new scheme is simple and an existent state estimation program can readily be converted to the this scheme.

2.4 Further Improvement (The third scheme)

One might think that if the measurements with the largest 10 or 20 residuals are eliminated (since the number of the bad data is usually very small, all the bad data will be included), then PSSE is performed to obtain a good state of the system. After that all the variables are calculated based on the good state and the bad data are identified by comparing the measurements with their calculated values. It seems a good scheme if it does not cause the system to be unobservable. Unfortunately, this may happen.

Although the new scheme in section 2.3 has some advantages over other methods, it suffers a drawback as IBE and IEM do, i.e. it might cause configuration degradation since only a set of measurements is used as correction information. In other words, it may cause the system to be unobservable.

In order to avoid any possible configuration degradation, a further improvement is made on scheme II. In scheme II, $\mathbf{\Delta PQV^c}$ is a vector with 2n-1 measurements which is used to correct bad data. Here $\mathbf{\Delta PQV^c}$ will be the vector of all the available measurements. This means that there is no changes in the measurement configuration during the identification procedure. Thus scheme III ensures that there is no configuration degradation.

The algorithm of the method then is as follows:

- Perform state estimation (PSSE).
- Calculate performance index J(X) to determine if BD exist.
- If BD exist, calculate normalized residuals and divide all the measurements into two sets based on a low detection threshold.
- Assume that the measurements with their normalized residual values higher than the detection threshold are suspected to be bad data $\mathbf{PQV}^{\mathbf{s}}$, and use all the measurements (including the s suspected bad data) as correction information $\mathbf{PQV}^{\mathbf{c}}$.
- Perform a PSSE (this is not needed in the first iteration, since it has been done in the first step above) to get $\mathbf{\Delta PQV}^{\mathbf{s}}$.
- Once the corrections of the suspected bad data points are obtained, the suspected bad data $\mathbf{PQV}^{\mathbf{s}}$ are corrected according to

$$\mathbf{PQV}^{\mathbf{s,k+1}} = \mathbf{PQV}^{\mathbf{s,k}} + \mathbf{\Delta PQV}^{\mathbf{s}}$$

  to find improved values for $\mathbf{PQV}^{\mathbf{s}}$. This step is iterative as indicated by the iteration count k. Once convergence is obtained, a decision is made based on comparing the measurements with their corresponding calculated values.

2.4.1 Tests and results

The scheme have been tested on IEEE 14 bus test system. The overall measurement redundancy is about 2.3. The standard deviation of the error are set equal to 0.02 p.u. (on a 100 MVA base) for real and reactive power measurements and 0.002

p.u. for voltage magnitude measurements. The bad data is simulated by injecting 20 standard deviation to the true values[22]. The cases selected to report include single and multiple bad data. To test this scheme, a program is developed which is based on a PSSE program. The flowchart of the program is presented in Fig.2.1. More details about measurements' simulations, considerations of redundancy, the size of bad data, and the program itself, will be discussed in the next chapter.

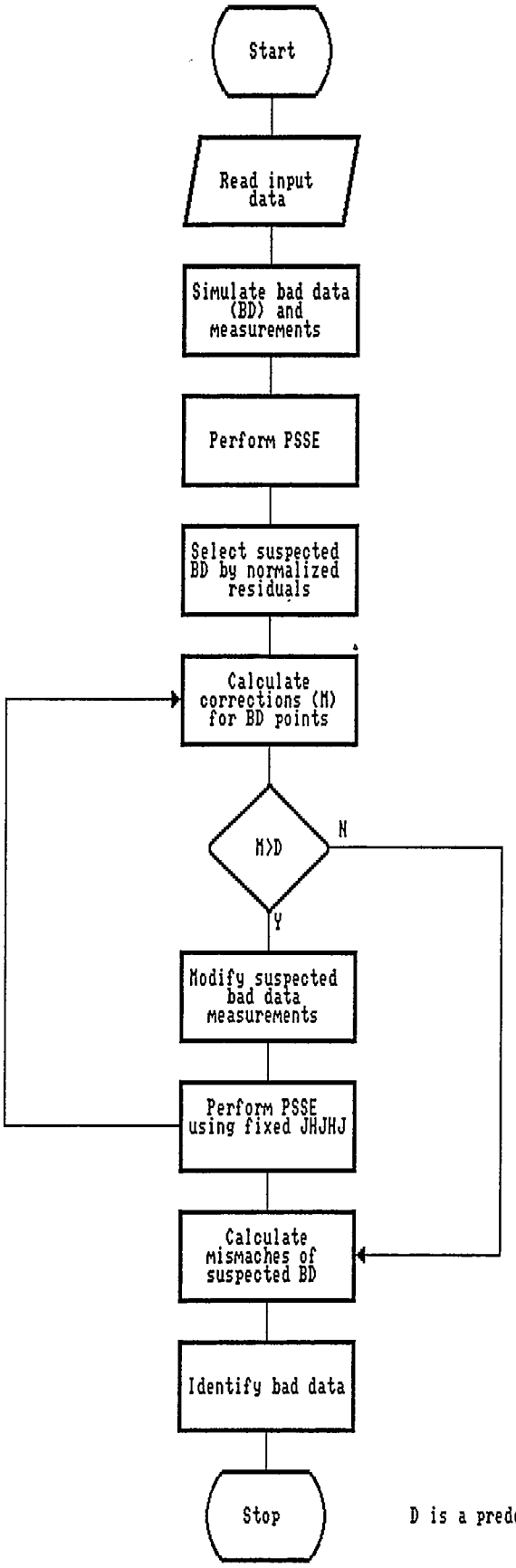

Fig.2.1 Flowchart outlining of scheme III

BD	*ID*	*ERR T*	*TMOIT*
1	Y	46.2%	6
2	Y	6.6%	3
3	Y	2.3%	7
4	Y	1.4%	3
5	Y	2.8%	2
6	Y	2.5%	2
7	Y	20.4%	19
8	Y	1.3%	3
9	Y	3.1%	24
10	Y	2.17%	18
11	Y	1.13%	3
12	Y	1.14%	2
13	Y	3.3%	20
14	Y	1.71%	3
15	Y	1.19%	2
16	Y	1.73%	12
17	Y	5.58%	4
18	Y	112.1%	23
19	Y	3.4%	5
20	Y	2.12%	1
21	Y	0.75%	8
22	Y	26.4%	7
23	Y	2.2%	2
24	Y	21.04%	21
25	N	–	0
26	Y	0.95%	5
27	Y	1.34%	4
28	Y	45.4%	6
29	Y	6.19%	2
30	Y	9.04%	3

Table 3.1: The results obtained by using IEEE-14 bus system configuration B. The first column is the bad data locations. The second column tells whether the bad data is identified. the third column is the percentage error of the corrected values with respect to ture values, the last column is the number of the iterations.

BD	*ID*	*ERR T*	*TMOIT*
31	Y	8.23%	6
32	Y	1.18%	2
33	Y	1.9%	9
34	Y	4.89%	4
35	Y	15.38%	1
36	Y	15.26%	2
37	Y	5.44%	3
38	Y	12.42%	1
39	Y	42.28%	1
40	Y	33.56%	2
41	Y	1.47%	2
42	Y	9.25%	3
43	Y	11.04%	4
44	Y	38.71%	1
45	Y	11.26%	7
46	Y	13.67%	13
47	Y*	1.06%	1
48	Y	19.33%	6
49	N	156.59%	0
50	Y	4.05%	11
51	Y	1.28%	2
52	Y	21.42%	3
53	Y	0.42%	2
54	Y	12.28%	7
55	Y	19.89%	5
56	Y	7.58%	2
57	Y	0.29%	2
58	Y	0.58%	2
59	Y	0.207%	2
60	Y	4.48%	2

Table 3.2: The results obtained by using IEEE-14 bus system configuration B. The first column is the bad data locations. The second column tells whether the bad data is identified. the third column is the percentage error of the corrected values with respect to ture values, the last column is the number of the iterations.

MULTBD	*ID*	*ERR T*	*TMOIT*
P_{N6}	Y	6.1%	5
Q_{N10}	Y	7.8%	5
$P_{L9}3$	Y	1.7%	5
$Q_{L13}4$	Y	7.1%	5
V_3	Y	0.2%	5

Table 3.3: The results obtained by using IEEE-14 bus system configuration B. The first column is the multiple bad data locations. The second column tells whether the bad data is identified. the third column is the percentage error of the corrected values with respect to ture values, the last column is the number of the iterations.

MULTBD	*ID*	*ERR T*	*TMOIT*
P_{N10}	Y	7.3%	4
Q_{N6}	Y	4.4%	4
$P_{L11}3$	Y	9.2%	4
$Q_{L16}4$	Y	2.7%	4
V_6	Y	0.15%	4

Table 3.4: The results obtained by using IEEE-14 bus system configuration B. The first column is the multiple bad data locations. The second column tells whether the bad data is identified. the third column is the percentage error of the corrected values with respect to ture values, the last column is the number of the iterations.

The above results show that this scheme can identify bad data with a success rate of 97%. However, the success rate when the normalized residuals are used is 87%. The normalized residuals will be discussed in greater details in the next chapter. For a single bad data case, the computational requirement is higher than the normalized residuals method. The examples show that 5 bad data can be identified in 4 and 5 iterations respectively. However, the IBE needs at least 5 times as many iterations.

2.4.2 Conclusion

Scheme III is derived from scheme II. The sensitivity matrix in scheme III relates the changes of suspected bad data $\mathbf{\Delta PQV^s}$ to the changes of all the available measurements $\mathbf{\Delta PQV^c}$ in PSSE. This scheme will not cause any configuration degradation. The advantages of scheme II mentioned in section 2.3.2 are enhanced. The IEEE 14-bus system was used to test the scheme using single and multiple BD.

It should be mentioned here, however, that an apparently similar scheme was referred to in [24].

Chapter 3

The Study of Perturbation Method for Bad Data Identification

This chapter is devoted to the study of the Perturbation Method (PM) for bad data identification.

Section 3.1 is taken form [16] as a summary of the previous work done on the PM for bad data identification. In section 3.2, the PM in PSSE is generalized to overcome the restrictions imposed in the first introduction of the method in [16]. Also, the main program is introduced in this section. In section 3.3, test conditions are discussed, the normalized residuals method is studied in this section. In section 3.4, several indicators are defined and studied; testing results are presented and analysed. In section 3.5, effects of varying some factors on the results are studied. Conclusions appear in the last section.

3.1 Perturbation Method For Bad Data Identification

This section is a summary of the method and is taken from [16].

In 1989, a method for bad data (BD) identification in power system state estimation was proposed in [16]. This method is referred to in this thesis as the Perturbation Method (PM). The data assumed available in this method are similar to those assumed by the independent equations method which is introduced in section 2.1. This data structure is too restrictive and not practical. The purpose of this chapter is to find ways to remove these restrictions. This method is summarized as follows:

Equation 2.1 is repeated here for convenience.

$$\begin{bmatrix} \Delta P_{pq} \\ \Delta Q_{pq} \end{bmatrix} = \begin{bmatrix} \text{Spq-i} \end{bmatrix} \begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} \tag{3.1}$$

where $\begin{bmatrix} \text{Spq-i} \end{bmatrix} = \begin{bmatrix} \text{JKpq-j} \end{bmatrix} \begin{bmatrix} \text{JK}_{ij} \end{bmatrix}^{-1}$.

Decoupling is used, i.e., the steps to be followed to identify bad P-injection data, are paralleled in case the bad data are Q-injection . Therefore, the procedure will be illustrated for the P case only.

$$\begin{bmatrix} \Delta P_{node} \end{bmatrix} = \begin{bmatrix} \text{SENP} \end{bmatrix} \begin{bmatrix} \Delta P_{line} \end{bmatrix} \tag{3.2}$$

where ΔP_{line} is simply the difference between the measured line power flow at the metered points and the corresponding calculated values. **SENP** is a sensitivity matrix relating ΔP_{line} and ΔP_{node}; it is the inverse of a submatrix of (3.1).

The method of [16] is based on the following observation: if small random errors are injected into the nodal measurements and ΔP_{node} calculated, then the entry corresponding to a BD will not change as significantly as the entries corresponding to the healthy data. The procedure then is to slightly perturb the nodal measurements a few times (say five times) and each time calculate ΔP_{node}. Thus, a set of five

vectors of $\mathbf{\Delta P_{node}}$ is created. From this set, the following vectors are calculated:

- AVE is an average vector, averaging the five $\mathbf{\Delta P_{node}}$,
- σ is standard deviation vector, σ_i= the standard deviation of the five $\mathbf{\Delta P_{node}}$,
- σ/AVE : the i-th entry of which is σ_i/AVE_i.

It is reported that, in almost all cases, if the measurement of the injected power at node m is bad, then the absolute value of $\sigma_m/(AVE)_m$ is the smallest. This is found to be true even for multiple BD.

The tests for this method are conducted on the IEEE 14-bus system. The method can identify single and multiple bad P injections, with a success rate exceeding 90%. It is not as successful in identifying bad Q injections; this may be due to a programing error.

3.2 Generalizing the Perturbation Method

The original PM method is based on the IEM (see section 2.1). Due to the IEM model used, the method can only identify bad injections. Also, [16] has not explained why it can identify bad data and the comparison between the method with other existent methods has not been done.

It is hoped to extend the applicability of the method to identify not only bad data among injections, but also the bad data among line flows and voltages. Also, the extended method should be more thoroughly tested.

The following sections try to answer the following questions:

- Can the restrictions on the measurements topology be removed?

- How better or how worse does it work compared to other existing methods?

3.2.1 The Extended Perturbation Method for Bad Data Identification in PSSE

As mentioned in the last section, in order to test the method more extensively and to enhance the function of the method, all tests should be conducted in PSSE environment. PSSE environment means that no restrictions are placed on the measurement set beyond what is found in practice.

The algorithm of the perturbation method based on a PSSE model is as follows:

(1): Specify the number of perturbations, N,

(2): Simulate injection, line flow and voltage measurements,

(3): Simulate bad data,

(4): Generate the weighting matrix,

(5): Perform state estimation (PSSE),

(6): Calculate the residuals (mismatches) $\mathbf{\Delta PQV}$,

(7): If the count number K is equal to N go to (8), otherwise, perturb all the input measurements, i.e., inject small random increments in all the measurements (say 0.5 % of the measurement value), then go to step 5,

(8): Calculate the average of the obtained set of $\mathbf{\Delta PQV}$ and its standard deviation,

(9): Calculate σ/AVE and other possible indicators to be discussed later on.

A program for this purpose is developed and will be explained as follows.

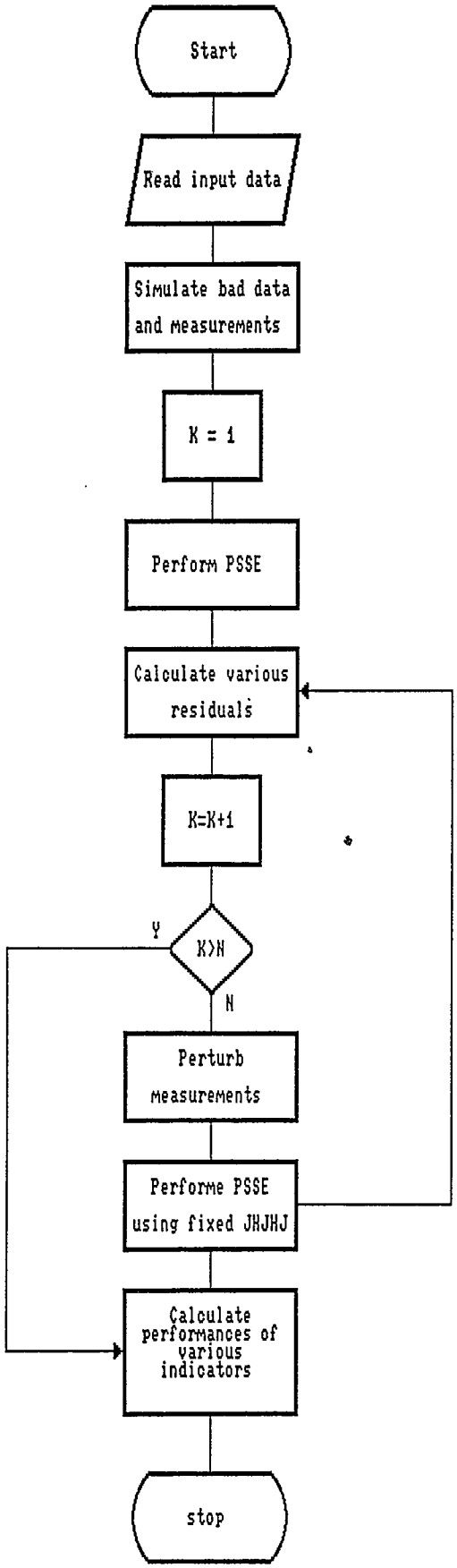

Fig.3.0 Flowchart outlining of the proposed method

3.2.2 Explanation of the Program Flowchart

Figure 2.1 shows the flow chart of the program used. The following explanatory notes are numbered to correspond with the block numbers in the flow chart they pertain to:

(2): Subroutine 'SYSDATA' is used to read system parameters along with other data. The input data files for the IEEE 14-bus test system and the IEEE 30-bus test system are given in [62],

(3): In this block, subroutine 'FRANK' is called to perform Load flow calculation in order to establish the true values of the state variables; subroutines 'PQCAL', 'PQLCAL' and 'CLF' are used to calculate true values of all the system variables including: real and reactive power injections, flows, and nodal voltages; subroutine 'MGB' is called to simulate measurements and bad data,

(5): Subroutine 'PSSE' is called to estimate system states,

(6): Call Subroutine 'RESLFAT' to calculate the various residuals,

(9): Subroutine 'MGBP' is called to perturb measurements,

(10): Subroutine 'PSSEP' is called to estimate system states,

(11): Call subroutine 'RESLIDC' to calculate the various indicators for identification.

The developed program also includes the following subroutines:

MGB, MGBP, WMGP, RESFCT, RESLC, SORTGREAT, PSSE, PSSEP, SYSDATA, PQLCAL, EDELTA, TEST, STAM, PQCAL, CLF, DELTA, TRAN, XRR, INVERM, FACTO, MG, WMG, UTION, FRANK and RANDOM. The first 8 subroutines, together with the main program, are presented in Appendix A. The rest of the subroutines are not listed because they can be found in [17].

3.3 Test Conditions

To test the perturbation method and for the purpose of measurement generation and result comparison, all true values are assumed to be known. The well known IEEE 14-bus test system and the IEEE 30-bus test system are used. Most tests are conducted on the IEEE 14-bus test system and some of them are made on the IEEE 30-bus test system.

Before presenting the tests results, the following will be discussed :

- How to evaluate the method?
- How to generate simulated bad measurements?
- How does the normalized residual method perform?

3.3.1 How to Evaluate the Perturbation Method

To judge a method, two methods can be used: The first is analytical; the second is to use a Monte Carlo technique. In our studies, the latter is used mainly.

According to [16], the indication for bad data identification is σ/AVE, where AVE is an average vector, averaging the five ΔP (assuming the number of perturbation is five), where ΔP is the plain residual (mismatch between calculated values and measurements and it is obtained after convergence is reached). In the literature, three residuals are used: plain residual, weighted residual and normalized residual. Therefore, three indicators can be constructed. These are: σ/R_p, σ/R_w and σ/R_n.

For each indicator, a vector is constructed; the smallest entry in that vector is hoped to indicate the bad data location.

In this study, these three indicators along with other possible indicators are discussed and tested.

The evaluation of any of the identification indicators is based mainly on its success rate. In order to clarify the sucess rate concept, the normalized residuals are used to illustrate how tests are conducted in this chapter.

For an instrumentation configuration, an overall redundancy is assumed and each time an error is introduced to a measurement (for single bad data case) to form the bad data, then perform PSSE and calculate normalized residuals of all the measurements after convergence is attained. All the residuals then are rearranged in a descending order. The largest normalized residual is hoped to correspond to the bad data location. If measurement with largest normalized residual is indeed a bad datum, the identification indication success in this case. Otherwise, it fails. Then, next a measurement error is injected to form the next bad datum. This process is repeated until all the measurements have been identified.

For a IEEE 14 bus instrumentation configuration with an overall redundancy of K=2, the state variables is 27, therefore the total number of measurements is 54. If 45 cases out of 54 cases are identified correctly, the success rate is 83 % i.e. (45/54)=83%.

The sucess rate is an important indication which tells how the method used works. It will be used throughout this studies.

Success rate (SR) is defined as:

$$SR = NSI/TNM$$

where NSI denotes the number of cases identified successfully, TNM stands for the total number of measurements used.

The comparison is made comparing SR of an identification indication with that of either normalized residuals or other residuals. The former is used mainly in this studies. The reason will be given in the section 3.3.3.

3.3.2 Simulated Measurements Generation

The procedure is the same as that in [17] and most of this subsection is taken from [17].

In the perturbation tests, the true values of all system variables are assumed; measurements are simulated by introducing random errors to these true values.

The instrumentation configuration of the IEEE 14-bus test system is defined as follows:

- Instrumentation configuration set A:

$$P_{1-14} : 10105011000000$$

$$Q_{1-14} : 10100011006000$$

$$PL_{1-40} : 1011011711010001011011000001011011011100$$

$$QL_{1-40} : 1011011011010001011011000001011011011100$$

$$V_{1-14} : 10180011000000$$

  where P is nodal real power, Q is nodal reactive power, PL is real power flow, and QL is reactive power flow, V is nodal voltage.

Each position in the above strings corresponds to a possible measurement. If the measurement is used, then a nonzero occupies that position; if it is not used a

"0" occupies its corresponding position. For example, the "1" in the first position of P_{1-14} string indicates that the real power nodal injection at nodal 1 is measured. Later on, when it is necessary to use meters of different accuracies, the accuracy of the measurement can be indicated in a string similar to the above.

Meters are classified into four categories according to their accuracies. Therefore, if a measurement is used and it is of category 2, then a "2" would appear on its corresponding position in the above measurements' strings. When "5" appears in the above measurements' strings, it means that the corresponding measurement is simulated as bad data with the meter type of 5-4="1", i.e. meter type is 1. Similarly, when "6","7"or "8" appear in the above measurements' strings, it means that the corresponding measurements are simulated as bad data with a meter type of 2, 3 or 4 respectively.

Simulated measurements are generated according to

$$M_i = ERR_i \times T_i \pm FERR_i$$

where M and T denote simulated measurements and true values, respectively. ERR_i is assumed to be normally distributed with a maximum of 3σ ERR. FERR_i is assumed constant but its sign is chosen at random. The details about measurement simulation and meter categories can be found in [17].

3.3.3 The Normalized Residuals

The normalized residuals are defined in section 1.2. Identification methods based on the normalized residuals are widely accepted [22]. Therefore, these will be used for comparison in this chapter. Although the normalized residuals are widely accepted and used, the literature does not report on how powerful the method is. For this

reason the normalized residuals will be studied in this section in detail. Many tests have been conducted in this investigation. A few of them are presented in the following.

It is concluded in [7, 22, 24] that the normalized residuals are better indicators than the weighted residuals in most cases. Therefore, the normalized residuals rather than the weighted residuals are focused on in this thesis. The tests we conducted agree with the above conclusion, therefore, the weighted residual will not be discussed further.

The ability of an indicator to identify BD is affected by the size of the bad measurement, its location, the data base, the instrumentation configuration, the overall redundancy, the local redundancy, etc. In fact, our studies shows that the success rate (SR) can vary from lower than 50% to higher than 95% depending on these factors. The effects of these factors will be discussed individually as follows:

Effects of Local Redundancy on the Normalized Residuals

Local redundancy at a bus is defined in [7] as the number of measurements divided by the number of unknowns counting only measurements and unknowns at that bus plus at all buses up to two switch-yards away.

The first example will show how important the local redundancy is.

Instrumentation configuration 14-A1 is used with a overall redundancy 2.2, The local redundancy for measurement Q_{l2} is 1.67.

The results have shown that measurement Q_{l2} is mis-identified as bad data in 25 cases. However, after the two measurements, Q_{L1} and Q_{L3}, are added in the configuration, the local redundancy increases up to 1.9, and the mis-identified cases

decreases from 25 to 0.

The Effects of Overall Redundancy

The following table shows the variations of the success rate (SR) when using the normalized residuals with the overall redundancy:

REDUNDANCY	*SR using* R_n
1.5	40%
2.2	87%
3.0	90%
4.0	93%

Table 3.1: Effects of measurement redundancy.

The results show that the success rate of normalized residuals increases with the overall redundancy varying from 40% to 93%. The improvement is 53%. However, the improvement in SR when overall redundancy increased from 2.2 to 4.0 is not as sharp as that when the overall redundancy increased from 1.5 to 2.2. Therefore, considering economical reasons, an overall redundancy in the range of 2.2 to 3.0 can be considered reasonable. An overall redundancy of 2.2 is used throughout this chapter unless otherwise stated.

The reason why the success rate of the R_n indicator is so low when the overall redundancy is 1.5 is that Q_{L2} and P_{L10} are mis-identified as bad data 10 and 14 times respectively. The low overall redundancy results in very poor local redundancy of Q_{L2} and P_{L10} which leads to mis-identifications. It is shown in the previous section that adding two extra measurements to increase the local redundancy of Q_{L2} will reduce the number of mis-identifications greatly. It can be concluded that when

the redundancy is low, the problem of measurement topology (meter placement) become more important. When the overall redundancy is high, the problem of poor local redundancy is less likely to arise.

The Effects of Instrumentation Configuration

The next test shows the effects of the instrumentation configuration on the success rate. Table 3.2 indicates that different instrumentation configuration has different

INSCN	*SR using* R_n
14A	85%
14B	87%
14C	78%
14D	73%
14E	75%
14F	80%
14G	78%
14H	78%

Table 3.2: Effects of different instrumentation configurations on the success rate using the normalized residuals. Column 1 is the configuration code.

success rate. Some of them are better than others. This means that the success rate is affected by measurements configuration. This is another indication of the importance of the problem of meter placement. This problem is beyond the scope of this thesis.

The measurements configurations mentioned in Table 3.2 are not completely randomly configured. Visual inspection was done in order to avoid very poor local redundancy.

The reason why different configuration has different success rate is thought to

be that different locations have different responses to the same disturbance. In other words, some locations are more sensitive to the disturbance than others, this is determined by network topological and distribution of load and sources, etc.

The Effects of Bad Data Size

The next example shows the effect of the size of bad data on the identification process.

SIZE OF BD	*SR using* R_n
4σ	58%
20σ	85%
40σ	90%
60σ	92%

Table 3.3: Effects of the size of bad data. The first column shows the errors introduced to simulate bad data, the second column is the success rate using the R_n indicator.

In this study, the IEEE 14- bus system that has an overall redundancy of 2.2 is used. When the bad data size is 4 σ, the success rate of the normalized residual is less than 58%. However, when the size of the bad data increases to 60 σ, the success rate goes up to 92%. It seems that there is no significant change of the success rate when the size of bad data increases from 40 σ to 60 σ. In our studies, the bad data size is set to 20 σ. The results show that the larger the bad data size is, the easier it can be identified.

3.4 Definitions and Tests of New Indicators

In this section, four indicators related to the normalized residuals will be defined and studied. A similar four will be defined for the weighted residuals and similarly for the plain residuals (a total of 12 indicators). The testing conditions are: Overall redundancy is set to 2.2 with local redundancy kept uniform by visual inspection. The size of bad data is set to 20σ. A measurement is perturbed by introducing a random error with a maximum value of 0.5%. However, the effect of the size of the perturbation is studied later. The number of the perturbations is set to 5.

3.4.1 Results Using the Three Indicators: σ_n/R_{np}, σ_w/R_{wp} and σ_p/R_{pp}

Three indicators are defined: σ_p/R_{pp}, σ_w/R_{wp}, σ_n/R_{np}. R_{pp} (called perturbed plain residual, the first p means plain, the second p means perturbed) is an average vector, averaging the five R_ps, R_p is the plain residual, and σ_p is the standard deviation of the five $R_p s$. Similarly, R_{wp} (called perturbed weighted residuals, the first w means weighted, the second p means perturbed) is an average vector, averaging the five $R_w s$, R_w is the weighted residual, and σ_w is the standard deviation of the five $R_w s$. R_{np} (called as perturbed normalized residuals, the first n means normalized, the second p means perturbed) is an average vector, averaging the five $R_n s$, R_n is normalized residual, and σ_n is the standard deviation of the five $R_n s$. For each test, if the smallest value is of the indicator corresponding to a BD then that indicator is successful in identifying it. The tests for these three indicators are presented in table 3.4.

The above results clearly show that introducing σ, as part of the indicator,

INSCN	R_n	σ_n/R_{np}	R_w	σ_w/R_{wp}	R_p	σ_p/R_{pp}
14A	85%	13%	70%	22%	52%	33%
14B	87%	27%	77%	38%	40%	32%
14C	78%	17%	72%	25%	53%	33%
14D	73%	25%	67%	25%	45%	33%
14E	75%	33%	62%	43%	40%	32%
14F	80%	37%	75%	45%	27%	37%
14G	78%	32%	65%	40%	50%	47%
14H	78%	18%	70%	33%	47%	32%

Table 3.4: Success rates of indicators. The first column gives the code of the different instrumentation configuration.

worsen the results.

3.4.2 Results Using the Three Indicators: σ_n, σ_w and σ_p

Since σ_n, σ_w and σ_p are part of above indicators, the behavior of those σ_n, σ_w and σ_p will directly affect the results. Therefore, studying the behavior of those σ_n, σ_w and σ_p is essential.

INSCN	σ_n	σ_w	σ_p
14A	88%	68%	48%
14B	83%	78%	38%
14C	85%	72%	25%
14D	77%	67%	25%
14E	73%	62%	43%
14F	82%	75%	45%
14G	80%	65%	40%
14H	78%	70%	33%

Table 3.5: Success rates of indicators. The first column gives the code of the instrumentation configurations; column 2 ,3 and 4 are the success rates of σ_n, σ_w and σ_p respectively.

Table 3.5 clearly shows that σ_n/R_{np}, σ_w/R_{wp} and σ_p/R_{pp} fail as identification indicators. It can be concluded that the perturbed corrections are more scattered if they correspond to a bad datum than they are when they correspond to a healthy one. This is contradictory to the findings of [16]. It should be mentioned however, that differences exist between the algorithm used in this investigation and that of [16].

In any case, since σ_n, σ_w and σ_p are relatively large when they correspond to a BD, and so also the correction itself, the following indicators are suggested: $\sigma_n \times R_{np}$, $\sigma_w \times R_{wp}$ and $\sigma_p \times R_{pp}$. These indicators will be discussed later on.

The above results also suggest using σ_n, σ_w and σ_p as indicators by themselves. Results of testing these are shown in table 3.6:

INSCN	R_n	σ_n	R_w	σ_w	R_p	σ_p
14A	85%	88%	70%	68%	52%	48%
14B	87%	83%	78%	67%	40%	47%
14C	78%	85%	72%	57%	53%	53%
14D	73%	77%	67%	58%	45%	37%
14E	75%	73%	62%	63%	40%	47%
14F	80%	82%	75%	70%	27%	20%
14G	78%	80%	65%	70%	50%	48%
14H	78%	78%	70%	67%	47%	37%

Table 3.6: The comparison of success rates of indications σ_n, σ_p and σ_p with their own residuals.

The average success rate of the normalized residual is 79.25%, but that of σ_n is 80.75%. For the normalized cases, the σ indicator is 1.5% better than the normalized residuals. For the weighted case, the σ indicator is 12.38% worse than the weighted residual. For plain residuals case, the σ indicator is 3% better than the plain

residual.

3.4.3 Results Using the Three Indicators: R_{np}, R_{wp} and R_{pp}

Similar tests were done using R_{np}, R_{wp} and R_{pp} as indicators. The results are shown in Table 3.7, which essentially gives the effect of perturbation on the success rate.

INSCN	R_n	R_{np}	R_w	R_{wp}	R_p	R_{pp}
14A	85%	90%	70%	72%	52%	55%
14B	87%	92%	78%	77%	40%	53%
14C	78%	82%	72%	67%	53%	57%
14D	73%	75%	67%	68%	45%	45%
14E	75%	80%	62%	73%	40%	53%
14F	80%	85%	75%	75%	27%	33%
14G	78%	85%	65%	75%	50%	55%
14H	78%	83%	70%	70%	47%	45%

Table 3.7: Comparison of success rates of indicators R_{np}, R_{wp} and R_{pp} with their corresponding residuals R_n, R_w and R_p respectively.

The above results show that, on the average, the perturbed normalized residuals as indicators of bad data are 4% better than the normalized residuals without perturbation. The average success rates of perturbed normalized residual and normalized residual are 84% and 80% respectively. It must be pointed out that in each configuration studied, the perturbed normalized residuals as indicators of bad data are better than the normalized residuals. The improvement varies from 2% to 7%.

It is clear that the normalized residuals are more effective than the weighted or the plain residuals. Therefore, discussion will be focused on the normalized residuals performance.

The above results clearly show that the perturbed residuals as indicators of bad

data are superior to their corresponding unperturbed residuals, particularly for the normalized case.

Since the magnitudes of R_{np}, R_{wp} and R_{pp} for the bad data points are comparatively large, in fact, in most cases, they are largest, they should improve the effectiveness of the indicators σ_n/R_{np}, σ_w/R_{wp} and σ_p/R_{pp}. Those indicators were presented in section 3.4.1.and found ineffective. Therefore, the reason why the indicators σ_n/R_{np}, σ_w/R_{wp} and σ_p/R_{pp} are not effective must be due to σ_n, σ_w and σ_n. This suggests the indicators in the following sections.

3.4.4 Results Using the Three Indicators: $\sigma_p \times R_{pp}$, $\sigma_w \times R_{wp}$ and $\sigma_n \times R_{np}$

The logic behind choosing these indicators should be apparent from the preceeding sections.

INSCN	R_n	$\sigma_n \times R_{np}$	R_{wp}	$\sigma_w \times R_{wp}$	R_p	$\sigma_p \times R_{pp}$
14A	85%	90%	70%	72%	52%	55%
14B	87%	88%	78%	72%	40%	52%
14C	78%	80%	72%	63%	53%	55%
14D	73%	75%	67%	60%	45%	43%
14E	75%	80%	62%	73%	40%	52%
14F	80%	85%	75%	72%	27%	27%
14G	78%	85%	65%	72%	50%	55%
14H	78%	80%	70%	70%	47%	42%

Table 3.8: The success rates of indicators.

The results presented in table 3.8 show that, on the average, the indicator $\sigma_n \times R_{np}$ for bad data identification is 3% better than R_n. The average success

rates of perturbed normalized residual and normalized residual are 83% and 80% respectively. Also, in every configuration, the indicator $\sigma_n \times R_{np}$ for bad data identification is better than R_n, the improvement ranges from 1% to 5%.

Table 3.8 also shows that the indicator $\sigma_n \times R_{np}$ is superior to other indicators.

3.4.5 An Overall Comparison of Indicators

So far, four type of indicators have been tested. The indicators based on the normalized residuals are clearly superior to those based on the weighted and the plain residuals. In this section, the indicators based on the normalized residuals will be compared. The results of using those indicators are shown in Table 3.9.

INSCN	R_n	R_{np}	σ_n	$\sigma_n \times R_{np}$
14A	85%	90%	88%	90%
14B	87%	92%	83%	88%
14C	78%	82%	85%	80%
14D	73%	75%	77%	75%
14E	75%	80%	73%	80%
14F	80%	85%	82%	85%
14G	78%	85%	80%	85%
14H	78%	83%	78%	80%

Table 3.9: The success rates of different indicators obtained using different instrumentation configurations.

The above results are plotted in Figs. 3.1 to 3.3.

Fig. 3.1 clearly shows that R_{np} is more effective than R_n in every cases. The improvement ranges from 2% to 7%.

Fig. 3.2 clearly shows that, in general, the indicator σ_n is better than the normalized residuals, but not in every case.

Fig. 3.3 clearly shows that like indicator R_{np}, the indicator $\sigma_n \times R_{np}$ is better than the normalized residual in every case. The improvement ranges from 2% to 7%.

The following table is made by taking the average of each column of Table 3.9:

R_n	R_{np}	σ_n	$\sigma_n \times R_{np}$
80%	84%	81%	83%

Table 3.10: Comparison of the indicators on average base.

The above results show that indicator R_{np} is the best indicator for bad data identification; The indicators $\sigma_n \times R_{np}$ and σ_n are less effective than R_{np}. All these indicator are better than the widely accepted normalized residual indicator. Also, R_{np} and $\sigma_n \times R_{np}$ are better than R_n not only on an average bases, but also in every case (see table 3.8). Although indicator σ_n is superior to R_n on an average bases, it is not superior to R_n in every case.

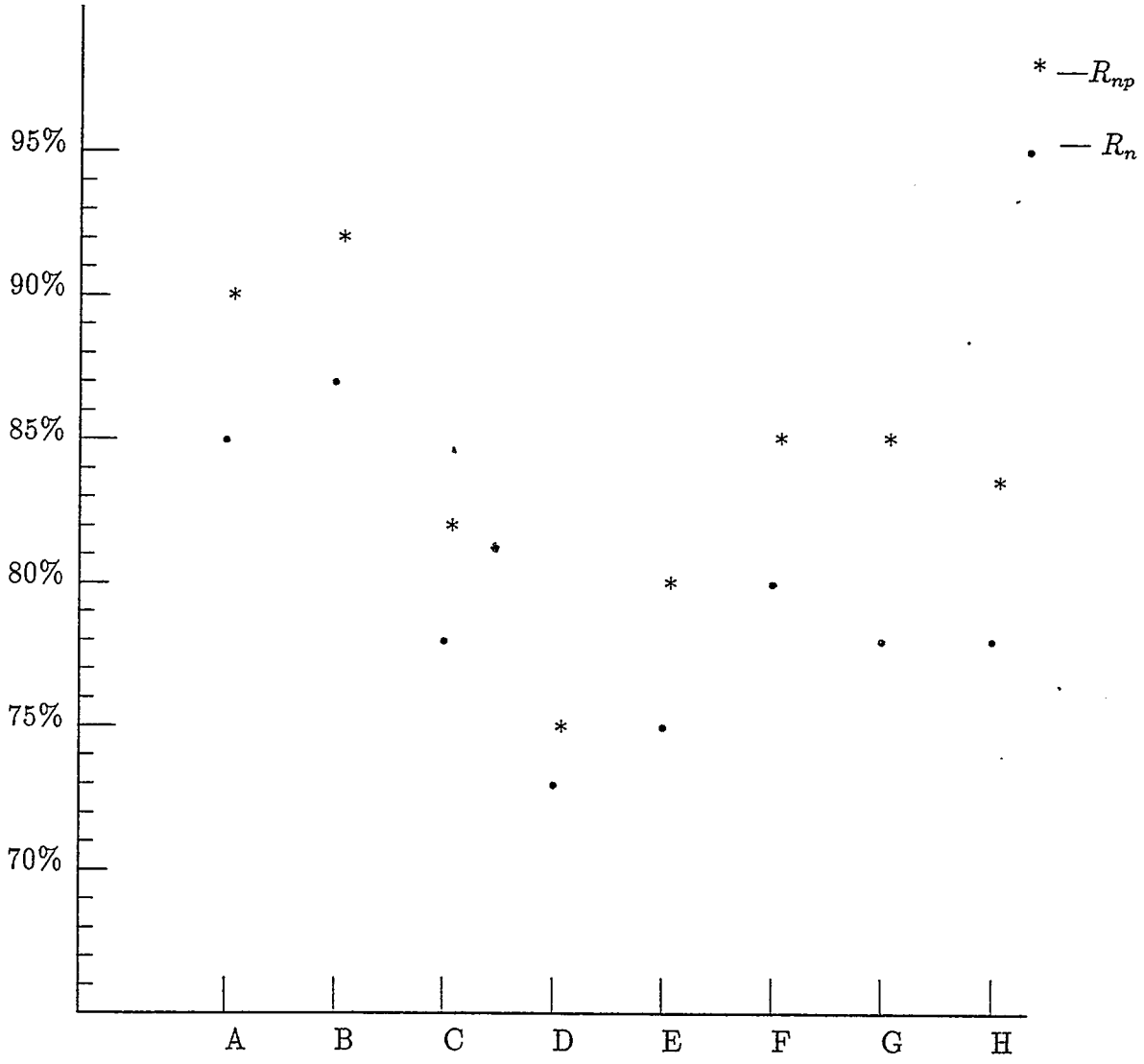

Figure 3.1: Success rates of indicators R_{np} and R_n obtained using different IEEE 14 bus test system instrumentation configurations.

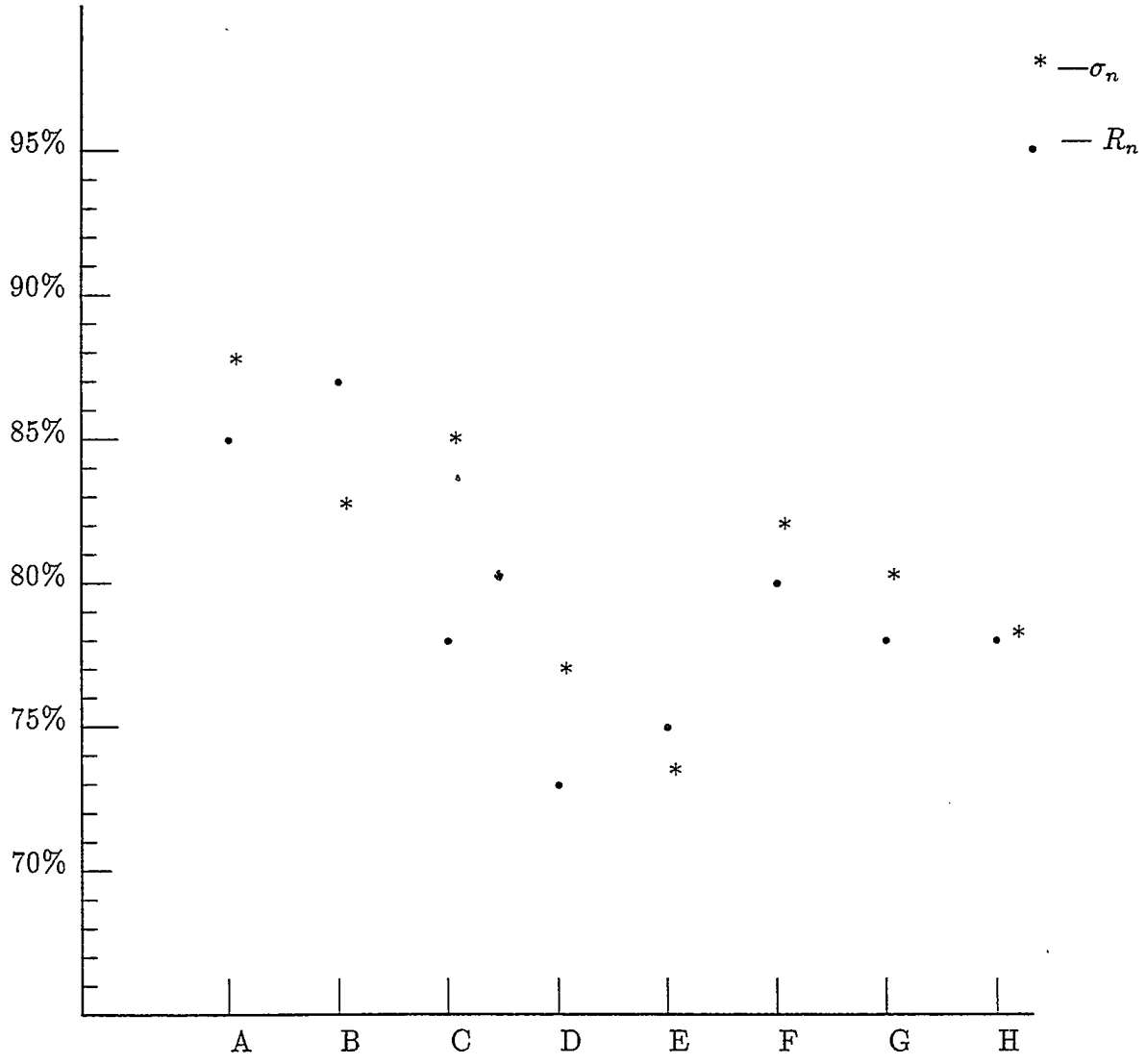

Figure 3.2: Success rates of indicators σ_n and R_n obtained using different IEEE 14 bus test system instrumentation configurations.

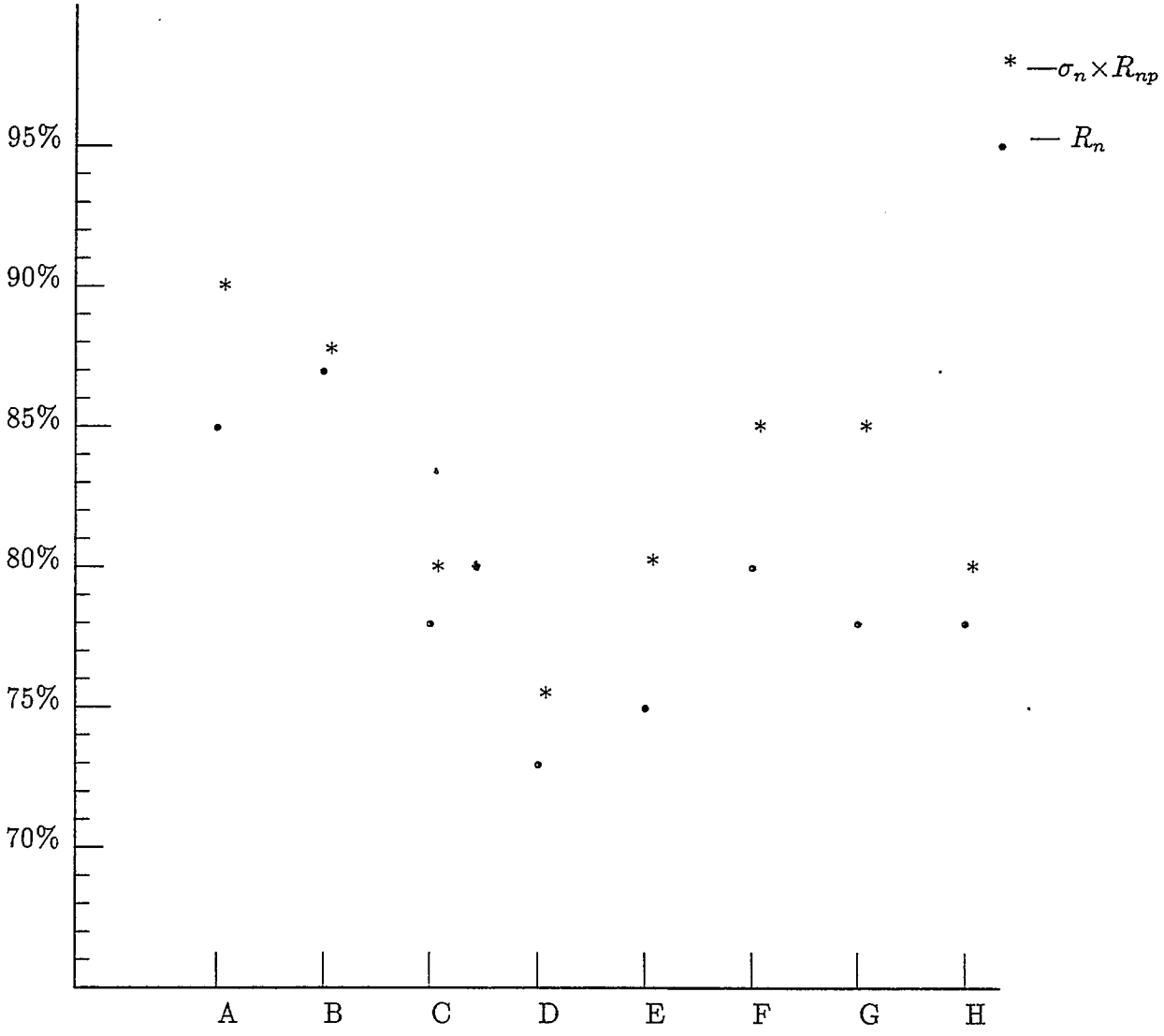

Figure 3.3: Success rates of indicators $\sigma_n \times R_{np}$ and R_n obtained using different IEEE 14 bus test system instrumentation configurations.

3.4.6 Studies on the IEEE 30-Bus System

The IEEE 30 bus system is used to conduct more tests. The instrumentation configuration used can be found in the program listing in Appendix A. The system data can be found in [62]. The test conditions are the same as those used in the previous tests conducted on the IEEE 14 bus system.

R_{np}-R_n	σ_n-R_n	$\sigma_n \times R_{np}$-R_n
7.5%	1.0%	4.6%

Table 3.11: The effectiveness of various indicators relative to R_n.

The results show that R_{np} as indicator for bad data identification is the best.

The results obtained using the IEEE 30 bus system agree with those obtained using the IEEE-14 bus system.

3.5 Effects of Various Parameters on the Results

All the tests conducted are based on the test condition discussed in that section. As mentioned in section 3.3.3, the overall and local redundancy of measurements and bad data size, etc. will affect the effectiveness of the normalized residual. Do they alter the above conclusions? Are indicators R_{np}, σ_n, $\sigma_{np} \times R_{np}$ remain more effective than the normalized residuals when these factors vary? In addition to these factors, do the number of perturbations and the size of the perturbation affect the obtained conclusions?

The following studies are designed to answer these questions. In the following studies, indicators R_n, R_{np}, σ_n, $\sigma_{np} \times R_{np}$ are selected to be tested since they are obviously better than the others.

IEEE-14 bus system instrumentation configuration 14-A is used to conduct the following tests unless otherwise stated.

3.5.1 Effect of Redundancy of Measurement

The following values of the overall redundancy are used: 1.5, 2.0, 3.0 and 4.0.

$REDCY$	R_n	R_{np}	σ_n	$\sigma_n \times R_{np}$
1.5	40%	46%	40%	41%
2.2	85%	90%	88%	90%
3.0	90%	96%	95%	96%
4.0	93%	97%	94%	97%

Table 3.12: Success rates of the indicators obtained using different overall redundancies.

The results show:

(1): The success rates of all indicators will increase with the increase of the overall redundancy. The rate of improvement, however, is more pronounced when the redundancy is low.

(2): The average success rate of indicator R_{np} is 5.3% higher than that of the normalized residual. The improvement ranges from 4% to 6%.

(3): The average success rate of indicator σ_n is 4% higher than that of the normalized residual. The improvement ranges from 0% to 5%.

(4): The average success rate of indicator $\sigma_n \times R_{np}$ is 2.3% higher than that of the normalized residual. The improvement ranges from 1% to 6%.

Fig 3.4 clearly shows how the overall redundancy affects the effectiveness of indicators R_{np} and R_n.

It is clear that no matter how the overall redundancy changes, R_{np} is superior to R_n.

The reason why the success rate are very low when the overall redundancy is reduced to 1.5 is that this configuration suffers from the poor local redundancy so that Q_{L2} and P_{L10} are mis-identified as bad data 10 and 14 times respectively.

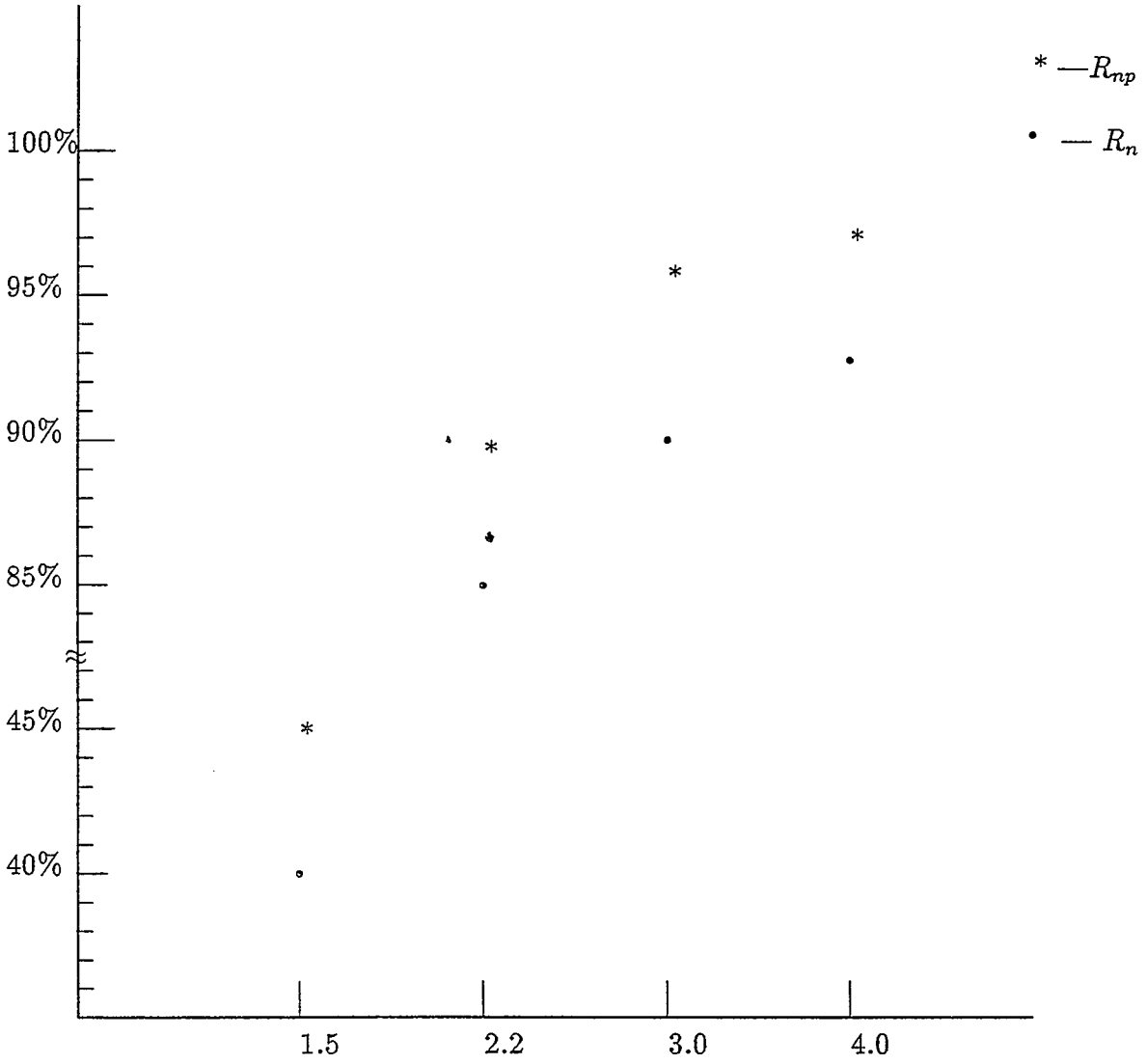

Figure 3.4: Success rates of indicators R_{np} and R_n obtained using different overall redundancies.

3.5.2 Effects of Bad Data Size

The size of bad data is varied from 4 σ to 60 σ.

SBD	R_n	R_{np}	σ_n	$\sigma_n \times R_{np}$
4σ	58%	68%	60%	65%
20σ	85%	90%	88%	90%
40σ	90%	93%	92%	92%
60σ	92%	95%	95%	95%

Table 3.13: Effects of size of the bad data.

The results show:

(1): The success rates of all indicators increase with the increase of bad data size.

(2): The success rates are much more sensitive to changes in bad data size when the BD size is in the lower range (4σ to 20σ).

(3): The average success rate of R_{np} is 5.3% higher than that of R_n. The improvement ranges from 3% to 10%. The 10% improvement is obtained when the bad data size is 4σ. This means that the more difficult the case, the more powerful the R_{np} indicator is comparison with the R_n indicator.

(4): The average success rate of indicator σ_n is 2.5% higher than that of the normalized residuals. The improvement ranges from 2% to 3%.

(5): The average success rate of indicator $\sigma_n \times R_{np}$ is 4.3% higher than that of the normalized residual. The improvement ranges from 2% to 7%.

Fig 3.5 clearly shows how the size of bad data affects the performance of indicators R_{np} and R_n.

It is clear that the R_n is least effective no matter how bad data size changes.

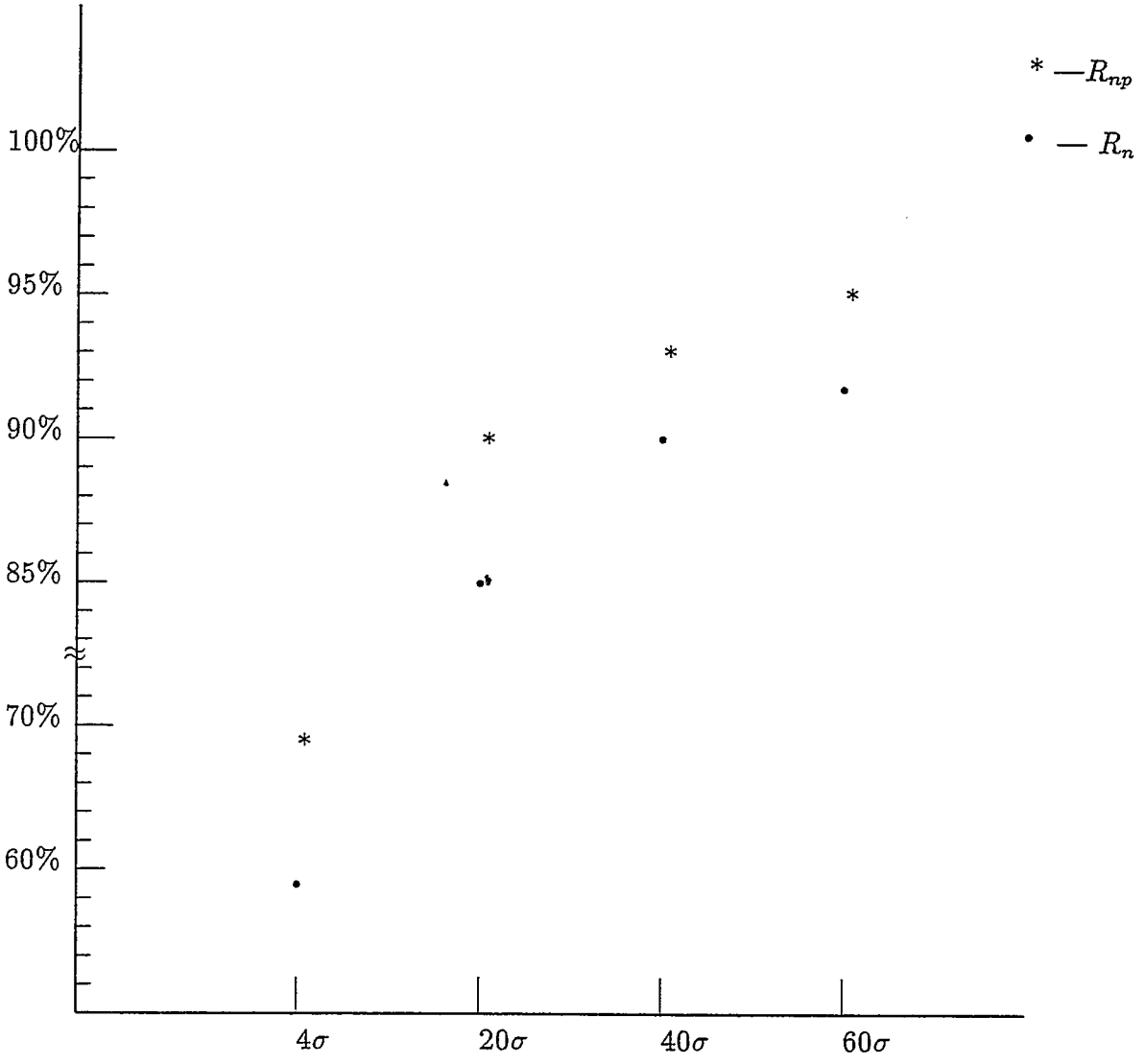

Figure 3.5: Success rates of indicators R_{np} and R_n obtained using different size of bad data.

3.5.3 Effects of Measurement Accuracy

Following tests are to study the effects of metering accuracy. The tests are conducted by changing the type of the meters, i.e. changing the sizes of the measurements errors. In order to compare the results easily, the type of all the measurement's meters used in a configuration will be the same each time and will vary from 1 to 4. A type 1 meter is a better quality meter and therefore more expensive.

R_n	R_{np}	σ_n	$\sigma_n \times R_n$	$\sigma_n \times R_{np}$
1	88%	95%	90%	95%
2	82%	90%	83%	90%
3	73%	85%	80%	83%
4	70%	82%	72%	82%

Table 3.14: Effects of measurements' accuracies (meter's quality).

Table 3.15 gives the averages of the corresponding columns of Table 3.14.

R_n	R_{np}	σ_n	$\sigma_n \times R_{np}$
78%	88%	81%	88%

Table 3.15: Success rates of indicators R_{np} and R_n obtained using different types of meters.

The results show that indicator R_{np} is 10% superior to the normalized residual on the average. The improvement ranges from 8% to 12%. It should be mentioned that when the instrumentation are getting worse (all the meters are type 4, or 3), the improvement reaches 12%. This demonstrates that the more difficult the case is, the more powerful indicator R_{np} is.

When all the meters are type 1, the success rate of indicator R_{np} is 95%.

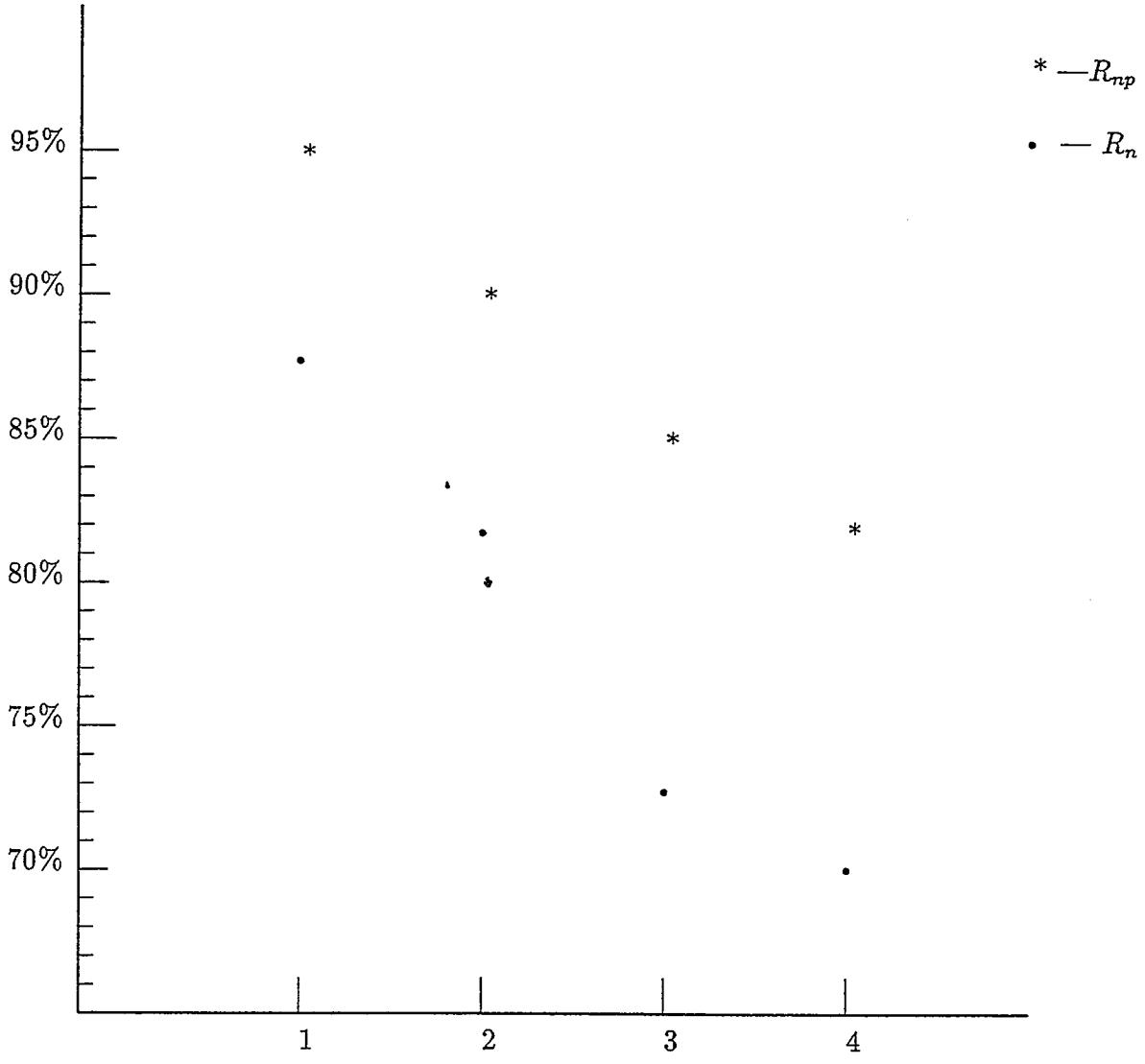

Figure 3.6: Success rates of indicators R_{np} and R_n obtained using different types of meter.

3.5.4 Effects of the Perturbation Size

Tests are done to study the effects of perturbation size. To conduct this study, the size of the perturbation is selected to lie at random with the following absolute maximum values: 0.1%, 0.5%, 1% and 5% of the measured values.

PS	R_n	R_{np}	$\sigma_n * R_{np}$	σ_n
0.1%	85%	90%	88%	90%
0.5%	85%	90%	88%	90%
1.0%	85%	90%	88%	90%
5.0%	85%	90%	83%	90%

Table 3.16: Effects of the perturbation size (PS) on the success rate of different indicators.

The results show that the success rate of the indicators are not affected by the perturbation size, except in one case where the success rate of $\sigma_n \times R_{np}$ is decreased to 83%.

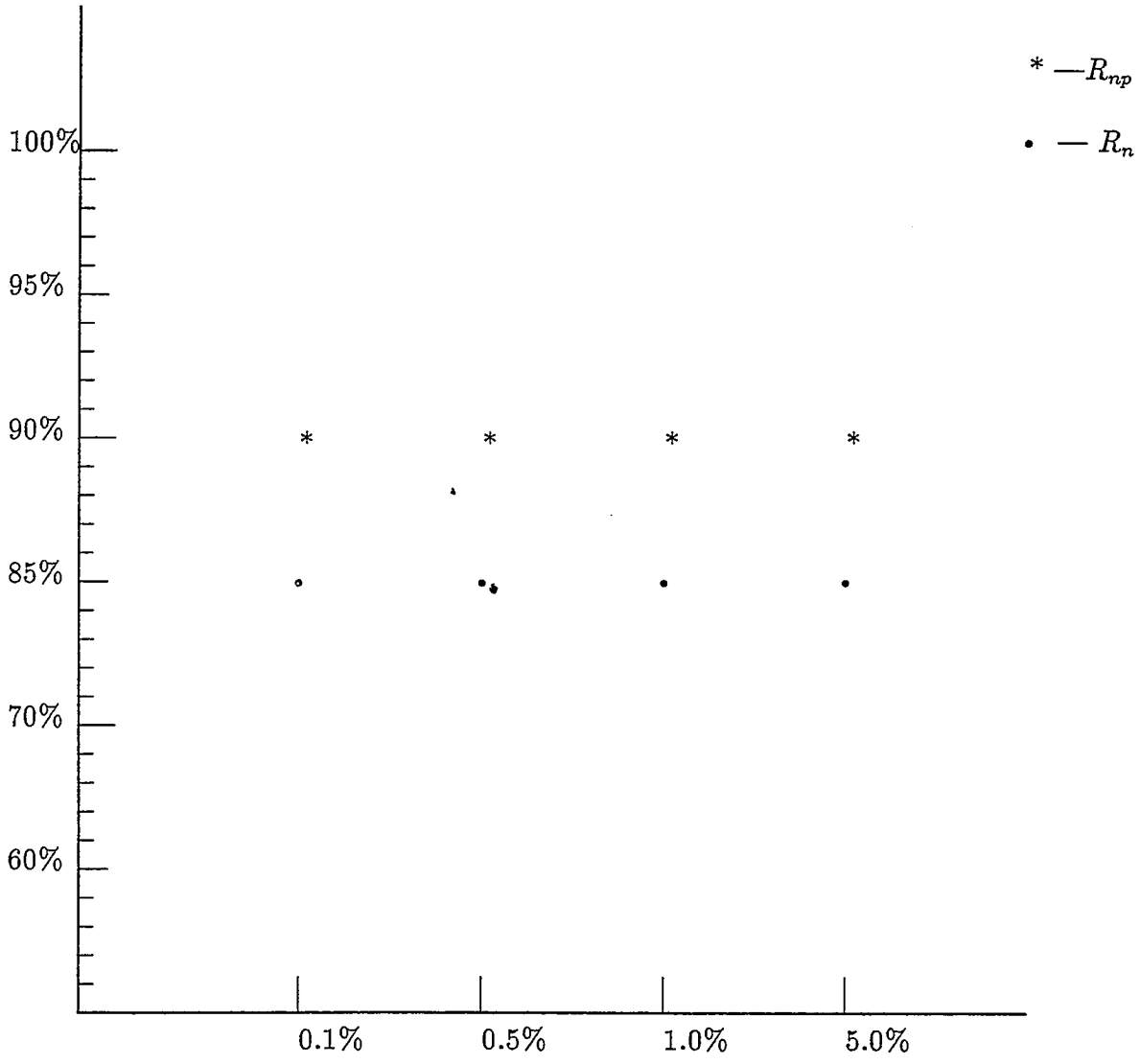

Figure 3.7: Success rates of indicators R_{np} and R_n obtained using different sizes of perturbation.

3.5.5 Effects of Number of Perturbations

DP	R_n	R_{np}	σ_n	$\sigma_n \times R_{np}$
2	85%	82%	82%	82 %
3	85%	92%	88%	90 %
5	85%	90%	88%	90 %
10	85%	92%	82%	92 %

Table 3.17: Effects of the number of perturbations on the success rate. Column 1 shows the number of perturbations (NP) used.

Studies are conducted using different number of perturbations. The results are summarized in Table 3.17. Taking the number of perturbations greater than 3 does not appear to offer any advantages.

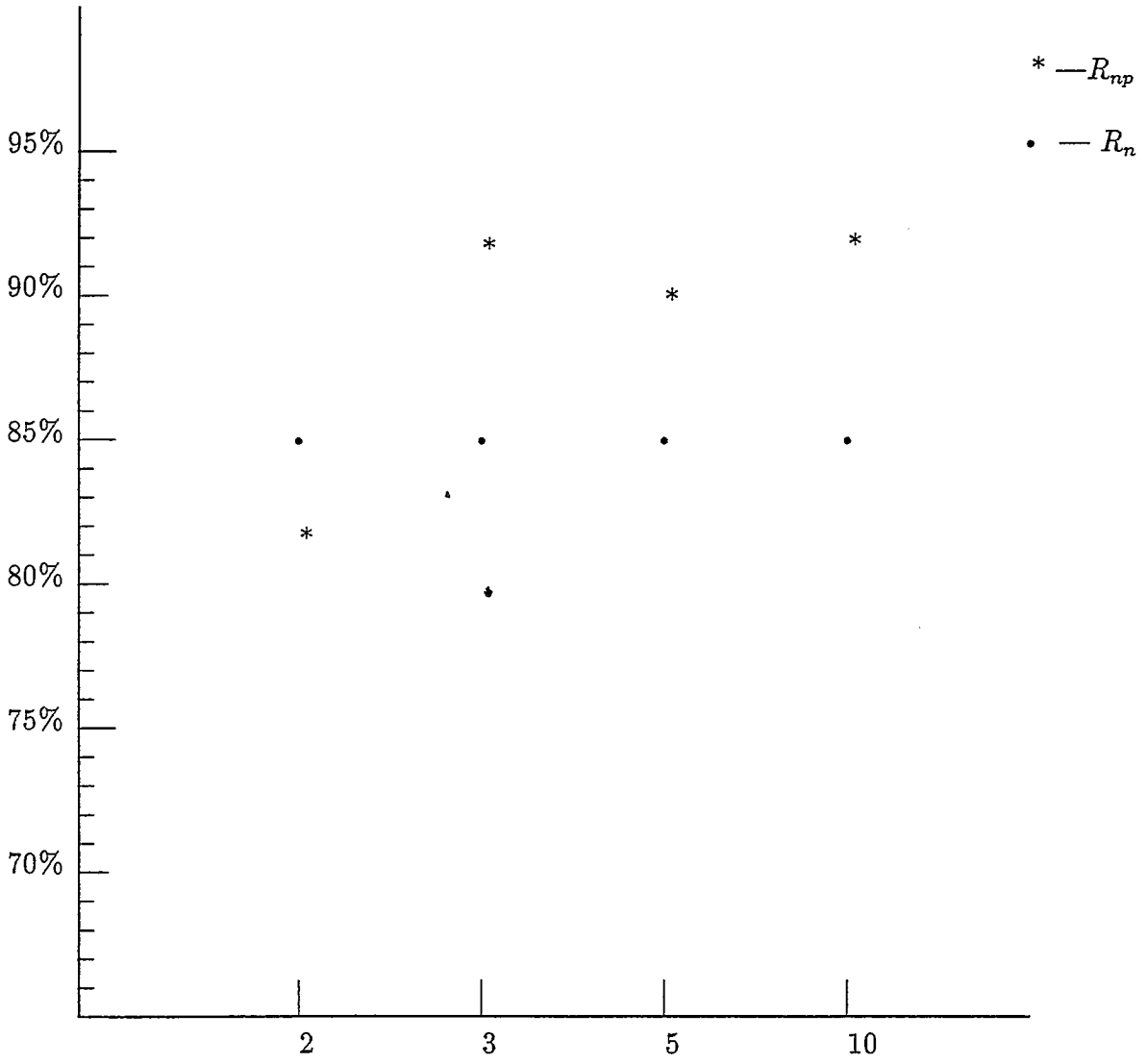

Figure 3.8: Success rates of indicators R_{np} and R_n obtained using different numbers of perturbations.

3.6 Conclusion

In this chapter, the perturbation method for bad data identification in PSEE is studied. Four classes of possible indicators (total number of indicators is 12) are defined and many cases are studied.

Results show that the perturbed normalized residuals R_{np} is a most effective indicator. It is superior to the widely accepted normalized residuals. The improvement in the success rate can be as high as 12%. The indicator R_{np} is better than the normalized residual indicator not only on an average bases, but also in every case. Therefore, it is more reliable than other indicators.

The computational requirements for this method is little higher than those of the normalized residual method.

The indicator $\sigma_n \times R_{np}$ is second best. Indicator σ_n follows. Although these two indicators are better than the normalized residual on the average, they are affected by the changes in the test conditions.

The performance of indicator σ_n / R_{np} inferior to that of R_{np}.

Appendix A

The Main Program And Some Subroutins For Perturbation Method

```
C     MAIN PROGRAM

C     THIS PROGRAM IS USED TO TEST THE PERTURBATION METHOD OF BAD DATA
C     IDENTIFICATION. THE PROGRAM MAINLY CONSISTS OF THREE PARTS: PART
C     ONE IS TO SIMULATE MEASUREMENTS, PART TWO EXECUTES PSSE PROGRAM
C     AND PART THREE PERFORM THE BAD DATA IDENTIFICATION.
      INTEGER FJM,WMN,IR,DFM,BUS

C     THIS STATEMENT IS FOR THE 14-BUS SYSTEM
      PARAMETER(NB=14,NLINES=20,NT=40,NNBB=28,NM=60,NMM=122,
     1NNBB1=27,NB1=13,NL=9,K1=10,K2=11,DFM=22,BUS=14,
     1ERR1=0.002,ERR2=0.02,FERR1=0.003,FERR2=0.0035,N=5,nbb=20)



c     THIS STATEMENT IS FOR THE 30-BUS SYSTEM
c      PARAMETER(NB=30,NLINES=41,NT=82,NNBB=60,NM=132,NMM=254,
c     1NNBB1=59,NB1=29,NL=24,K1=25,K2=26,DFM=53,BUS=30,
c     1ERR1=0.002,ERR2=0.02,FERR1=0.003,FERR2=0.0035,N=5,nbb=20)

      PARAMETER(CRIT=0.005,KMAX=100,NUDR=100,
     1NCASE=10)

c     NUDR is the number of uniformely distributed random numbers
c     used to generate one normally distributed number.

      COMPLEX YC(NLINES),ZLINE(NLINES),ST(NT),SP,E(NB)
      DIMENSION MTOBO(NT,4),FM(DFM,DFM),VSP(NB),CONN(NNBB)
      REAL PSP(NB),QSP(NB),V(NB),D(NB),P(NB),Q(NB),
     1PNM(NB),QNM(NB),VT(NB),DT(NB)
      REAL PLM(NT),QLM(NT),PLT(NT),QLT(NT)
      REAL JK11(NB,NB),JK12(NB,NB),JK21(NB,NB),
     1JK22(NB,NB),G(NB,NB),B(NB,NB)
      REAL JK31(NT,NB),JK32(NT,NB)
      REAL JK41(NT,NB),JK42(NT,NB)
      REAL JK51(NB,NB),JK52(NB,NB)
      REAL JKEM(NMM,NNBB),IM(NNBB1,NNBB1),NY(NLINES)

      REAL CONM(NMM),CONMM(NMM),CON1(NNBB1),CON2(NNBB1)
      REAL ESTIM(NNBB1,NNBB1)
      REAL PLC(NT),QLC(NT),PNC(NB),QNC(NB)
      REAL VM(NB),ZM(NMM),WMM(NMM,NMM),CONMMM(NMM)
      REAL NDR,bm
      REAL A(N),C(N),AVE(NM), SIG(NM), SIGOAVE(NM), CONNN1(NM,N)
      REAL CONNN2(NM,N),CONNN3(NM,N),WGH(NM),AVEO(NM,2),AINP(NM)
      REAL SMALL
```

```
      REAL CONNN10(NM,2),CONNN20(NM,2),CONNN30(NM,2),CONNNI(NM)
      REAL SIGO(NM,2),SIGOAVEO(NM,2),SIGTAVEO(NM,2)
      REAL SIGtAVE(NM), sigsave(nm)
      REAL JCT(NCASE),JMT(NCASE),JCM(NCASE)
      REAL AJCT(NCASE),AJMT(NCASE),AJCM(NCASE)
      REAL SUT(NCASE),SUM(NCASE),SUC(NCASE)


      INTEGER AM(NMM),CTI(NCASE),MTI(NCASE),CMI(NCASE)

C     DFM: DIMENSION OF FM MATRIX, EQUAL TO THE NUMBER OF UNKNOWNS IN
C          THE LOAD FLOW PROGRAM, DFM = NB + NL -1

      REAL CON(NM),JKE(NM,NNBB1),JKET(NNBB1,NM)
      REAL AESTIM(NNBB1,NM),WM(NM,NM)
      REAL JKED(NNBB,NNBB),COND(NNBB),JKEI(5,NNBB),CONI(5)
      REAL SEN(5,NNBB),CONIP(5),JKEDI(NNBB,NNBB)
      REAL JI(NM,NNBB1),JIJ(NM,NM),SB(NM),NOMLC(NM)
      DATA WMN/3/
      DATA VT(1),DT(1)/1.06,0.0/



C********************** 14-A    **********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 14-A FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
C*****************************************************************

      DATA (AM(I),I=1,14)/0,1,0,0,0,2,0,0,0,0,1,0,0,1/
      DATA (AM(I),I=15,28)/0,1,0,0,0,2,0,0,0,0,1,0,0,1/
      DATA (AM(I),I=29,68)/1,1,0,1,2,1,1,0,3,0,1,0,2,1,0,0,1,2,0,1,
     10,1,0,0,2,1,1,0,2,0,1,2,0,1,0,1,0,1,2,2/
      DATA (AM(I),I=69,108)/1,1,0,1,2,1,1,0,3,0,1,0,2,1,0,0,1,2,0,1,
     10,1,0,0,2,1,1,0,2,0,1,2,0,1,0,1,0,1,2,2/
      DATA (AM(I),I=109,122)/0,1,0,0,0,2,0,0,0,0,0,0,0,0/
C*****************************************************************

C********************** 14-B    **********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 14-B FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
C*****************************************************************
C
```

```
C     THESE DATA STATEMENTS ARE FOR 14-BUS SYSTEM
c      DATA (AM(I),I=1,14)/0,0,1,0,0,4,0,0,1,2,0,0,0,0/
c      DATA (AM(I),I=15,28)/0,0,1,0,0,4,0,0,1,2,0,0,0,0/
c      DATA (AM(I),I=29,68)/1,1,3,1,1,2,1,0,1,0,4,0,1,0,0,3,3,0,0,1,
c     10,1,0,2,0,1,1,0,1,2,1,0,0,0,2,1,0,0,1,4/
c      DATA (AM(I),I=69,108)/1,1,3,1,1,2,1,0,0,0,1,0,1,1,0,3,3,0,0,1,
c     10,1,0,2,0,1,1,0,1,2,1,0,0,0,2,1,0,0,1,4/
c      DATA (AM(I),I=109,122)/0,0,1,0,0,4,0,0,1,1,0,0,0,0/

C*********************** 14-C    ***********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 14-C FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
C*********************************************************************
C
C     THESE DATA STATEMENTS ARE FOR 14-BUS SYSTEM
c      DATA (AM(I),I=1,14)/0,1,0,0,4,1,0,1,2,0,0,0,1,0/
c      DATA (AM(I),I=15,28)/0,1,0,0,4,1,0,1,2,0,0,0,1,0/
c      DATA (AM(I),I=29,68)/1,1,1,1,1,0,1,1,0,0,0,4,0,1,0,1,0,0,0,3,
c     11,0,0,0,1,3,0,1,0,1,0,1,0,0,4,1,0,2,0,1/
c      DATA (AM(I),I=69,108)/0,3,1,1,1,1,2,1,0,0,0,4,0,1,0,1,0,0,0,3,
c     11,0,0,0,1,3,0,1,0,1,0,1,0,0,4,1,0,2,0,1/
c      DATA (AM(I),I=109,122)/1,0,0,0,4,1,0,1,2,0,0,0,1,0/
C*********************************************************************

C*********************** 14-D    ***********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 14-D FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
C*********************************************************************
C
C     THESE DATA STATEMENTS ARE FOR 14-BUS SYSTEM
c      DATA (AM(I),I=1,14)/0,0,1,0,0,1,0,0,0,1,0,0,0,1/
c      DATA (AM(I),I=15,28)/0,0,1,0,0,1,0,0,0,2,0,0,0,1/
c      DATA (AM(I),I=29,68)/4,1,2,1,1,2,0,0,1,1,0,3,0,0,2,4,1,0,0,0,
c     12,0,1,0,2,1,3,0,0,1,1,0,1,0,3,0,2,0,4,1/
c      DATA (AM(I),I=69,108)/4,0,2,1,1,2,0,1,1,1,0,3,0,0,2,4,1,0,0,0,
c     12,0,1,0,2,0,3,1,0,1,1,0,1,0,3,0,2,0,4,1/
c      DATA (AM(I),I=109,122)/0,0,0,1,0,0,3,0,0,0,2,0,0,1/
C*********************************************************************

C*********************** 14-E    ***********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 14-E FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
```

```
C*************************************************************************
C
C     THESE DATA STATEMENTS ARE FOR 14-BUS SYSTEM
c      DATA (AM(I),I=1,14)/0,1,2,0,3,0,0,4,0,0,0,2,0,1/
c      DATA (AM(I),I=15,28)/0,1,2,0,3,0,0,4,0,0,0,2,0,1/
c      DATA (AM(I),I=29,68)/0,2,1,1,4,1,0,2,0,0,1,3,0,1,2,0,4,0,3,0,
c     11,1,0,0,2,2,3,0,0,4,0,0,0,0,3,0,0,2,1,0/
c      DATA (AM(I),I=69,108)/0,2,1,1,4,1,0,2,0,0,1,3,0,1,2,0,4,0,3,0,
c     11,1,0,0,2,2,3,0,0,4,0,0,0,0,3,0,0,2,1,0/
c      DATA (AM(I),I=109,122)/1,0,2,0,3,0,0,4,0,0,0,2,0,1/
C*************************************************************************

C********************** 14-F    **********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 14-A FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
C*************************************************************************
C
c      DATA (AM(I),I=1,14)/3,0,0,2,3,1,0,1,0,4,0,0,0,1/
c      DATA (AM(I),I=15,28)/3,0,0,2,3,1,0,1,0,4,0,0,0,1/
c      DATA (AM(I),I=29,68)/0,0,2,1,0,1,3,0,1,1,0,1,0,4,0,1,1,0,0,0,
c     10,3,0,2,2,0,0,1,0,4,0,3,1,0,0,2,1,0,1,0/
c      DATA (AM(I),I=69,108)/0,0,2,0,0,1,3,0,1,1,0,1,0,4,0,0,1,0,0,0,
c     10,3,0,2,2,0,0,1,0,4,0,3,1,0,0,2,0,0,1,0/
c      DATA (AM(I),I=109,122)/3,0,0,2,3,1,1,1,0,4,0,2,0,1/
C*************************************************************************

C********************** 14-G    **********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 14-G FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
C*************************************************************************
C
C     THESE DATA STATEMENTS ARE FOR 14-BUS SYSTEM
c      DATA (AM(I),I=1,14)/0,0,2,0,3,0,0,4,0,0,1,1,0,2/
c      DATA (AM(I),I=15,28)/0,0,2,0,3,0,0,4,0,1,0,1,0,2/
c      DATA (AM(I),I=29,68)/0,1,1,0,3,2,1,1,0,4,1,1,0,2,1,0,3,2,0,0,
c     12,0,3,0,0,4,0,0,2,1,0,1,1,1,0,4,0,1,1,0/
c      DATA (AM(I),I=69,108)/0,1,1,0,3,2,0,0,0,4,1,0,0,2,1,0,3,2,0,0,
c     12,0,3,0,0,4,0,0,2,0,0,1,0,1,0,4,0,0,1,0/
c      DATA (AM(I),I=109,122)/0,0,2,0,3,0,0,4,0,1,1,0,0,2/
C*************************************************************************

C********************** 14-G    **********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 14-G FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
```

```
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
C**********************************************************************
c
c      DATA (AM(I),I=1,14)/0,1,0,0,0,1,0,0,0,0,0,0,0,0/
c      DATA (AM(I),I=15,28)/0,1,0,0,0,1,0,0,0,0,0,0,0,0/
c      DATA (AM(I),I=29,68)/1,2,0,1,2,1,1,0,2,0,1,2,2,1,0,0,1,2,1,1,
c     10,1,0,0,2,1,1,0,2,0,1,2,0,1,0,1,0,1,2,2/
c      DATA (AM(I),I=69,108)/1,2,0,1,2,1,1,0,2,0,1,2,2,1,0,0,1,2,1,1,
c     10,1,0,0,2,1,1,0,2,0,1,2,0,1,0,1,0,1,2,2/
c      DATA (AM(I),I=109,122)/0,1,0,0,0,1,0,0,0,0,0,0,0,0/

C************************ 30-A     ***********************************
C     THIS IS INSTRUMENTATION CONFIGURATION 30-A FOR 14-BUS SYSTEM.
C     FOLLOWING DATA STATEMENTS SPECIFY WHICH MEASUREMENTS ARE
C     AVALIABLE AND SPECIFY THE ACCURACY CATEGORY OF EACH AVALIABLE
C     MEASUREMENTS.
C**********************************************************************
C
c      DATA (AM(I),I=1,30)/0,2,0,0,3,0,0,0,1,2,0,4,1,2,0,0,2,1,0,0,
c     10,4,1,0,0,0,1,0,3,1/
c      DATA (AM(I),I=31,60)/0,2,0,0,3,0,0,0,1,2,0,4,1,2,0,0,2,1,0,0,
c     10,4,1,0,0,0,1,0,3,1/
c      DATA (AM(I),I=61,101)/0,0,1,1,2,1,0,0,1,2,0,1,0,0,2,1,4,0,3,0,
c     10,2,0,3,1,1,1,0,3,1,2,1,0,4,0,1,1,1,3,1,0/
c      DATA (AM(I),I=102,142)/0,0,0,1,1,1,1,1,2,0,1,0,2,0,4,0,0,2,0,1,
c     10,0,0,3,2,1,0,1,1,0,3,1,2,0,1,1,1,1,0,0,0/
c      DATA (AM(I),I=143,183)/1,0,1,1,2,0,0,0,1,2,0,0,0,0,0,1,4,0,3,0,
c     10,2,0,3,1,1,1,0,3,0,2,1,0,4,0,0,1,0,3,1,0/
c      DATA (AM(I),I=184,224)/1,4,0,0,1,1,0,0,2,0,1,0,2,0,4,0,1,2,0,1,
c     10,0,0,3,2,1,0,1,0,0,3,0,2,0,0,1,0,1,0,0,0/
c      DATA (AM(I),I=225,254)/1,2,0,0,3,1,0,0,1,2,0,4,1,2,1,0,2,0,1,0,
c     10,4,1,0,0,2,0,0,3,1/
C**********************************************************************
      DATA NR,FJM,MFJM/1,0,0/

      DOUBLE PRECISION RAN

      OPEN(15,FILE='W14B.DAT',STATUS='OLD')
c      OPEN(15,FILE='W30c.DAT',STATUS='OLD')
        OPEN(50,FILE='5.DAT',STATUS='NEW')




C     THIS READS IN THE SYSTEM DATA FROM DATA FILE
```

```
      CALL SYSDATA(NB,NLINES,G,B,NT,K1,MTOBO,PSP,QSP,VSP,SP,
     1YC,ZLINE,K2)

   10 CONTINUE

      DO 20 I = 1,NB
      VT(I) = VT(1)
      DT(I) = DT(1)
   20 CONTINUE

      IF(K1.EQ.NB)GO TO 40
      DO 30 I=K2,NB
      VT(I) = VSP(I)
   30 CONTINUE
   40 CONTINUE


C     LOAD FLOW CALCULATION TO ESTABLISH THE TRUE VALUES OF STATE
C     VARIABLES.
      CALL FRANK(NB,NL,NNBB,K1,KMAX,CRIT,P,PSP,Q,QSP,VT,
     1DT,CONN,G,B,JK11,JK12,JK21,JK22,FM,DFM,FJM,MFJM,NR,1)

C     IR IS A SEED TO GENERATE RANDOM NUMBERS.
      IR=2043


c     BAD DATA POINTS SELECTION
      INN=0
      IN=0
      IINN=0
444   IINN=IINN+1
      IF(AM(IINN).EQ.0)GO TO 444
         IIN=AM(IINN)
         AM(IINN)=4+AM(IINN)
         IN=IN+1
c     PRINT OUT THE INFORMATION OF WHICH MEASUREMENTS ARE
C     AVALIABLE AND WHAT ARE THE ACCURACY CATEGORY OF
C     THE AVAILABLE MEASUREMENTS.
C     PRINT OUT THE INFORMATION OF WHICH MEASUREMENTS ARE
C     AVALIABLE AND WHAT ARE THE ACCURACY CATEGORY OF
C     THE AVAILABLE MEASUREMENTS.

c      WRITE(50,960)(AM(I),I=1,NB)
c  960 FORMAT('     P(I):',14I1)
c      WRITE(50,961)(AM(I+NB),I=1,NB)
c  961 FORMAT('     Q(I):',14I1)

c      WRITE(50,965)(AM(I+NNBB),I=1,NT)
```

```
c  965 FORMAT('     PL(I):',40I1)
c      WRITE(50,966)(AM(I+NNBB+NT),I=1,NT)
c  966 FORMAT('     QL(I):',40I1)
c      WRITE(50,967)(AM(I+NNBB+NT+NT),I=1,NB)
c  967 FORMAT('     V(I):',14I1)
c  998 CONTINUE




C     THIS GENERATES SIMULATED MEASUREMENTS OF A SYSTEM
      CALL MGB(AM,NUDR,VM,VT,DT,E,MTOBO,NT,NB,ST,
     1ZLINE,YC,NLINES,PLM,QLM,PLT,QLT,PNM,QNM,P,Q,NDR,NMM,NNBB,
     1ERR1,ERR2,FERR1,FERR2,IR,nbb)

C     THIS CALCULATE THE WEIGHTING MATRIX WMM
c      CALL WMGP(AM,WMM,NMM,PNM,QNM,PLM,QLM,VM,NB,NNBB,NT,
c     1ERR1,ERR2,FERR1,FERR2,WMN)

        KK=0.0
      IF(KK.EQ.0.0) GO TO 555

C     THIS GENERATES PERTURBED MEASUREMENTS OF A SYSTEM
  666 CALL MGBP(AM,NUDR,VM,VT,DT,E,MTOBO,NT,NB,ST,
     1ZLINE,YC,NLINES,PLM,QLM,PLT,QLT,PNM,QNM,P,Q,NDR,NMM,NNBB,
     1ERR1,ERR2,FERR1,FERR2,IR,nbb)


C     THIS IS CALCULATION OF PSSEP
      CALL PSSEP(V,D,NLINES,NY,YC,AM,WMM,NMM,PNM,QNM,PLM,QLM,
     1VM,NB,NNBB,NT,ERR1,ERR2,FERR1,FERR2,WMN,PNC,QNC,PLC,QLC,B,G,
     1MTOBO,E,ST,ZLINE,JK11,JK12,JK21,JK22,JK31,JK32,JK41,
     1JK42,JK51,JK52,JKEM,JKE,CON,CONM,WM,JKET,NNBB1,
     1AESTIM,ESTIM,CON1,IM,CON2,ITEST,CRIT,NM,KMAX,NB1)
     THIS IS THE END OF THE PSEEP

  555 CONTINUE
      IF(KK.GT.0.0) GO TO 556

C     THIS IS CALCULATION OF PSSE
      CALL PSSE(V,D,NLINES,NY,YC,AM,WMM,NMM,PNM,QNM,PLM,QLM,
     1VM,NB,NNBB,NT,ERR1,ERR2,FERR1,FERR2,WMN,PNC,QNC,PLC,QLC,B,G,
     1MTOBO,E,ST,ZLINE,JK11,JK12,JK21,JK22,JK31,JK32,JK41,
     1JK42,JK51,JK52,JKEM,JKE,CON,CONM,WM,JKET,NNBB1,
     1AESTIM,ESTIM,CON1,IM,CON2,ITEST,CRIT,NM,KMAX,NB1)
c     THIS IS THE END OF THE PSEE
```

```
  556 CONTINUE

C     CALCULATING THE FACTORS OF NORMALIZED AND WEIGHTED RESIDUALS
      CALL RESDFCT(JKE,NM,IM,NNBB1,JI,JKET,JIJ,WM,SB,NOMLC,WGH)



C     CALCULATING THE PLAIN, WEIGHTED AND NORMLIZED RESIDUARLS AND
C     STORING PERTURBED THE PLAIN, WEIGHTED AND NORMLIZED RESIDUARL
C     VECTORS
      KK=KK+1
      DO 667 I=1,NM
      CONNN1(I,KK)=CON(I)
      CONNN2(I,KK)=WGH(I)*CON(I)
      CONNN3(I,KK)=NOMLC(I)*CON(I)

  667 CONTINUE

C     WHEN FOLLOWING STATEMTNT IS USED,  ONLY THE ORDERS OF PLAIN, WEIGHTED
C     AND NORMALIZED RESIDUALS WILL BE GIVEN IN FORROWING THREE CALL STATEMENT
       IF(KK.GT.1)GO TO 999

C     MAKING THE ORDERS OF THE PLAIN, WEIGHTED
C     AND NORMALIZED RESIDUALS
        CALL SORTGREAT(CONNN1,CONNN10,NM)
   CALL SORTGREAT(CONNN2,CONNN20,NM)
   CALL SORTGREAT(CONNN3,CONNN30,NM)

c       WHEN PSSEPF IS USED
  999  IF(KK.LT.N) GO TO 666


C       CALCULATING THE AVERAGES OF PLAIN RESIDUALS
C       CALCULATING THE STANDARD DEVIATION OF THE PLAIN RESIDUALS
C       CALCULATING THE SIGMA OVER THE AVERAGE OF THE PLAIN RESIDUALS
C       CALCULATING THE SIGMA TIMES THE AVERAGE OF THE PLAIN RESIDUALS
        CALL INDCAL(CONNN1,NM,N,A,AVEI,SIGI,AVE,SIG,SIGOAVE,SIGtAVE)



C       MAKING THE ORDERS OF PLAIN RESIDUALS, THE AVERAGES OF PLAIN
C       RESIDUALS, THE STANDARD DEVIATION OF THE PLAIN RESIDUALS
C       THE SIGMA OVER THE AVERAGE OF THE PLAIN RESIDUALS
C       THE SIGMA TIMES THE AVERAGE OF THE PLAIN RESIDUALS
CALL SORTGREAT(AVE,AVEO,NM)
        CALL SORTGREAT(SIG,SIGO,NM)
        CALL SORT(SIGOAVE,SIGOAVEO,NM)
```

```
        CALL SORTGREAT(SIGtAVE,SIGtAVEO,NM)

C       PRINTING PLAIN RESIDUALS, THE AVERAGES OF PLAIN
C       RESIDUALS, THE STANDARD DEVIATION OF THE PLAIN RESIDUALS
C       THE SIGMA OVER THE AVERAGE OF THE PLAIN RESIDUALS
C       THE SIGMA TIMES THE AVERAGE OF THE PLAIN RESIDUALS
c        WRITE(50,981)(I,CONNN1O(I,1),CONNN1O(I,2),AVEO(I,1),AVEO(I,2),
c     1SIGO(I,1),SIGO(I,2),SIGOAVEO(I,1),SIGOAVEO(I,2),
c     1SIGtAVEO(I,1),SIGtAVEO(I,2),I=1,10)


C       PRINTING THE LARGEST OR SMALLEST PLAIN RESIDUALS,
C       THE AVERAGES OF PLAIN RESIDUALS, THE STANDARD
C       DEVIATION OF THE PLAIN RESIDUALS
C       THE SIGMA OVER THE AVERAGE OF THE PLAIN RESIDUALS
C       THE SIGMA TIMES THE AVERAGE OF THE PLAIN RESIDUALS
c       WRITE(50,981)(IN,CONNN1O(1,1),CONNN1O(1,2),AVEO(1,1),AVEO(1,2),
c     1SIGO(1,1),SIGO(1,2),SIGOAVEO(1,1),SIGOAVEO(1,2),
c     1SIGtAVEO(1,1),SIGtAVEO(1,2))


  981   FORMAT(4X,I3,1X,F4.0,1x,F8.4,1X,F4.0,1x,F8.4,1X,F4.0,1x,F8.4,
     11X,F4.0,1x,F8.4,1X,F4.0,1x,F8.4)


C       CALCULATING THE AVERAGES OF WEIGHTD RESIDUALS
C       CALCULATING THE STANDARD DEVIATION OF THE WEIGHTED RESIDUALS
C       CALCULATING THE SIGMA OVER THE AVERAGE THE WEIGHTED RESIDUALS
C       CALCULATING THE SIGMA TIME THE AVERAGE THE WEIGHTED RESIDUALS
        CALL INDCAL(CONNN2,NM,N,A,AVEI,SIGI,AVE,SIG,SIGOAVE,SIGtAVE)

C       MAKING THE ORDERS OF WEIGHTED RESIDUALS, THE AVERAGES OF WEIGHTED
C       RESIDUALS, THE STANDARD DEVIATION OF THE WEIGHTED RESIDUALS
C       THE SIGMA OVER THE AVERAGE OF THE WEIGHTED RESIDUALS
C       THE SIGMA TIMES THE AVERAGE OF THE WEIGHTED RESIDUALS
CALL SORTGREAT(Ave,AVEO,NM)
        CALL SORTGREAT(SIG,SIGO,NM)
        CALL SORT(SIGOAVE,SIGOAVEO,NM)
        CALL SORTGREAT(SIGtAVE,SIGtAVEO,NM)

C       PRINTING WEIGHTED RESIDUALS, THE AVERAGES OF WEIGHTED
C       RESIDUALS, THE STANDARD DEVIATION OF THE WEIGHTED RESIDUALS
C       THE SIGMA OVER THE AVERAGE OF THE WEIGHTED RESIDUALS
C       THE SIGMA TIMES THE AVERAGE OF THE WEIGHTED RESIDUALS
        WRITE(50,981)(I,CONNN2O(I,1),CONNN2O(I,2),AVEO(I,1),AVEO(I,2),
```

```
     1SIGO(I,1),SIGO(I,2),SIGOAVEO(I,1),SIGOAVEO(I,2),
     1SIGtAVEO(I,1),SIGtAVEO(I,2),I=1,10)


C       PRINTING THE LARGEST OR SMALLEST WEIGHTED RESIDUALS,
C       THE AVERAGES OF WEIGHTED RESIDUALS, THE STANDARD
C       DEVIATION OF THE WEIGHTED RESIDUALS
C       THE SIGMA OVER THE AVERAGE OF THE WEIGHTED RESIDUALS
C       THE SIGMA TIMES THE AVERAGE OF THE WEIGHTED RESIDUALS
        WRITE(50,981)(IN,CONNN2O(1,1),CONNN2O(1,2),AVEO(1,1),AVEO(1,2),
     1SIGO(1,1),SIGO(1,2),SIGOAVEO(1,1),SIGOAVEO(1,2),
     1SIGtAVEO(1,1),SIGtAVEO(1,2))


C       CALCULATING THE AVERAGES OF THE NORMLIZED RESIDUALS
C       CALCULATING THE STANDARD DEVIATION OF THE NORMLIZED RESIDUALS
C       CALCULATING THE SIGMA OVER THE AVERAGE THE NORMLIZED RESIDUALS
C       CALCULATING THE SIGMA TIME THE AVERAGE THE NORMLIZED RESIDUALS
        CALL INDCAL(CONNN3,NM,N,A,AVEI,SIGI,AVE,SIG,SIGOAVE,SIGtAVE)


C       MAKING THE ORDERS OF NORMALIZED RESIDUALS, THE AVERAGES OF
C       THE AVERAGES OF NORMALIZED RESIDUALS, THE STANDARD DEVIATION
C       OF THE NORMALIZED RESIDUALS, THE SIGMA OVER THE AVERAGE
C       OF THE NORMALIZED RESIDUALS,
C       THE SIGMA TIMES THE AVERAGE OF THE NORMALIZED RESIDUALS
CALL SORTGREAT(Ave,AVEO,NM)
        CALL SORTGREAT(SIG,SIGO,NM)
        CALL SORT(SIGOAVE,SIGOAVEO,NM)
        CALL SORTGREAT(SIGtAVE,SIGtAVEO,NM)



C       PRINTING NORMALIZED RESIDUALS, THE AVERAGES OF
C       THE AVERAGES OF NORMALIZED RESIDUALS, THE STANDARD DEVIATION
C       OF THE NORMALIZED RESIDUALS, THE SIGMA OVER THE AVERAGE
C       OF THE NORMALIZED RESIDUALS,
C       THE SIGMA TIMES THE AVERAGE OF THE NORMALIZED RESIDUALS
        WRITE(50,981)(I,CONNN3O(I,1),CONNN3O(I,2),AVEO(I,1),
     1AVEO(I,2),SIGO(I,1),SIGO(I,2),SIGOAVEO(I,1),SIGOAVEO(I,2),
     1SIGtAVEO(I,1),SIGtAVEO(I,2),I=1,nm)

C      PRINTING THE LARGEST OR SMALLEST NORMALIZED RESIDUALS, THE AVERAGES OF
C      THE AVERAGES OF NORMALIZED RESIDUALS, THE STANDARD DEVIATION
C      OF THE NORMALIZED RESIDUALS, THE SIGMA OVER THE AVERAGE
C      OF THE NORMALIZED RESIDUALS,
C      THE SIGMA TIMES THE AVERAGE OF THE NORMALIZED RESIDUALS
       WRITE(50,981)(IN,CONNN3O(1,1),CONNN3O(1,2),AVEO(1,1),AVEO(1,2),
```

```
     1SIGO(1,1),SIGO(1,2),SIGOAVEO(1,1),SIGOAVEO(1,2),
     1SIGtAVEO(1,1),SIGtAVEO(1,2))


c      BAD DATA POINTS SELECTION
         AM(IINN)=IIN
         IF(IINN.LT.122) GO TO 444


  888 STOP
      END
```

```
C*********************************************************************
      SUBROUTINE WMGP(AM,WMM,NMM,PNM,QNM,PLM,QLM,VM,NB,NNBB,NT,
     1ERR1,ERR2,FERR1,FERR2,WMN)
C*********************************************************************
C
C     THIS SUBROUTINE IS USED TO PRODUCE THE WEIGHTING MATRIX.
C     THE SUBROUTINE SHOwmgWN HERE IS TO CREAT WEIGHTING MATRIX ${\bf WU}$.
C
C     PARAMETERS PASSED:
C         AM = ARRAY OF INDICES SPECIFYING WHETHER CORRESPONDING MEASUREMENTS
C              ARE AVALIABLE OR NOT AND SPECIFY THEIR ACCURACY CATEGORIES
C         NMM = NUMBER OF SYSTEM VARIABLES (= NNBB + 2 X NT + NB)
C         PNM = ARRAY OF SIMULATED NODAL REAL POWER INJECTIONS
C         QNM = ARRAY OF SIMULATED NODAL REACTIVE POWER INJECTIONS
C         PLM = ARRAY OF SIMULATED REAL POWER POWER FLOWS
C         QLM = ARRAY OF SIMULATED REACTIVE POWER FLOWS
C         VM = ARRAY OF SIMULATED MAGNITUDES OF NODAL VOLTAGES
C         NB = NUMBER OF BUSES
C         NNBB = 2 X NB
C         NT = 2 X NLINES
C         ERR1 = ERROR PROPORTIONAL TO MEASUREMENTS FOR VOLTAGE MEASUREMENTS
C         ERR2 = ERROR PROPORTIONAL TO MEASUREMENTS FOR OTHER MEASUREMENTS
C         FERR1 = FIXED ERROR FOR VOLTAGE MEASUREMENTS
C         FERR2 = FIXED ERROR OTHER MEASUREMENTS
C
C     PARAMETERS RETURNED:
C         WMG = WEIGHTING MATRIX USED IN PSSE
C
C     CALLED BY:
C         MAIN PROGRAM
C
C     CALLS:
C         NONE
C
C*********************************************************************



      REAL WMM(NMM,NMM),PNM(NB),QNM(NB),PLM(NT),QLM(NT),VM(NB)

      INTEGER AM(NMM),WMN

      DO 800 I=1,NMM
      DO 800 J=1,NMM
```

```
    WMM(I,J)=0.0
800 CONTINUE

    IF (WMN.EQ.1) GO TO 100

    IF (WMN.EQ.12) GO TO 500

    DO 32 I=1,NB
    IF (AM(I).EQ.0) THEN
    ERR=ERR2
    FERR=FERR2
    ELSE IF (AM(I).EQ.1) THEN
    ERR=ERR2
    FERR=FERR2
    ELSE IF (AM(I).EQ.2) THEN
    ERR=2*ERR2
    FERR=2*FERR2
    ELSE IF (AM(I).EQ.3) THEN
    ERR=3*ERR2
    FERR=3*FERR2
    ELSE IF (AM(I).EQ.4) THEN
    ERR=4*ERR2
    FERR=4*FERR2
    ELSE IF (AM(I).EQ.5) THEN
    SIG=ERR2
    FERR=FERR2
    ELSE IF (AM(I).EQ.6) THEN
    SIG=2*ERR2
    FERR=2*FERR2
    ELSE IF (AM(I).EQ.7) THEN
    SIG=3*ERR2
    FERR=3*FERR2
    ELSE
    SIG=4*ERR2
    FERR=4*FERR2

    ENDIF


    C=(1/(ABS(PNM(I))*ERR+FERR))
    CC=(ABS(PNM(I)))

    IF (WMN.EQ.2) THEN
    WMM(I,I)=C
    ELSE IF (WMN.EQ.3) THEN
    WMM(I,I)=C*C
    ELSE IF (WMN.EQ.4) THEN
    WMM(I,I)=C*C*CC
```

```
ELSE IF (WMN.EQ.5) THEN
WMM(I,I)=C*C*CC*CC
ELSE IF (WMN.EQ.6) THEN
WMM(I,I)=C*CC
ELSE IF (WMN.EQ.7) THEN
WMM(I,I)=C*CC*CC
ELSE IF (WMN.EQ.8) THEN
WMM(I,I)=C/(CC*CC)
ELSE IF (WMN.EQ.9) THEN
WMM(I,I)=C*C/(CC*CC)
ELSE IF (WMN.EQ.10) THEN
WMM(I,I)=C/CC
ELSE
WMM(I,I)=C*C/CC
ENDIF

IF (AM(I+NB).EQ.0) THEN
ERR=ERR2
FERR=FERR2
ELSE IF (AM(I+NB).EQ.1) THEN
ERR=ERR2
FERR=FERR2
ELSE IF (AM(I+NB).EQ.2) THEN
ERR=2*ERR2
FERR=2*FERR2
ELSE IF (AM(I+NB).EQ.3) THEN
ERR=3*ERR2
FERR=3*FERR2
ELSE IF (AM(I+NB).EQ.4) THEN
ERR=4*ERR2
FERR=4*FERR2
ELSE IF (AM(I+NB).EQ.5) THEN
SIG=ERR2
FERR=FERR2
ELSE IF (AM(I+NB).EQ.6) THEN
SIG=2*ERR2
FERR=2*FERR2
ELSE IF (AM(I+NB).EQ.7) THEN
SIG=3*ERR2
FERR=3*FERR2
ELSE
SIG=4*ERR2
FERR=4*FERR2

ENDIF

D=(1/(ABS(QNM(I))*ERR+FERR))
DD=(ABS(QNM(I)))
```

```
      IF (WMN.EQ.2) THEN
      WMM(I+NB,I+NB)=D
      ELSE IF (WMN.EQ.3) THEN
      WMM(I+NB,I+NB)=D*D
      ELSE IF (WMN.EQ.4) THEN
      WMM(I+NB,I+NB)=D*D*DD
      ELSE IF (WMN.EQ.5) THEN
      WMM(I+NB,I+NB)=D*D*DD*DD
      ELSE IF (WMN.EQ.6) THEN
      WMM(I+NB,I+NB)=D*DD
      ELSE IF (WMN.EQ.7) THEN
      WMM(I+NB,I+NB)=D*DD*DD
      ELSE IF (WMN.EQ.8) THEN
      WMM(I+NB,I+NB)=D/(DD*DD)
      ELSE IF (WMN.EQ.9) THEN
      WMM(I+NB,I+NB)=D*D/(DD*DD)
      ELSE IF (WMN.EQ.10) THEN
      WMM(I+NB,I+NB)=D/DD
      ELSE
      WMM(I+NB,I+NB)=D*D/DD
      ENDIF

   32 CONTINUE

      DO 34 I=1,NT

      IF (AM(I+NNBB).EQ.0) THEN
      ERR=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB).EQ.1) THEN
      ERR=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB).EQ.2) THEN
      ERR=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I+NNBB).EQ.3) THEN
      ERR=3*ERR2
      FERR=3*FERR2
      ELSE IF (AM(I+NNBB).EQ.4) THEN
      ERR=4*ERR2
      FERR=4*FERR2
      ELSE IF (AM(I+NNBB).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
```

```
ELSE IF (AM(I+NNBB).EQ.7) THEN
SIG=3*ERR2
FERR=3*FERR2
ELSE
SIG=4*ERR2
FERR=4*FERR2
ENDIF



E=(1/(ABS(PLM(I))*ERR+FERR))
EE=(ABS(PLM(I)))

IF (WMN.EQ.2) THEN
WMM(I+NNBB,I+NNBB)=E
ELSE IF (WMN.EQ.3) THEN
WMM(I+NNBB,I+NNBB)=E*E
ELSE IF (WMN.EQ.4) THEN
WMM(I+NNBB,I+NNBB)=E*E*EE
ELSE IF (WMN.EQ.5) THEN
WMM(I+NNBB,I+NNBB)=E*E*EE*EE
ELSE IF (WMN.EQ.6) THEN
WMM(I+NNBB,I+NNBB)=E*EE
ELSE IF (WMN.EQ.7) THEN
WMM(I+NNBB,I+NNBB)=E*EE*EE
ELSE IF (WMN.EQ.8) THEN
WMM(I+NNBB,I+NNBB)=E/(EE*EE)
ELSE IF (WMN.EQ.9) THEN
WMM(I+NNBB,I+NNBB)=E*E/(EE*EE)
ELSE IF (WMN.EQ.10) THEN
WMM(I+NNBB,I+NNBB)=E/EE
ELSE
WMM(I+NNBB,I+NNBB)=E*E/EE
ENDIF

IF (AM(I+NNBB+NT).EQ.0) THEN
ERR=ERR2
FERR=FERR2
ELSE IF (AM(I+NNBB+NT).EQ.1) THEN
ERR=ERR2
FERR=FERR2
ELSE IF (AM(I+NNBB+NT).EQ.2) THEN
ERR=2*ERR2
FERR=2*FERR2
ELSE IF (AM(I+NNBB+NT).EQ.3) THEN
ERR=3*ERR2
FERR=3*FERR2
```

```
      ELSE IF (AM(I+NNBB+NT).EQ.4) THEN
      ERR=4*ERR2
      FERR=4*FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      ELSE
      SIG=4*ERR2
      FERR=4*FERR2
      ENDIF




      A=(1/(ABS(QLM(I))*ERR+FERR))
      AA=(ABS(QLM(I)))

      IF (WMN.EQ.2) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A
      ELSE IF (WMN.EQ.3) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A*A
      ELSE IF (WMN.EQ.4) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A*A*AA
      ELSE IF (WMN.EQ.5) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A*A*AA*AA
      ELSE IF (WMN.EQ.6) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A*AA
      ELSE IF (WMN.EQ.7) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A*AA*AA
      ELSE IF (WMN.EQ.8) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A/(AA*AA)
      ELSE IF (WMN.EQ.9) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A*A/(AA*AA)
      ELSE IF (WMN.EQ.10) THEN
      WMM(I+NNBB+NT,I+NNBB+NT)=A/AA
      ELSE
      WMM(I+NNBB+NT,I+NNBB+NT)=A*A/AA
      ENDIF

34    CONTINUE
```

```
DO 36 I=1,NB

IF (AM(I+NNBB+NT+NT).EQ.0) THEN
ERR=ERR1
FERR=FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.1) THEN
ERR=ERR1
FERR=FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.2) THEN
ERR=2*ERR1
FERR=2*FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.3) THEN
ERR=3*ERR1
FERR=3*FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.4) THEN
ERR=4*ERR1
FERR=4*FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.5) THEN
SIG=ERR1
FERR=FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
SIG=2*ERR1
FERR=2*FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
SIG=3*ERR1
FERR=3*FERR1
ELSE
SIG=4*ERR1
FERR=4*FERR1
ENDIF



B=(1/(ABS(VM(I))*ERR+FERR))
BB=(ABS(VM(I)))

IF (WMN.EQ.2) THEN
WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B
ELSE IF (WMN.EQ.3) THEN
WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B*B
ELSE IF (WMN.EQ.4) THEN
WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B*B*BB
ELSE IF (WMN.EQ.5) THEN
WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B*B*BB*BB
ELSE IF (WMN.EQ.6) THEN
WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B*BB
```

```
      ELSE IF (WMN.EQ.7) THEN
      WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B*BB*BB
      ELSE IF (WMN.EQ.8) THEN
      WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B/(BB*BB)
      ELSE IF (WMN.EQ.9) THEN
      WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B*B/(BB*BB)
      ELSE IF (WMN.EQ.10) THEN
      WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B/BB
      ELSE
      WMM(I+NNBB+NT+NT,I+NNBB+NT+NT)=B*B/BB
      ENDIF

   36 CONTINUE

      DO 210 I=1,NMM
      WMM(I,I)=WMM(I,I)*0.001
  210 CONTINUE
      GO TO 999

  500 DO 560 I =1,NMM
      IF (AM(I).EQ.0) GO TO 550
      IF (AM(I).EQ.1) THEN
      WMM(I,I)=2
      ELSE
      WMM(I,I)=1
      ENDIF
      GO TO 560
  550 WMM(I,I)=1
  560 CONTINUE

      GO TO 999

  100 DO 209 I=1,NMM
      WMM(I,I)=1.
  209 CONTINUE

  999 RETURN
      END


C*********************************************************************
      SUBROUTINE MGBP(AM,NUDR,VM,VT,DT,E,MTOBO,NT,NB,ST,ZLINE,YC,NLINES,
     1PLM,QLM,PLT,QLT,PNM,QNM,P,Q,NDR,NMM,NNBB,ERR1,ERR2,FERR1,FERR2,IR,
     1nbb)
C*********************************************************************
C
C     THIS SUBROUTINE IS USED TO GENERATE ALL THE SIMULATED SYSTEM
C     VARIABLES WHICH ARE SIMULATED MEASUREMENTS.
```

```
C
C     PARAMETERS PASSED:
C         AM = ARRAY OF INDICES SPECIFY WHETHER CORRESPONDING MEASUREMENTS
C              ARE AVALIABLE OR NOT AND SPECIFY THEIR ACCURACY CATEGORIES
C         NUDR = NUMBER OF UNIFORMALY DISTRIBUTED RANDOM NUMBERS USED TO
C                GENERATE NORMALLY DISTRIBUTED RANDOM NUMBERS
C         VT = ARRAY OF TRUE MAGNITUDES OF NODAL VOLTAGES
C         DT = ARRAY OF TRUE ANGLES OF NODAL VOLTAGES
C         E = ARRAY OF NODAL VOLTAGES
C         MTOBO = ARRAY OF: #, FROM_BUS, TO_BUS, LINE #
C         NT = 2 X NLINES
C         NB = NUMBER OF BUSES
C         ST = ARRAY OF COMPLEX LINE FLOWS
C         ZLINE = ARRAY OF LINE IMPEDANCES
C         YC = ARRAY OF LINE ADMITTANCES
C         NLINES = NUMBER OF LINES
C         PLT = ARRAY OF TRUE REAL POWER FLOWS
C         QLT = ARRAY OF TRUE REACTIVE POWER FLOWS
C         P = ARRAY OF TRUE NODAL REAL POWER INJECTIONS
C         Q = ARRAY OF TRUE NODAL REACTIVE POWER INJECTIONS
C         NMM = NUMBER OF SYSTEM VARIABLES (= NNBB + 2 X NT + NB)
C         NNBB = 2 X NB
C         ERR1 = ERROR PROPORTIONAL TO MEASUREMENTS FOR VOLTAGE MEASUREMENTS
C         ERR2 = ERROR PROPORTIONAL TO MEASUREMENTS FOR OTHER MEASUREMENTS
C         FERR1 = FIXED ERROR FOR VOLTAGE MEASUREMENTS
C         FERR2 = FIXED ERROR OTHER MEASUREMENTS
C         IR = A SEED USED TO GENERATE RANDOM NUMBERS
C
C     PARAMETERS RETURNED:
C         PNM = ARRAY OF SIMULATED NODAL REAL POWER INJECTIONS
C         QNM = ARRAY OF SIMULATED NODAL REACTIVE POWER INJECTIONS
C         PLM = ARRAY OF SIMULATED REAL POWER POWER FLOWS
C         QLM = ARRAY OF SIMULATED REACTIVE POWER FLOWS
C         VM = ARRAY OF SIMULATED MAGNITUDES OF NODAL VOLTAGES
C
C     CALLED BY:
C         MAIN PROGRAM
C
C     CALLS:
C         SUBROUTINE NRAN
C         SUBROUTINE RANDOM
C         SUBROUTINE CLF
C
C*********************************************************************

      REAL VM(NB),VT(NB),DT(NB),PLM(NB),QLM(NB),PNM(NB),QNM(NB),
     1PLT(NLINES),QLT(NLINES),P(NB),Q(NB),NDR
```

```
COMPLEX E(NB),ST(NT),ZLINE(NLINES),YC(NLINES)

INTEGER MTOBO(NT,4),AM(NMM),IR

DO 201 I = 1, NB

IF (AM(I+NNBB+NT+NT).EQ.0) THEN
SIG=ERR1
FERR=FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.1) THEN
SIG=ERR1
FERR=FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.2) THEN
SIG=2*ERR1
FERR=2*FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.3) THEN
SIG=3*ERR1
FERR=3*FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.4) THEN
SIG=4*ERR1
FERR=4*FERR1
ELSE IF (AM(I+NNBB+NT+NT).EQ.5) THEN
SIG=ERR1
FERR=FERR1
GO TO 191
ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
SIG=2*ERR1
FERR=2*FERR1
GO TO 191
ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
SIG=3*ERR1
FERR=3*FERR1
GO TO 191
ELSE
SIG=4*ERR1
FERR=4*FERR1
GO TO 191
ENDIF

CALL NRAN(NUDR,NDR,SIG,IR)
IF(NDR.GT.(3*SIG))THEN NDR=3*SIG
IF(NDR.LT.(-3*SIG))THEN NDR=-3*SIG
VM(I) = VT(I)*(1 + NDR)

CALL RANDOM(IR,RAN)
IF(RAN.GT.(0.0)) THEN
VM(I) =( VM(I) + FERR)*(1+0.005*RAN)
ELSE
```

```
      VM(I) =( VM(I) - FERR)*(1-0.005*RAN)
      ENDIF
      go to 201
c  191 VM(I)=1*VT(I)

c  191 VM(I)=0.0
  191 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      VM(I)=(VT(I)+nbb*SIG*VT(I)+FERR)*(1+0.005*RAN)
      ELSE
      VM(I)=(VT(I)-nbb*SIG*VT(I)-FERR)*(1-0.005*RAN)
      ENDIF

  201 CONTINUE

      DO 8 I = 1,NB
      AA = VT(I)*COS(DT(I))
      BB = VT(I)*SIN(DT(I))
      E(I) = CMPLX(AA,BB)
    8 CONTINUE

      CALL CLF(MTOBO,NT,E,NB,ST,ZLINE,YC,NLINES)

C     LINE FLOW MEASUREMENTS GENERATION

      DO 20 I = 1, NT

      IF (AM(I+NNBB).EQ.0) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB).EQ.1) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB).EQ.2) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I+NNBB).EQ.3) THEN
      SIG=3*ERR2
      FERR=3*FERR2

      ELSE IF (AM(I+NNBB).EQ.4)  THEN
      SIG=4*ERR2
      FERR=4*FERR2
      ELSE IF (AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      GO TO 192
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
```

```
      SIG=2*ERR2
      FERR=2*FERR2
      GO TO 192
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      GO TO 192
      ELSE
      SIG=4*ERR2
      FERR=4*FERR2
      GO TO 192
      ENDIF

      CALL NRAN(NUDR,NDR,SIG,IR)
      IF (NDR.GT.(3*SIG)) THEN NDR=3*SIG
      IF (NDR.LT.(-3*SIG)) THEN NDR=-3*SIG
      PLT(I)=REAL(ST(I))
      PLM(I) = REAL(ST(I))*(1 + NDR)

      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      PLM(I) =( PLM(I) + FERR)*(1+0.005*RAN)
      ELSE
      PLM(I) =( PLM(I) - FERR)*(1-0.005*RAN)
      ENDIF
      go to 292
c  192 PLM(I)=1*REAL(ST(I))

c  192 PLM(I)=0.0
  192 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      PLM(I)=(REAL(ST(I))+nbb*SIG*REAL(ST(I))+FERR)*(1+0.005*RAN)
      ELSE
      PLM(I)=(REAL(ST(I))-nbb*SIG*REAL(ST(I))-FERR)*(1-0.005*RAN)
      ENDIF
      PLT(I)=REAL(ST(I))

  292 IF (AM(I+NNBB+NT).EQ.0) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.1) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.2) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.3) THEN
      SIG=3*ERR2
```

```
      FERR=3*FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.4) THEN
      SIG=4*ERR2
      FERR=4*FERR2
      ELSE IF (AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      GO TO 193
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      GO TO 193
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      GO TO 193
      ELSE
      SIG=4*ERR2
      FERR=4*FERR2
      GO TO 193
      ENDIF


      CALL NRAN(NUDR,NDR,SIG,IR)
      IF (NDR.GT.(3*SIG)) THEN NDR=3*SIG
      IF (NDR.LT.(-3*SIG))THEN NDR=-3*SIG
      QLT(I) = AIMAG(ST(I))
      QLM(I) = AIMAG(ST(I))*(1 + NDR)

      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      QLM(I) =( QLM(I) + FERR)*(1+0.005*RAN)
      ELSE
      QLM(I) =( QLM(I) - FERR)*(1-0.005*RAN)
      ENDIF
      go to 20
c  193  QLM(I)=1*AIMAG(ST(I))


c  193 QLM(I)=0.0
  193 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      QLM(I)=(AIMAG(ST(I))+nbb*SIG*AIMAG(ST(I))+FERR)*(1+0.005*RAN)
      ELSE
      QLM(I)=(AIMAG(ST(I))-nbb*SIG*AIMAG(ST(I))-FERR)*(1-0.005*RAN)
      ENDIF
      QLT(I) = AIMAG(ST(I))
```

```
   20 CONTINUE

C     NODAL MEASUREMENTS GENERATION

      DO 22 I = 1, NB

      IF (AM(I).EQ.0) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I).EQ.1) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I).EQ.2) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I).EQ.3) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      ELSE IF (AM(I).EQ.4) THEN
      SIG=4*ERR2
      FERR=4*FERR2
      ELSE IF(AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      GO TO 194
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      GO TO 194
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      GO TO 194
      ELSE
      SIG=4*ERR2
      FERR=4*FERR2
      GO TO 194
      ENDIF

      CALL NRAN(NUDR,NDR,SIG,IR)
      IF (NDR.GT.(3*SIG)) THEN NDR=3*SIG
      IF (NDR.LT.(-3*SIG)) THEN NDR=-3*SIG
      PNM(I) = P(I)*(1 + NDR)
c     PNM(I) = P(I)
      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      PNM(I) =( PNM(I) + FERR)*(1+0.005*RAN)
```

```
      ELSE
      PNM(I) =( PNM(I) - FERR)*(1-0.005*RAN)
      ENDIF
      GO to 294
c  194 PNM(I)=1*P(I)

  194 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      PNM(I)=(P(I)+nbb*SIG*P(I)+FERR)*(1+0.005*RAN)
      ELSE
      PNM(I)=(P(I)-nbb*SIG*P(I)-FERR)*(1-0.005*RAN)
      ENDIF

  294 IF (AM(I+NB).EQ.0) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NB).EQ.1) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NB).EQ.2) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I+NB).EQ.3) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      ELSE IF (AM(I+NB).EQ.4) THEN
      SIG=4*ERR2
      FERR=4*FERR2
      ELSE IF(AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      GO TO 195
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      GO TO 195
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      GO TO 195
      ELSE
      SIG=4*ERR2
      FERR=4*FERR2
      GO TO 195
      ENDIF

      CALL NRAN(NUDR,NDR,SIG,IR)
      IF (NDR.GT.(3*SIG)) THEN NDR=3*SIG
```

```
      IF (NDR.LT.(-3*SIG))THEN NDR=-3*SIG
      QNM(I) = Q(I)*(1 + NDR)
c      QNM(I) = Q(I)
      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      QNM(I) =( QNM(I) + FERR)*(1+0.005*RAN)
      ELSE
      QNM(I) =( QNM(I) - FERR)*(1+0.005*RAN)
      ENDIF
      go to 22
c  195 QNM(I)=1*Q(I)

  195 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      QNM(I)=(Q(I)+nbb*SIG*Q(I)+FERR)*(1+0.005*RAN)
      ELSE
      QNM(I)=(Q(I)-nbb20*SIG*Q(I)-FERR)*(1-0.005*RAN)
      ENDIF

   22 CONTINUE

      RETURN
      END
C*************************************************************************
      SUBROUTINE MGB(AM,NUDR,VM,VT,DT,E,MTOBO,NT,NB,ST,ZLINE,YC,NLINES,
     1PLM,QLM,PLT,QLT,PNM,QNM,P,Q,NDR,NMM,NNBB,ERR1,ERR2,FERR1,FERR2,IR,
     1nbb)
C*************************************************************************
C
C     THIS SUBROUTINE IS USED TO GENERATE ALL THE SIMULATED SYSTEM
C     VARIABLES WHICH ARE SIMULATED MEASUREMENTS.
C
C     PARAMETERS PASSED:
C         AM = ARRAY OF INDICES SPECIFY WHETHER CORRESPONDING MEASUREMENTS
C              ARE AVALIABLE OR NOT AND SPECIFY THEIR ACCURACY CATEGORIES
C         NUDR = NUMBER OF UNIFORMALY DISTRIBUTED RANDOM NUMBERS USED TO
C                GENERATE NORMALLY DISTRIBUTED RANDOM NUMBERS
C         VT = ARRAY OF TRUE MAGNITUDES OF NODAL VOLTAGES
C         DT = ARRAY OF TRUE ANGLES OF NODAL VOLTAGES
C         E = ARRAY OF NODAL VOLTAGES
C         MTOBO = ARRAY OF: #, FROM_BUS, TO_BUS, LINE #
C         NT = 2 X NLINES
C         NB = NUMBER OF BUSES
C         ST = ARRAY OF COMPLEX LINE FLOWS
C         ZLINE = ARRAY OF LINE IMPEDANCES
C         YC = ARRAY OF LINE ADMITTANCES
C         NLINES = NUMBER OF LINES
C         PLT = ARRAY OF TRUE REAL POWER FLOWS
```

```
C          QLT = ARRAY OF TRUE REACTIVE POWER FLOWS
C          P = ARRAY OF TRUE NODAL REAL POWER INJECTIONS
C          Q = ARRAY OF TRUE NODAL REACTIVE POWER INJECTIONS
C          NMM = NUMBER OF SYSTEM VARIABLES (= NNBB + 2 X NT + NB)
C          NNBB = 2 X NB
C          ERR1 = ERROR PROPORTIONAL TO MEASUREMENTS FOR VOLTAGE MEASUREMENTS
C          ERR2 = ERROR PROPORTIONAL TO MEASUREMENTS FOR OTHER MEASUREMENTS
C          FERR1 = FIXED ERROR FOR VOLTAGE MEASUREMENTS
C          FERR2 = FIXED ERROR OTHER MEASUREMENTS
C          IR = A SEED USED TO GENERATE RANDOM NUMBERS
C
C     PARAMETERS RETURNED:
C          PNM = ARRAY OF SIMULATED NODAL REAL POWER INJECTIONS
C          QNM = ARRAY OF SIMULATED NODAL REACTIVE POWER INJECTIONS
C          PLM = ARRAY OF SIMULATED REAL POWER POWER FLOWS
C          QLM = ARRAY OF SIMULATED REACTIVE POWER FLOWS
C          VM = ARRAY OF SIMULATED MAGNITUDES OF NODAL VOLTAGES
C
C     CALLED BY:
C          MAIN PROGRAM
C
C     CALLS:
C          SUBROUTINE NRAN
C          SUBROUTINE RANDOM
C          SUBROUTINE CLF
C
C*********************************************************************

      REAL VM(NB),VT(NB),DT(NB),PLM(NB),QLM(NB),PNM(NB),QNM(NB),
     1PLT(NLINES),QLT(NLINES),P(NB),Q(NB),NDR

      COMPLEX E(NB),ST(NT),ZLINE(NLINES),YC(NLINES)

      INTEGER MTOBO(NT,4),AM(NMM),IR

      DO 201 I = 1, NB

      IF (AM(I+NNBB+NT+NT).EQ.0) THEN
      SIG=ERR1
      FERR=FERR1
      ELSE IF (AM(I+NNBB+NT+NT).EQ.1) THEN
      SIG=ERR1
      FERR=FERR1
      ELSE IF (AM(I+NNBB+NT+NT).EQ.2) THEN
      SIG=2*ERR1
      FERR=2*FERR1
      ELSE IF (AM(I+NNBB+NT+NT).EQ.3) THEN
      SIG=3*ERR1
```

```
      FERR=3*FERR1
      ELSE IF (AM(I+NNBB+NT+NT).EQ.4) THEN
      SIG=4*ERR1
      FERR=4*FERR1
      ELSE IF (AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR1
      FERR=FERR1
      GO TO 191
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
      SIG=2*ERR1
      FERR=2*FERR1
      GO TO 191
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR1
      FERR=3*FERR1
      GO TO 191
      ELSE
      SIG=4*ERR1
      FERR=4*FERR1
      GO TO 191
      ENDIF

      CALL NRAN(NUDR,NDR,SIG,IR)
      IF(NDR.GT.(3*SIG))THEN NDR=3*SIG
      IF(NDR.LT.(-3*SIG))THEN NDR=-3*SIG
      VM(I) = VT(I)*(1 + NDR)

      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      VM(I) =( VM(I) + FERR)
      ELSE
      VM(I) =( VM(I) - FERR)
      ENDIF
      go to 201
c  191 VM(I)=1*VT(I)

c  191 VM(I)=0.0
  191 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      VM(I)=(VT(I)+nbb*SIG*VT(I)+FERR)
      ELSE
      VM(I)=(VT(I)-nbb*SIG*VT(I)-FERR)
      ENDIF

  201 CONTINUE

      DO 8 I = 1,NB
      AA = VT(I)*COS(DT(I))
```

```
      BB = VT(I)*SIN(DT(I))
      E(I) = CMPLX(AA,BB)
    8 CONTINUE

      CALL CLF(MTOBO,NT,E,NB,ST,ZLINE,YC,NLINES)

C     LINE FLOW MEASUREMENTS GENERATION

      DO 20 I = 1, NT

      IF (AM(I+NNBB).EQ.0) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB).EQ.1) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB).EQ.2) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I+NNBB).EQ.3) THEN
      SIG=3*ERR2
      FERR=3*FERR2

      ELSE IF (AM(I+NNBB).EQ.4)  THEN
      SIG=4*ERR2
      FERR=4*FERR2
      ELSE IF (AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      GO TO 192
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      GO TO 192
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      GO TO 192
      ELSE
      SIG=4*ERR2
      FERR=4*FERR2
      GO TO 192
      ENDIF

      CALL NRAN(NUDR,NDR,SIG,IR)
      IF (NDR.GT.(3*SIG)) THEN NDR=3*SIG
      IF (NDR.LT.(-3*SIG)) THEN NDR=-3*SIG
      PLT(I)=REAL(ST(I))
```

```
      PLM(I) = REAL(ST(I))*(1 + NDR)

      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      PLM(I) =( PLM(I) + FERR)
      ELSE
      PLM(I) =( PLM(I) - FERR)
      ENDIF
      go to 292
c  192 PLM(I)=1*REAL(ST(I))

c  192 PLM(I)=0.0
  192 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      PLM(I)=(REAL(ST(I))+nbb*SIG*REAL(ST(I))+FERR)
      ELSE
      PLM(I)=(REAL(ST(I))-nbb*SIG*REAL(ST(I))-FERR)
      ENDIF
      PLT(I)=REAL(ST(I))

  292 IF (AM(I+NNBB+NT).EQ.0) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.1) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.2) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.3) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      ELSE IF (AM(I+NNBB+NT).EQ.4) THEN
      SIG=4*ERR2
      FERR=4*FERR2
      ELSE IF (AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      GO TO 193
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      GO TO 193
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      GO TO 193
      ELSE
```

```
      SIG=4*ERR2
      FERR=4*FERR2
      GO TO 193
      ENDIF


      CALL NRAN(NUDR,NDR,SIG,IR)
      IF (NDR.GT.(3*SIG)) THEN NDR=3*SIG
      IF (NDR.LT.(-3*SIG))THEN NDR=-3*SIG
      QLT(I) = AIMAG(ST(I))
      QLM(I) = AIMAG(ST(I))*(1 + NDR)

      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      QLM(I) =( QLM(I) + FERR)
      ELSE
      QLM(I) =( QLM(I) - FERR)
      ENDIF
      go to 20
c  193  QLM(I)=1*AIMAG(ST(I))


c  193 QLM(I)=0.0
  193 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      QLM(I)=(AIMAG(ST(I))+nbb*SIG*AIMAG(ST(I))+FERR)
      ELSE
      QLM(I)=(AIMAG(ST(I))-nbb*SIG*AIMAG(ST(I))-FERR)
      ENDIF
      QLT(I) = AIMAG(ST(I))


   20 CONTINUE

C     NODAL MEASUREMENTS GENERATION

      DO 22 I = 1, NB

      IF (AM(I).EQ.0) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I).EQ.1) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I).EQ.2) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I).EQ.3) THEN
```

```
      SIG=3*ERR2
      FERR=3*FERR2
      ELSE IF (AM(I).EQ.4) THEN
      SIG=4*ERR2
      FERR=4*FERR2
      ELSE IF(AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      GO TO 194
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      GO TO 194
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      GO TO 194
      ELSE
      SIG=4*ERR2
      FERR=4*FERR2
      GO TO 194
      ENDIF

      CALL NRAN(NUDR,NDR,SIG,IR)
      IF (NDR.GT.(3*SIG)) THEN NDR=3*SIG
      IF (NDR.LT.(-3*SIG)) THEN NDR=-3*SIG
      PNM(I) = P(I)*(1 + NDR)
c     PNM(I) = P(I)
      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      PNM(I) =( PNM(I) + FERR)
      ELSE
      PNM(I) =( PNM(I) - FERR)
      ENDIF
      GO to 294
c  194 PNM(I)=1*P(I)

  194 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      PNM(I)=(P(I)+nbb*SIG*P(I)+FERR)
      ELSE
      PNM(I)=(P(I)-nbb*SIG*P(I)-FERR)
      ENDIF

  294 IF (AM(I+NB).EQ.0) THEN
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NB).EQ.1) THEN
```

```
      SIG=ERR2
      FERR=FERR2
      ELSE IF (AM(I+NB).EQ.2) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      ELSE IF (AM(I+NB).EQ.3) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      ELSE IF (AM(I+NB).EQ.4) THEN
      SIG=4*ERR2
      FERR=4*FERR2
      ELSE IF(AM(I+NNBB+NT+NT).EQ.5) THEN
      SIG=ERR2
      FERR=FERR2
      GO TO 195
      ELSE IF (AM(I+NNBB+NT+NT).EQ.6) THEN
      SIG=2*ERR2
      FERR=2*FERR2
      GO TO 195
      ELSE IF (AM(I+NNBB+NT+NT).EQ.7) THEN
      SIG=3*ERR2
      FERR=3*FERR2
      GO TO 195
      ELSE
      SIG=4*ERR2
      FERR=4*FERR2
      GO TO 195
      ENDIF

      CALL NRAN(NUDR,NDR,SIG,IR)
      IF (NDR.GT.(3*SIG)) THEN NDR=3*SIG
      IF (NDR.LT.(-3*SIG))THEN NDR=-3*SIG
      QNM(I) = Q(I)*(1 + NDR)
c      QNM(I) = Q(I)
      CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      QNM(I) =( QNM(I) + FERR)
      ELSE
      QNM(I) =( QNM(I) - FERR)
      ENDIF
      go to 22
c  195 QNM(I)=1*Q(I)

  195 CALL RANDOM(IR,RAN)
      IF(RAN.GT.(0.0)) THEN
      QNM(I)=(Q(I)+nbb*SIG*Q(I)+FERR)
      ELSE
      QNM(I)=(Q(I)-nbb*SIG*Q(I)-FERR)
```

```
      ENDIF

   22 CONTINUE

      RETURN
      END


      SUBROUTINE RESDFCT(JKE,NM,IM,NNBB1,JI,JKET,JIJ,WM,SB,NOMLC,WGH)
      REAL JKE(NM,NNBB1),JKET(NNBB1,NM),IM(NNBB1,NNBB1),JI(NM,NNBB1)
      REAL JIJ(NM,NM),WM(NM,NM),SB(NM),NOMLC(NM),WGH(NM)

C     CALCULATING THE RESULT OF JKF*IM
      CALL XRR(JKE,NM,NNBB1,IM,NNBB1,JI)

C     CALCULATING THE RESULT OF JKF*IM
      CALL XRR(JI,NM,NNBB1,JKET,NM,JIJ)

C     CALCULATING THE FACTORS OF NORMALIZED AND WEIGHTED RESIDUALS
      DO 777 I=1, NM
      SB(I)=(1/WM(I,I))-JIJ(I,I)
  777 CONTINUE
      DO 780 I=1, NM
      IF(ABS(SB(I)).LT.0.00001)GO TO 778
      NOMLC(I)=sqrt(ABS(1/SB(I)))
      go to 780
  778 NOMLC(I)=1.0
  780 CONTINUE
      DO 779 I=1, NM
         WGH(I)=sqrt(ABS(WM(I,I)))
  779 CONTINUE
      RETURN
END



C       CALCULATING THE AVERAGES OF PLAIN RESIDUALS
C       CALCULATING THE STANDARD DEVIATION OF THE PLAIN RESIDUALS
C       CALCULATING THE SIGMA OVER THE AVERAGE OF THE PLAIN RESIDUALS
C       CALCULATING THE SIGMA TIMES THE AVERAGE OF THE PLAIN RESIDUALS
      SUBROUTINE INDCAL(CONNNI,NM,N,A,AVEI,SIGI,AVE,SIG,SIGOAVE,SIGtAVE)
        REAL CONNNI(NM,N),A(NM),AVEI,AVE(NM),SIGI,SIG(NM),SIGOAVE(NM)
        REAL SIGtAVE(NM)

   DO 165 I = 1,NM
DO 155 J = 1,N
A(J) = CONNNI(I,J)
```

```
  155 CONTINUE
CALL STAN(A,N,AVEI,SIGI)
AVE(I) = AVEI
SIG(I) = SIGI
       IF(ABS(AVEI).LT.0.00001)GO TO 160
SIGOAVE(I) = SIGI/AVEI
       SIGtAVE(I) = SIGI*AVEI
Go to 165
  160 sigoave(i)=99.99

  165 CONTINUE
       RETURN
       END
C*****************************************************************
      SUBROUTINE PSSE(V,D,NLINES,NY,YC,AM,WMM,NMM,PNM,QNM,PLM,QLM,
     1VM,NB,NNBB,NT,ERR1,ERR2,FERR1,FERR2,WMN,PNC,QNC,PLC,QLC,B,G,
     1MTOBO,E,ST,ZLINE,JK11,JK12,JK21,JK22,JK31,JK32,JK41,
     1JK42,JK51,JK52,JKEM,JKE,CON,CONM,WM,JKET,NNBB1,
     1AESTIM,ESTIM,CON1,IM,CON2,ITEST,CRIT,NM,KMAX,NB1)
c*****************************************************************
      REAL JK11(NB,NB),JK12(NB,NB),JK21(NB,NB),JK22(NB,NB)
      REAL JK31(NT,NB),JK32(NT,NB),JK41(NT,NB),JK42(NT,NB),
     1JK51(NB,NB),JK52(NB,NB),JKET(NNBB1,NM),JKE(NM,NNBB1)
      REAL JKEM(NMM,NNBB)
      REAL V(NB),D(NB),NY(NLINES),B(NB,NB),G(NB,NB)
      REAL CONM(NMM),WMM(NMM,NMM),WM(NM,NM)
      REAL AESTIM(NNBB1,NM),ESTIM(NNBB1,NNBB1),CON(NM),CON1(NNBB1),
     1CON2(NNBB1),IM(NNBB1,NNBB1)
      INTEGER MTOBO(NT,4),AM(NMM),KMAX

      REAL PNM(NB),QNM(NB),PLM(NT),QLM(NT),VM(NB)
      REAL PNC(NB),QNC(NB),PLC(NT),QLC(NT)

      COMPLEX E(NB),ST(NT),ZLINE(NLINES),YC(NLINES)

C     THIS GIVES INITIAL VALUES FOR PSSE

  340 DO 60 I = 1,NB
      V(I) = 1.0
   60 D(I) = 0.0
   70 CONTINUE

      DO 85 I = 1, NLINES
      NY(I)=AIMAG(YC(I))
   85 CONTINUE

      K=0
   90 CONTINUE
```

```
C     THIS CALCULATE THE WEIGHTING MATRIX WMM
      CALL WMGP(AM,WMM,NMM,PNM,QNM,PLM,QLM,VM,NB,NNBB,NT,
     1ERR1,ERR2,FERR1,FERR2,WMN)

C     THIS CALCULATES PNC AND QNC FROM CURRENT VALUES OF V AND D
      CALL PQCAL(PNC,QNC,V,D,B,G,NB)

C     THIS CALCULATES PLC AND QLC FROM CURRENT VALUES OF V AND D
      CALL PQLCAL(PLC,QLC,V,D,B,G,NB,MTOBO,E,ST,ZLINE,YC,NLINES,NT)

C     THIS CALCULATES THE DIFFERENCES BETWEEN MEASUREMENTS AND
C     CALCULATED VALUES
      CALL EDELTA(PNM,QNM,PLM,QLM,VM,PNC,QNC,PLC,QLC,V,CONM,
     1NMM,NB,NT)


C     WHEN FIXED JACOBIAN METHOD IS USED, THE FOLLOWING STATEMENT
C     SHOULD BE USED.
      IF (K.GT.0.0) GO TO 3333

C     CALCULATING THE JACOBIAN MATRIX FOR STATE ESTIMATION (JKE)
      CALL EJACOB(JK11,JK12,JK21,JK22,JK31,JK32,JK41,JK42,JK51,JK52,
     1JKEM,PNC,QNC,V,D,NB,NT,B,G,NNBB,NMM,MTOBO,NY,NLINES,PNM,QNM,
     1PLM,QLM,VM)

 3333 CONTINUE

      III=1
      DO 200 I=1,NMM
      IF (am(I).EQ.0) GO TO 200
      DO 212 J=2,NNBB
      JKE(III,J-1)=JKEM(I,J)
  212 CONTINUE
      CON(III)=CONM(I)
      WM(III,III)=WMM(I,I)
      III=III+1
  200 CONTINUE

C     WHEN FIXED JACOBIAN METHOD IS USED, THE FOLLOWING STATEMENT
C     SHOULD BE USED.
      IF (K.GT.0) GO TO 3334

C     FINDING THE TRANSPOSE MATRIX OF JKE--JKET
      CALL TRAN(JKET,NNBB1,NM,JKE)

 3334 CONTINUE
```

```
C     CALCULATING THE RESULT OF JKET*WM
      CALL XRR(JKET,NNBB1,NM,WM,NM,AESTIM)

C     CALCULATING THE RESULT OF AESTIM*JKE
      CALL XRR(AESTIM,NNBB1,NM,JKE,nnbb1,ESTIM)

C     CALCULATING THE RESULT OF AESTIM*CON
      CALL XRR(AESTIM,NNBB1,NM,CON,1,CON1)

C     CALCULATING THE INVERSE MATRIX OF ESTIM
      CALL INVERM(IM,ESTIM,NNBB1)

C     CALCULATING THE RESULT OF IM*CON1
      CALL XRR(IM,NNBB1,NNBB1,CON1,1,CON2)

      DO 100 I=2,NB
      D(I)=D(I)+CON2(I-1)
  100 CONTINUE
      DO 105 I=1,NB
      V(I)=V(I)+V(I)*CON2(I+NB1)

  105 CONTINUE

C     THIS TEST THE PROGRAM IS CONVERGED OR DIVERGED OR
C     NOT BOTH
      CALL TEST(CON2,ITEST,NB,CRIT)
      IF(ITEST.EQ.1.OR.ITEST.EQ.2)GO TO 110
      K=K+1
      IF (K.LT.KMAX)  GO TO 90
      WRITE(6,*)'K=KMAX'

  110 CONTINUE
      WRITE(6,*)'ITEST=',ITEST

      RETURN
      END
```

```
C*****************************************************************
      SUBROUTINE PSSEP(V,D,NLINES,NY,YC,AM,WMM,NMM,PNM,QNM,PLM,QLM,
     1VM,NB,NNBB,NT,ERR1,ERR2,FERR1,FERR2,WMN,PNC,QNC,PLC,QLC,B,G,
     1MTOBO,E,ST,ZLINE,JK11,JK12,JK21,JK22,JK31,JK32,JK41,
     1JK42,JK51,JK52,JKEM,JKE,CON,CONM,WM,JKET,NNBB1,
     1AESTIM,ESTIM,CON1,IM,CON2,ITEST,CRIT,NM,KMAX,NB1)
c*****************************************************************
      REAL JK11(NB,NB),JK12(NB,NB),JK21(NB,NB),JK22(NB,NB)
      REAL JK31(NT,NB),JK32(NT,NB),JK41(NT,NB),JK42(NT,NB),
     1JK51(NB,NB),JK52(NB,NB),JKET(NNBB1,NM),JKE(NM,NNBB1)
      REAL JKEM(NMM,NNBB)
      REAL V(NB),D(NB),NY(NLINES),B(NB,NB),G(NB,NB)
      REAL CONM(NMM),WMM(NMM,NMM),WM(NM,NM)
      REAL AESTIM(NNBB1,NM),ESTIM(NNBB1,NNBB1),CON(NM),CON1(NNBB1),
     1CON2(NNBB1),IM(NNBB1,NNBB1)
      INTEGER MTOBO(NT,4),AM(NMM),KMAX

      REAL PNM(NB),QNM(NB),PLM(NT),QLM(NT),VM(NB)
      REAL PNC(NB),QNC(NB),PLC(NT),QLC(NT)

      COMPLEX E(NB),ST(NT),ZLINE(NLINES),YC(NLINES)

C     THIS GIVES INITIAL VALUES FOR PSSE

  340 DO 60 I = 1,NB
      V(I) = 1.0
   60 D(I) = 0.0
   70 CONTINUE

      DO 85 I = 1, NLINES
      NY(I)=AIMAG(YC(I))
   85 CONTINUE

      K=0
   90 CONTINUE

C     THIS CALCULATE THE WEIGHTING MATRIX WMM
c      CALL WMGP(AM,WMM,NMM,PNM,QNM,PLM,QLM,VM,NB,NNBB,NT,
c     1ERR1,ERR2,FERR1,FERR2,WMN)

C     THIS CALCULATES PNC AND QNC FROM CURRENT VALUES OF V AND D
      CALL PQCAL(PNC,QNC,V,D,B,G,NB)

C     THIS CALCULATES PLC AND QLC FROM CURRENT VALUES OF V AND D
      CALL PQLCAL(PLC,QLC,V,D,B,G,NB,MTOBO,E,ST,ZLINE,YC,NLINES,NT)

C     THIS CALCULATES THE DIFFERENCES BETWEEN MEASUREMENTS AND
C     CALCULATED VALUES
```

```
      CALL EDELTA(PNM,QNM,PLM,QLM,VM,PNC,QNC,PLC,QLC,V,CONM,
     1NMM,NB,NT)


C     WHEN FIXED JACOBIAN METHOD IS USED, THE FOLLOWING STATEMENT
C     SHOULD BE USED.
c      IF (K.GT.0.0) GO TO 3333

C     CALCULATING THE JACOBIAN MATRIX FOR STATE ESTIMATION (JKE)
c      CALL EJACOB(JK11,JK12,JK21,JK22,JK31,JK32,JK41,JK42,JK51,JK52,
c     1JKEM,PNC,QNC,V,D,NB,NT,B,G,NNBB,NMM,MTOBO,NY,NLINES,PNM,QNM,
c     1PLM,QLM,VM)

 3333 CONTINUE

      III=1
      DO 200 I=1,NMM
      IF (am(I).EQ.0) GO TO 200
c      DO 212 J=2,NNBB
c      JKE(III,J-1)=JKEM(I,J)
c  212 CONTINUE
      CON(III)=CONM(I)
c      WM(III,III)=WMM(I,I)
      III=III+1
  200 CONTINUE

C     WHEN FIXED JACOBIAN METHOD IS USED, THE FOLLOWING STATEMENT
C     SHOULD BE USED.
c      IF (K.GT.0) GO TO 3334

C     FINDING THE TRANSPOSE MATRIX OF JKE--JKET
c      CALL TRAN(JKET,NNBB1,NM,JKE)

c 3334 CONTINUE

C     CALCULATING THE RESULT OF JKET*WM
c      CALL XRR(JKET,NNBB1,NM,WM,NM,AESTIM)

C     CALCULATING THE RESULT OF AESTIM*JKE
c      CALL XRR(AESTIM,NNBB1,NM,JKE,nnbb1,ESTIM)

C     CALCULATING THE RESULT OF AESTIM*CON
      CALL XRR(AESTIM,NNBB1,NM,CON,1,CON1)

C     CALCULATING THE INVERSE MATRIX OF ESTIM
c      CALL INVERM(IM,ESTIM,NNBB1)

C     CALCULATING THE RESULT OF IM*CON1
```

```
      CALL XRR(IM,NNBB1,NNBB1,CON1,1,CON2)

      DO 100 I=2,NB
      D(I)=D(I)+CON2(I-1)
  100 CONTINUE
      DO 105 I=1,NB
      V(I)=V(I)+V(I)*CON2(I+NB1)

  105 CONTINUE

C     THIS TEST THE PROGRAM IS CONVERGED OR DIVERGED OR
C     NOT BOTH
      CALL TEST(CON2,ITEST,NB,CRIT)
      IF(ITEST.EQ.1.OR.ITEST.EQ.2)GO TO 110
      K=K+1
      IF (K.LT.KMAX)  GO TO 90
      WRITE(6,*)'K=KMAX'

  110 CONTINUE
      WRITE(6,*)'ITEST=',ITEST

      RETURN
      END




      SUBROUTINE SORTGREAT(AIN,AOUT,NN)
C     THE INPUT TO THIS SUB IS ARRAY AIN(NN), THE OUTPUT IS AOUT(NN,2).
C     THE OUTPUT ARRAY WILL CONTAIN THE ELEMENTS OF AIN, SORTED OUT,
C     THE SMALLER IN ABSOLUTE VALUE IS ORDERED FIRST.  THE ORDER OF
C     THE ELEMENT AS IT WAS IN AIN IS STORED IN THE FIRST COLUMN OF AOUT.
C     ARRAY AIN WILL CONTAIN IN PLACE OF A PROCESSED ELEMENT, THE ABSOLUTE
C     OF THE LARGEST ENTRY + 1.0.


INTEGER ORDER
REAL AIN(NN), AOUT(NN,2)
ALARGE =0.0
c     THIS DO LOOP WILL SET "ALARGE" TO A VALUE GREATER THAN THE
C     GREATEST ABSOLUTE VALUE OF "AIN" ELEMENTS + 1.
C DO 2 I=1,NN
C A=ABS(AIN(I))
C IF(A.LT.ALARGE)ALARGE=A-1.0
C2 CONTINUE
DO 10 KK=1,NN
SMALLEST = ALARGE
```

```
DO 5 I=1,NN
        VALUE=ABS(AIN(I))
IF(VALUE.GT.SMALLEST)THEN
SMALLEST = VALUE
ORDER = I
ELSE
CONTINUE
ENDIF
5 CONTINUE
AOUT(KK,1)=ORDER
AOUT(KK,2)=SMALLEST
AIN(ORDER) = ALARGE
10 CONTINUE
RETURN
END
```

```
      SUBROUTINE SORT(AIN,AOUT,NN)
C     THE INPUT TO THIS SUB IS ARRAY AIN(NN), THE OUTPUT IS AOUT(NN,2).
C     THE OUTPUT ARRAY WILL CONTAIN THE ELEMENTS OF AIN, SORTED OUT,
C     THE SMALLER IN ABSOLUTE VALUE IS ORDERED FIRST.  THE ORDER OF
C     THE ELEMENT AS IT WAS IN AIN IS STORED IN THE FIRST COLUMN OF AOUT.
C     ARRAY AIN WILL CONTAIN IN PLACE OF A PROCESSED ELEMENT, THE ABSOLUTE
C     OF THE LARGEST ENTRY + 1.0.

INTEGER ORDER
REAL AIN(NN), AOUT(NN,2)
ALARGE =1.0
c     THIS DO LOOP WILL SET "ALARGE" TO A VALUE GREATER THAN THE
C     GREATEST ABSOLUTE VALUE OF "AIN" ELEMENTS + 1.
DO 2 I=1,NN
A=ABS(AIN(I))
IF(A.GT.ALARGE)ALARGE=A+1.0
2 CONTINUE
DO 10 KK=1,NN
SMALLEST = ALARGE
DO 5 I=1,NN
        VALUE=ABS(AIN(I))
IF(VALUE.LT.SMALLEST)THEN
SMALLEST = VALUE
ORDER = I
ELSE
CONTINUE
ENDIF
5 CONTINUE
```

```
AOUT(KK,1)=ORDER
AOUT(KK,2)=SMALLEST
AIN(ORDER) = ALARGE
10 CONTINUE
RETURN
END



      SUBROUTINE  STAN(A,N,AVE,SIG)
DIMENSION A(N)
SUM = 0.0
DO 10 I = 1,N
SUM = SUM +ABS(A(I))
10 CONTINUE
AVE = SUM/N
SIG = 0.0
DO 20 I=1,N
SIG = SIG + (abs(A(I)) - AVE)*(abs(A(I)) - AVE)
20 CONTINUE
SIG = SQRT(SIG/(N-1))
RETURN
END




C*********************************************************************
      SUBROUTINE SYSDATA(NB,NLINES,G,B,NT,K1,MTOBO,PSP,QSP,VSP,SP,
     1YC,ZLINE,K2)
C*********************************************************************
C
C     THIS SUBROUTINE READS IN THE SYSTEM PARAMETERS AND OTHER DATA
C
C     PARAMETER PASSED:
C         K1 = NUMBER OF LOAD BUS + 1
C         K2 = NUMBER OF LOAD BUS + 2, NEEDED TO READ SPECIFIED VOLTAGES
C         NB = NUMBER OF BUSES
C         NLINE = NUMBER OF LINES
C
C     PARAMETER RETURNED:
C         G = ADMITTANCE
C         B = SUSCEPTANCE
C         NT = 2 $\times$ NLINES
C         MTOBO = ARRAY OF: #, FROM_BUS, TO_BUS, LINE #
```

```
C         PSP = ARRAY OF SPECIFIED NODAL REAL POWER INJECTIONS
C         QSP = ARRAY OF SPECIFIED NODAL REACTIVE POWER INJECTIONS
C         SP = ARRAY OF COMPLEX NODAL POWER INJECTIONS
C         YC = ARRAY OF LINE CHANGING ADMITTANCES
C         ZLINE = ARRAY OF LINE IMPEDANCES
C
C     CALLED BY:
C         MAIN PROGRAM
C
C     CALLS:
C         NONE
C
C*************************************************************************

      REAL G(NB,NB),B(NB,NB),PSP(NB),QSP(NB),VSP(NB)

      COMPLEX SP,YC(NLINES),ZLINE(NLINES)

      INTEGER MTOBO(NT,4),K2

C     READING LINE ADMITTANCE

      DO 5 I=1,NB
      DO 5 J=1,NB
      B(I,J)=0.0
    5 G(I,J)=0.0
      LN=NB+NLINES
      DO 10 K=1,LN
      READ(15,*) I,J,G(I,J),B(I,J)
      G(J,I)=G(I,J)
      B(J,I)=B(I,J)
   10 CONTINUE

C     READING BUS DATA

      DO 15 I=1,NB
      READ(15,*)J,SP
      PSP(J)=-REAL(SP)
      QSP(J)=-AIMAG(SP)
   15 CONTINUE

      DO 4 I = 1,NT
      READ(15,*)(MTOBO(I,J),J=1,4)
    4 CONTINUE
      READ(15,*)(YC(I),ZLINE(I),I=1,NLINES)
      DO 20 I=K2,NB
      READ(15,*)VSP(I)
   20 CONTINUE
```

```
      RETURN
      END



C****************************************
      SUBROUTINE NRAN(NUDR,NDR,SIG,IR)
C****************************************
C
C
C     THIS SUBROUTINE IS USED TO CREAT NORMALLY DISTRIBUTED
C     NUMBERS USING UNIFORMLY DISTRIBUTED NUMBERS BASED ON
C     THE CENTRAL LIMITATION THEORY
C
C
C     PARAMETERS PASSED:
C         NUDR = NO. OF NORMALLY DISTRIBUTED RANDOM NUMBER
C                INTEND TO GENERATE
C         SIG = STANDARD DEVIATION OF THE NORMALLY DISTRIBUTED
C               RAMDOM NUMBERS CREATED. THIS VALUE SHOULD BE
C               SPECIFIED AT THE BEGINNING OF THE SUBROUTINE
C         IR = A SEED USED TO PRODUCE UNIFORMLY DISTRIBUTED
C              RAMDOM NUMBERS
C
C
C     PARAMETER RETURNED:
C         NDR = NORMALLY DISTRIBUTED RANDOM NUMBER
C
C     CALLED BY:
C         SUBROUTINE MG
C
C     CALLS:
C         SUBROUTINE RANDOM
C
C*******************************************************************************


      REAL NDR,RAN,SIG

      INTEGER NUDR,IR

      SUM=0.0

      DO 10 I=1,NUDR
      CALL RANDOM(IR,RAN)
      SUM=SUM+RAN
```

```
   10 CONTINUE

      NDR=SUM*SIG*SQRT(3./NUDR)

      RETURN
      END

C*********************************************************************
      SUBROUTINE EJACOB(JK11,JK12,JK21,JK22,JK31,JK32,JK41,JK42,JK51,
     1JK52,JKEM,PNC,QNC,V,D,NB,NT,B,G,NNBB,NMM,MTOBO,NY,NLINES,PNM,QNM,
     2PLM,QLM,VM)
C*********************************************************************
C
C     THIS SUBROUTINE CALCULATES JACOBIAN MATRIX.
C     PNC AND QNC ARE NEEDED TO CALCULATE THE DIAGONAL
C     ELEMENTS OF  SUBEJACOBIANS. THE OUTPUT IS JKEM.
C
C     PARAMETERS PASSED:
C         PNC = ARRAY OF CALCULATED NODAL REAL POWER INJECTIONS
C         QNC = ARRAY OF CALCULATED NODAL REACTIVE POWER INJECTIONS
C         V = ARRAY OF MAGNITUDE OF NODAL VOLTAGES
C         D = ARRAY OF ANGLE OF NODAL VOLTAGE ANGLES
C         NB = NUMBER OF BUSES
C         NT = 2 $\times$ NLINES
C         B = SUSCEPTANCES
C         G = ADMITTANCES
C         NNBB = 2 \times$ NB
C         NMM = NUMBER OF SYSTEM VARIABLES (= NNBB + 2 $\times $ NT + NB)
C         MTOBO = ARRAY OF: #, FROM_BUS, TO_BUS, LINE #
C         NY =
C         NLINES = NUMBER OF LINES
C
C     PARAMETERS RETURNED:
C         JKEM = JACOBIAN MATRIX FOR STATE ESTIMATION
C
C     CALLED BY:
C         MAIN PROGRAM
C
C     CALLS:
C          NONE
C
C*****************************************************

      REAL JK11(NB,NB),JK12(NB,NB),JK21(NB,NB),JK22(NB,NB)
      REAL JK31(NT,NB),JK32(NT,NB),JK41(NT,NB),JK42(NT,NB),
     1JK51(NB,NB),JK52(NB,NB),NY(NLINES)
      REAL JKEM(NMM,NNBB),PNC(NB),QNC(NB),V(NB),D(NB),B(NB,NB),G(NB,NB)
```

```
      INTEGER MTOBO(NT,4)

C     GENERATING JK11,JK12,JK21,JK22,JK31,JK32,JK41,JK42,
C     JK51 AND JK52


      DO 10 I=1,NB
      DO 20 J=1,NB
      IF (I.EQ.J) GOTO 20
      IF (G(I,J).EQ.0.0.AND.B(I,J).EQ.0.0) GO TO 20
      DIJ=D(I)-D(J)
      SINDIJ=SIN(DIJ)
      COSDIJ=COS(DIJ)
      JK11(I,J)=V(I)*V(J)*(G(I,J)*SINDIJ-B(I,J)*COSDIJ)
      JK12(I,J)=V(I)*V(J)*(G(I,J)*COSDIJ+B(I,J)*SINDIJ)
      JK21(I,J)=-JK12(I,J)
      JK22(I,J)=JK11(I,J)
   20 CONTINUE
   10 CONTINUE

      DO 30 I=1,NB
      JK11(I,I)=(-V(I)*V(I)*B(I,I)-QNC(I))
      JK12(I,I)=(V(I)*V(I)*G(I,I)+PNC(I))
      JK21(I,I)=(-V(I)*V(I)*G(I,I)+PNC(I))
      JK22(I,I)=(-V(I)*V(I)*B(I,I)+QNC(I))
   30 CONTINUE

      DO 40 II=1,NT
      KK=MTOBO(II,1)
      I=MTOBO(II,2)
      J=MTOBO(II,3)
      L=MTOBO(II,4)
      DIJ=D(I)-D(J)
      SINDIJ=SIN(DIJ)
      COSDIJ=COS(DIJ)

      JK31(KK,I)=V(I)*V(J)*(-G(I,J)*SINDIJ+B(I,J)*COSDIJ)
      JK31(KK,J)=-JK31(KK,I)

      JK32(KK,I)=V(I)*V(J)*(G(I,J)*COSDIJ+B(I,J)*SINDIJ)
      JK32(KK,J)=JK32(KK,I)
      JK32(KK,I)=JK32(KK,I)-2*V(I)*V(I)*G(I,J)

      JK41(KK,I)=V(I)*V(J)*(B(I,J)*SINDIJ+G(I,J)*COSDIJ)
      JK41(KK,J)=-JK41(KK,I)

      JK42(KK,I)=-V(I)*V(J)*(B(I,J)*COSDIJ-G(I,J)*SINDIJ)
      JK42(KK,J)=JK42(KK,I)
```

```
      JK42(KK,I)=JK42(KK,I)+2*V(I)*V(I)*B(I,J)
      JK42(KK,I)=JK42(KK,I)-2*V(I)*V(I)*NY(L)

   40 CONTINUE

      DO 50 I=1,NB
      JK52(I,I)=V(I)
   50 CONTINUE

C     CREATING JKEM MATRIX USING JK11,JK12,JK21,JK22,
C     JK31,JK32,JK41,JK42,JK51 AND JK52

      DO 60 I=1,NB
      DO 60 J=1,NB
      JKEM(I,J)=JK11(I,J)
   60 CONTINUE

      DO 70 I=1,NB
      DO 70 J=1,NB
      JKEM(I,J+NB)=JK12(I,J)
   70 CONTINUE

      DO 80 I=1,NB
      DO 80 J=1,NB
      JKEM(I+NB,J)=JK21(I,J)
   80 CONTINUE

      DO 90 I=1,NB
      DO 90 J=1,NB
      JKEM(I+NB,J+NB)=JK22(I,J)
   90 CONTINUE

      DO 100 I=1,NT
      DO 100 J=1,NB
      JKEM(I+NB+NB,J)=JK31(I,J)
  100 CONTINUE

      DO 110 I=1,NT
      DO 110 J=1,NB
      JKEM(I+NB+NB,J+NB)=JK32(I,J)
  110 CONTINUE

      DO 120 I=1,NT
      DO 120 J=1,NB
      JKEM(I+NB+NB+NT,J)=JK41(I,J)
  120 CONTINUE

      DO 130 I=1,NT
```

```
      DO 130 J=1,NB
      JKEM(I+NB+NB+NT,J+NB)=JK42(I,J)
  130 CONTINUE

      DO 140 I=1,NB
      DO 140 J=1,NB
      JKEM(I+NB+NB+NT+NT,J)=JK51(I,J)
  140 CONTINUE

      DO 150 I=1,NB
      DO 150 J=1,NB
      JKEM(I+NB+NB+NT+NT,J+NB)=JK52(I,J)
  150 CONTINUE

  999 RETURN
      END


C*****************************************************************
      SUBROUTINE PQLCAL(PLC,QLC,V,D,B,G,NB,MTOBO,E,ST,ZLINE,
     1YC,NLINES,NT)
C*****************************************************************
C     THIS SUBROUTINE CALCULATES THE COMPLEX LINE FLOWS AT EACH END
C     OF EACH LINE
C
C
C     PARAMETER PASSED:
C         V = ARRAY OF MAGNITUDE OF NODAL VOLTAGES
C         D = ARRAY OF ANGLE OF NODAL VOLTAGE ANGLES
C         B = SUSCEPTANCES
C         G = ADMITTANCES
C         NB = NUMBER OF BUSES
C         MTOBO = ARRAY OF: #, FROM_BUS, TO_BUS, LINE #
C         E = ARRAY OF COMPLEX NODAL VOLTAGES
C         ST = ARRAY OF COMPLEX LINE FLOWS (FOR EACH END OF EACH LINE)
C         ZLINE = ARRAY OF LINE IMPEDANCES
C         YC = ARRAY OF LINE CHARGING ADMITTANCES
C         NLINE = NUMBER OF LINES
C         NT = 2 $\times$ NLINES
C
C     PARAMETER RETURNED:
C         PLC = ARRAY OF CALCULATED REAL POWER FLOWS
C         QLC = ARRAY OF CALCULATED REACTIVE POWER FLOWS
C
C     CALLED BY:
C         MAIN PROGRAM
C
C     CALLS:
```

```
C          SUBROUTINE CLF
C
C*****************************************************


      REAL PLC(NT),QLC(NT),V(NB),D(NB),G(NB,NB),B(NB,NB)

      COMPLEX E(NB),ST(NT),ZLINE(NLINES),YC(NLINES)


      DO 10 I = 1,NB
      AA = V(I)*COS(D(I))
      BB = V(I)*SIN(D(I))
      E(I) = CMPLX(AA,BB)
   10 CONTINUE

      CALL CLF(MTOBO,NT,E,NB,ST,ZLINE,YC,NLINES)

C     LINE FLOW CALCULATION USING CURRENT V AND D VALUES

      DO 20 I = 1,NT
      PLC(I) = REAL(ST(I))
      QLC(I) = AIMAG(ST(I))
   20 CONTINUE

      RETURN
      END

C     ************************************
      SUBROUTINE PQCAL(P,Q,V,D,B,G,NB)
C     ************************************

      REAL P(NB),Q(NB),V(NB),D(NB),G(NB,NB),B(NB,NB)

      DO 30 I=1,NB
      P(I)=0.0
      Q(I)=0.0
      DO 25 J=1,NB
      IF(G(I,J).EQ.0.0.AND.B(I,J).EQ.0.0)GO TO 25
      DIJ=D(I)-D(J)
      SINDIJ = SIN(DIJ)
      COSDIJ = COS(DIJ)
      P(I) = P(I) + V(J)*(G(I,J)*COSDIJ + B(I,J)*SINDIJ)
      Q(I) = Q(I) + V(J)*(-B(I,J)*COSDIJ + G(I,J)*SINDIJ)
   25 CONTINUE
      P(I) =P(I)*V(I)
      Q(I) = Q(I)*V(I)
   30 CONTINUE
```

```
      RETURN
      END



C     ****************************
      SUBROUTINE RANDOM(R,RAN)
C     ****************************

C     THE PROGRam IS A RANDOM PROGRam TO GET THE RANDOM
C     NUMBER. R IS A 'SEED' AND RAN IS A RANDOM NUMBER
      INTEGER R,A
      R=R*2045+1497794
      A=R/1048576
      R=R-A*1048576
      RAN=1.D0*FLOAT(R)/1048576.D0
      RAN=(RAN-0.5)*2
      RETURN
      END

C     ************************************************************
      SUBROUTINE FRANK(NB,NL,NNBB,K1,KMAX,CRIT,P,PSP,Q,QSP,
     1V,D,CONN,G,B,JK11,JK12,JK21,JK22,FM,DFM,FJM,MFJM,NR,NEW)
C     ************************************************************

C     IF THE FACTORIZED FM IS AVAILABLE AND THE FIXED JACOBIAN
C     METHOD IS TO BE USED USING THAT FM, THEN NEW=0
C     THIS SUBROUTINE CALCULATES THE NODAL VOLTAGES V AND D
C     USING EITHER NR METHOD (FJM = 0) OR THE FIXED JACOBIAN
C     METHOD (FJM = 1).  BEFORE CALLING THE SUBROUTINE,
C     INITIAL VALUES FOR V AND D MUST BE SPECIFIED.
C     NB      # OF BUSES
C     K1      # OF LOAD NODES + 1
C     K2      # OF LOAD NODES + 2
C     NL      # OF LOAD NODES
C     K       # OF ITERATIONS FOR CONVERGENCE
C     KMAX    MAXIMUM  # OF ITERATIONS ALLOWED
C     CRIT    CONVERGENCE CRITERION
C     ITEST   CONVERGENCE TESTING INDICATOR

      INTEGER DFM,FJM
      REAL JK11(NB,NB),JK12(NB,NB),JK21(NB,NB),JK22(NB,NB)
      REAL P(NB),PSP(NB),Q(NB),QSP(NB),V(NB),D(NB),CONN(NNBB)
      REAL G(NB,NB),B(NB,NB),FM(DFM,DFM)
      K=0
      IF(NEW.NE.1)GO TO 37
   10 CALL PQCAL(P,Q,V,D,B,G,NB)
```

```
      CALL JACOBB(JK11,JK12,JK21,JK22,FM,P,Q,V,D,NB,DFM,K1,B,G)
      CALL FACTOO(DFM,FM)
   35 CONTINUE
      IF(FJM.NE.1 )GO TO 40
   37 CALL PQCAL(P,Q,V,D,B,G,NB)
   40 CALL DELTAA(PSP,QSP,P,Q,CONN,NB,NNBB)
      CALL TESTPOO(NB,NNBB,K1,ITEST,CRIT,CONN,PSP,QSP,K,FJM,MFJM,NR)
      IF(ITEST.EQ.1.OR.ITEST.EQ.2)GO TO 60
      IF(K.LT.KMAX)GO TO 45
      WRITE(6,150)KMAX,FJM
  150 FORMAT(15X,22HFAILED TO CONVERGE IN ,I2,
     126HITERATION BY METHOD FJM = ,I2//)
GO TO 60
   45 CALL UTIONN(NB,NNBB,DFM,CONN,FM)
      DO 50 I=2,NB
      D(I) = D(I) + CONN(I-1)
   50 CONTINUE
      DO 55 I = 2,K1
   55 V(I) = V(I)*(1.0 + CONN(I + NB - 2))
      K = K+1
   32 IF(FJM.EQ.1)GO TO 35
      GO TO 10
   60 RETURN
      END
```

SELECTED BIBLIOGRAPHY

[1] F. C. Schweppe and J. Wildes, "Power System Static-State Estimation, Part I: Exact Model," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, pp. 120-125, January 1970.

[2] F. C. Schweppe and D. B. Rom, "Power System Static-State Estimation, Part II: Approximate Model," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, pp. 125-130, January 1970.

[3] F. C. Schweppe, "Power System Static-State Estimation, Part III: Implementation," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, pp. 130-135, January 1970.

[4] R. E. Larson, W. F. Tinney and J. Peschon, "State Estimation in Power Systems, Part I: Theory and Feasibility," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, No. 3, pp. 345-352, March 1970.

[5] J. F. Dopazo, O. A. Klitin, G. W. Stagg and L. S. Van Slyck, "State Calculation of Power Systems from Line Flow Measurements," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, No. 7, pp. 1698-1709, September/October 1970.

[6] J. F. Dopazo, O. A. Klitin and L. S. VanSlyck, "State Calculation of Power Systems from Line Flow Measurements, Part II," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-91, pp. 145-151, January/February 1972.

[7] E. Handschin, F. C. Schweppe, J. Kohlas and A. Fiechter, "Bad Data Analysis for Power System State Estimation," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-94, pp.329-337, March/April 1975.

[8] H. M. Merril and F. C. Schweppe, "Bad Data Suppression in Power System Static Estimation," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-90, pp.2718-2725, November/December 1971.

[9] M. R. Irving, R. C. Owen and M. J. H. Sterling, "Power System State Estimation Using Linear Programming," *Proceeding of IEE*, Vol. 125, pp. 879-885, 1978.

[10] W. W. Kotiuga and M. Vidyasagar, "Bad Data Rejection Properties of Weighted Absolute Value Techniques Applied to Static State Estimation," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-101, No. 4, pp. 844-851, April 1982.

[11] J. D. Lenk, *Hand Book of Electronic Meters Theory and Application*, Prentice Hall, Inc., 1981.

[12] E. Frank, *Electrical Measurement Analysis*, McGraw-Hill Book Company, 1959.

[13] B. A. Barry, *Errors in Practical Measurement in Science, Engineering, and Technology*, John Wiley & Sons. Inc., 1978.

[14] F. A. R. Shirazi, *The Path Method in Power System State Estimation*, MS thesis, NJIT, 1988.

[15] S. lin, *Independant Equations Method for Power System State-estimation*, MS thesis, NJIT, 1988.

[16] W. Hubbi "Constant sensitivity matrix for bad data identification and reconstraction in power systems" *Presented at 31th Midwest Symposiumon on Circuits and Systems*, 1989.

[17] Yanshi. Wang *The Studies Of Effects Of Weighting Matries State-estimation*, MS thesis, NJIT, 1990.

[18] W. Hubbi and J. Wu, "Convergence Criterion for Power System State Estimation," *30th Midwest Symposiumon on Circuits and Systems*, pp. 827-830.

[19] W. F. Tinney and C. E. Hart, "Power Flow Solution By Newton's Method," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-86, No. 11, pp. 1449-1460, November, 1967.

[20] K. L. Lo, et al. *Development of a Static State Estimator, Part I: Estimation and Bad Data Suppression.* IEEE Trans. Power Apparatus and Systems, Vol. PAS-102, No. 8, pp. 2486-2491, August 1983.

[21] K. L. Lo, et al. *Development of a Static State Estimator, Part II: Bad data Replacement and Generation of Pseudomeasurements.* IEEE Trans. Power Apparatus and Systems, Vol. PAS-102, No. 8, pp. 2492-2500, August 1983.

[22] A. Monticelli and A. Garcia. *Reliable Bad Data Processing for Real-Time State Estimation.* IEEE Trans. Power Apparatus and Systems, Vol. PAS-102, No. 5, pp. 1126-1139, May 1983.

[23] O. J. M. Smith, "Power System State Estimation," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, No. 3, pp.363-379, March 1970.

[24] L. Mili, Th. Van cutsem and M. Ribbens-Pavella, "Bad Data Identification Methods In Power System State Estimation-A Comparative Study" *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-104, No. 11, pp. 3037-3045, November 1985.

[25] L. Mili, Th. Van cutsem and M. Ribbens-Pavella, "Hypothesis Testing Identifation" *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-103, No. 11, pp. 3239-3247, November 1984.

[26] M. M. Adibi, R. J. Kafka, K. A. Clements and J. P. Stovall, "Intergration of Remote Measurement Calibration With State Estimation - A Feasibility Study," *IEEE/PES 1990 Summer Meeting*, Minneapolis, Minnesota, July 15-19, 1990.

[27] G. W. Stagg, J. F. Dapozo, O. A. Klitin and L. S. Vanslyck, "Techniques for the Real-Time Monitoring of Power System Operations," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-89, No. 4, pp. 545-555, April 1970.

[28] X. Niande, W. shiying, Y. Erkeng, "An application of estimation identification approach of multiple bad data in power system state estimation," *IEEE*

Transactions on Power Apparatus and Systems, Vol. PAS-103, pp. 225-233, Feb. 1984.

[29] R.D. Shultz, R.A. Smith, L.L. Witthun, "Succested bad data substitution using a search emulation procedure," *IEEE Transactions on Power Systems*, pp. 219-224, Dec. 1987.

[30] L. Mili, Th.Van Cutsem, "Implementation of the hypothesis testing identification in power system state estimation," *IEEE Transactions on Power Systems*, pp. 887-893, Aug. 1988.

[31] I. W. Slutsker, "Bad data identification in power system state estimation based on measurement compensation and linear residual calculation ," *IEEE Transactions on Power Systems*, pp. 55-60, Feb. 1989.

[32] L.S. Van Slyck, J.J. Allemong, "Opreating experience with the AEP state estimator," *IEEE Transactions on Power Systems*, pp. 521-528, May. 1988.

[33] F.F. WU, W.H.E. Liu, "Observability analysis and bad data processing for state estimation with equality constraints ," *IEEE Transactions on Power Systems*, pp. 541-548, May. 1988.

[34] E. E. Fetzer and P. M. Anderson, "Observability in the State Estimation of Power Systems," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-94, No. 6, pp.1981-1985, November/December 1975.

[35] H. Takata, E. Uchino and S. Takata, "A Judging Function of the State-Estimation Accuracy and its Application to the Electric System," *IEEE Trans-*

actions on Power Apparatus and Systems, Vol. PAS-100, No. 12, pp.5048-5052, December 1981.

[36] K. L. Lo, P. S. Ong, R. D. McColl, A. M. Moffatt and J. L. Sulley, "Development of a Static State Estimator, Part I: Estimation and Bad Data Suppression," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-102, No. 8, pp.2486-2491, August 1983.

[37] K. L. Lo, P. S. Ong, R. D. McColl, A. M. Moffatt and J. L. Sulley, "Development of a Static State Estimator, Part II: Bad Data Replacement and Generation of Pseudomeasurements," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-102, No. 8, pp.2492-2500, August 1983.

[38] N. Srinivasan, K. S. Prakasa Rao and C. S. Indulkar, "Some New Algorithms for State Estimation in Power Systems," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-103, No. 5, pp.982-987, May 1984.

[39] F. Zhuang and R. Balasubramanian, "A Transformation-decoupled Estimator for Power System State Estimation," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-104, No. 7, pp.1738-1744, July 1985.

[40] F. F. Wu, W. -H. E. Liu, L. Holten, A. Gjelsvik and S. Aam, "Observability Analysis and Bad Data Processing for State Estimation Using Hachtel's Augmented Matrix Method," *Conference Papers - 1987 Power Industry Application Conference: PICA 87*, Montreal, Que., Canada, 18-22 May 1987 (New York, NY, USA: IEEE 1987), p. 59-65.

[41] N. Srinivasan, M. A. Naarasimhan, S. Sen, S. Subbaiyya and V. Kumar, "An

Algorithm for a Decoupled State Estimator Using Line Flows," *Advances in Modelling & Simulation (France)*, vol. 9, no. 4, p. 1-11 (1987).

[42] Y. H. Moon, Y. M. Park and K. J. Lee, "Observable Island Identification for State Estimation Using Incidence Matrix," *Automatica (UK)*, vol. 24, no. 1, p. 71-5 (Jan. 1988).

[43] J. Cidras and A. Ollero, "New Algorithms for Electrical Power System Analysis Using Sparse Vector Methods," *Selected papers from the 4th IFAC/IFORS Symposium*, Zurich, Switerland, 26-29 Aug. 1986 (Oxford, UK: Pergamon 1987), P. 703-8 vol. 2.

[44] S. A. Azim and P. K. Sen, "State Estimation in Power Systems: Survey of State Estimation Techniques Based on the Weighted Least Square Approach," *Proceedings of the IASTED International Symposium*, Lugano, Switzerland, 29 June-1 July 1987 (Anaheim, CA, USA: ACTA Press 1987), p. 134-8.

[45] R. A. M. van Amerongen, "Optimal Power Flow Solved With Sequential Reduced Quadratic Programming," *Archiv fuer Elektrotechnic (West Germany)*, vol. 71, no. 3, p. 213-19 (1988).

[46] E. P. M. Brown, "Problems in Using State Estimation in Power Systems Which Have Poor Quality High Voltage Measurements, Phase Imbalance and High Levels of Harmonics," *Electric Power Systems Research (Switzerland)*, vol. 15, no. 2, p. 135-46 (Oct. 1988).

[47] M. W. Marcus, G. A. Maria and J. A. Findlay, "The Ontario Hydro State Estimator," *Conference Papers -1987 Power Industry Computer Application*

Conference: PICA 87, Montreal, Que.; Canada, 18-22 May 1987 (New York, NY, USA: IEEE 1987), p. 66-70

[48] S. -K. Chang and V. Brandwajn, "Adjusted Solutions in Fast Decoupled Load Flow," *Conference Papers -1987 Power Industry Computer Application Conference: PICA 87*, Montreal, Que., Canada, 18-22 May 1987 (New York, NY, USA: IEEE 1987), p. 347-53

[49] K. Behnam-Guilani, "Fast Decoupled Load Flow: the Hybird Model," *Conference Papers -1987 Power Industry Computer Application Conference: PICA 87*, Montreal, Que., Canada, 18-22 May 1987 (New York, NY, USA: IEEE 1987), p. 354-9

[50] D. Rajicic and A. Bose, "A Modification to the Fast Decoupled Power Flow for Networks With High r/x Ratios," *Conference Papers -1987 Power Industry Computer Application Conference: PICA 87*, Montreal, Que., Canada, 18-22 May 1987 (New York, NY, USA: IEEE 1987), p.360-3 (6 refs.)

[51] J. L. De La Fuente and J. Lumbreras, "A New Implementation of an Optimal Power Flow System Based on a General Purpose Nonlinear Programming Program," *Conference Papers -1987 Power Industry Computer Application Conference: PICA 87*, Montreal, Que., Canada, 18-22 May 1987 (New York, NY, USA: IEEE 1987), p. 422-8

[52] G. C. Contaxis and G. N. Korres, "Restoration of Observability in Power System State Estimation," *Proceedings of MELECON '87: Mediterranean Electrotechnical Conference and 34th Congress on Electronics Joint Conference*, Rome, Italy, 24-26 March 1987 (New York, NY, USA: IEEE 1987), p. 571-5

[53] R. Ramanathan and R. Natarajan, "Decoupled Load Referenced Power Flows," *Electric Power Systems Research (Switzerland)*, vol. 13, no. 3, p. 197-202 (Dec. 1987).

[54] R. D. Shultz, R. A. Smith and L. L. Witthuhn, "Suspected Bad Data Substitution Using a Search Emulation Procedure," *Electric Power Systems Research (Switzerland)*, vol. 13, no. 3, p. 219-24 (Dec. 1987).

[55] Z. liu, G. T. Heydt and M. C. Merow, "Decoupled Power System State Estimation Based on the Pseudoinverse Algorithm," *Fifth International Conference on Systems Engineering*, Dayton, OH, USA, 9-11 Sept. 1987 (New York, NY, USA: IEEE 1987), p. 289-94.

[56] W. C. Chu, W. J. Lee and M. S. Chen, "Load Model for Load Flow Studies," *Conference Record of the 1987 IEEE Industry Application Society Annual Meeting*, Atlanta, GA, USA, 18-23 Oct. 1987 (New York, NY, USA: IEEE 1987), p. 1319-24 vol. 2.

[57] K. L. Lo and M. M. Salem, "A Reliable and Efficient Linear Programming Algorithm for Two-level Power System State Estimation," *Electric Power Systems Research (Switzerland)*, vol. 14, no. 1, p. 51-60 (Feb. 1988).

[58] V. S. Khachatryan and E. A. Etmekchyan, "A Method for Power System Stead State Correction," *Electric Technology USSR (UK)*, no. 1, p. 82-102 (1987).

[59] E. F. Gorzelnik. "Solid-state Metering," *Electrical World (USA)*, vol. 202, no. 3, p. 47-52 (March 1988).

[60] K. L. Lo, M. M. Salem, R. D. McColl and A. M. Moffatt, "Two-lever State Estimation for Large Power System . II. Computational Experience," *IEE Proceedings. Part C. Generation, Transmission, and Distribution (UK)*, vol. 135, no. 4, p. 309-18 (July 1988).

[61] L. G. Dias and M. E. El-hawary, "Electric Power System Static Load Model Parameter Estimation Using Newton's Method," *Electric Machines and Power Systems (USA)*, vol. 14, no. 5, p. 317-28 (1988).

[62] Stagg, Glenn W, "Computer Methods IN Power System Analysis ," *New York McGraw-Hill*, (1968).