

4-2016

## **State Estimation – New Models on Power Distribution Networks Based on Weighted Least Squares Method in Line Current Measurements**

Hugo Vicente Barrera  
*Grand Valley State University*

Follow this and additional works at: <https://scholarworks.gvsu.edu/theses>



Part of the [Engineering Commons](#)

---

### **ScholarWorks Citation**

Vicente Barrera, Hugo, "State Estimation – New Models on Power Distribution Networks Based on Weighted Least Squares Method in Line Current Measurements" (2016). *Masters Theses*. 802.  
<https://scholarworks.gvsu.edu/theses/802>

This Thesis is brought to you for free and open access by the Graduate Research and Creative Practice at ScholarWorks@GVSU. It has been accepted for inclusion in Masters Theses by an authorized administrator of ScholarWorks@GVSU. For more information, please contact [scholarworks@gvsu.edu](mailto:scholarworks@gvsu.edu).

State Estimation – New models on power distribution networks based on weighted least squares method in line current measurements

Hugo Vicente Barrera

A Thesis Submitted to the Graduate Faculty of

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Masters of Science in Engineering

School of Engineering

April 2016

## Abstract

Calculation of Load Flow distribution is an important tool in Electrical Engineering that involves numerical analysis applied to Power Systems. State Estimation techniques have been developed and applied thoroughly mainly in the levels of generation and transmission. Research in the distribution level remains a challenge due to the intrinsic characteristics of the network. Introducing line current measurements in the state estimation process constitutes an additional issue due to distribution networks characteristics. In order to overcome these difficulties, it is necessary to develop mathematical models that simulate the behavior of those networks. The solution of the problem of state estimation by the least squares method, sometimes presents a bad conditioning of the gain matrix. Solving a badly conditioned problem results in a proximity to the singularity of the coefficient matrix. Also, the use of line current measurements in the state estimation process leads to numeric and observability problems in the systems including the cancelation of elements in the jacobian matrix in the plain state, which means that those measures are useless when starting from plain state. Also, the non-linearity of equations causes convergence difficulties in the iterative process.

The proposed work consisted of: (i) developing a state estimator for a determined radial network, (ii) introducing state variables of the developed method, (iii) comparing them with previously published work, (iv) determining the influence of estimating parameters instead of using measured values, and (v) verifying the validity of developed model using PowerWorld simulation software.

## Table of Contents

Abstract.....	3
List of Tables .....	6
List of Figures .....	7
Abbreviations.....	8
Chapter 1 Introduction .....	9
1.1 Introduction.....	9
1.2 Purpose of this work .....	13
1.3 Scope of this work .....	14
1.4 Assumptions .....	15
1.5 Definitions.....	15
Chapter 2 Review of Literature.....	16
2.1 Including current measurements in the generalized state estimation.....	16
2.2 State estimation based in a line current.....	20
2.3 State estimation based in branch current for distribution systems.....	22
2.4 Multiple solution detection in state estimation in presence of line current measurements .....	23
2.5 Line current measurement in state estimation .....	25
2.6 WLS state estimator using line current measurements .....	27
2.6.1 Modeling of current measurements .....	28
2.6.2 Difficulties in the use of current measurements .....	28
2.7 Solution through virtual measurements .....	29
2.8 Solution through equality restrictions .....	30
2.9 Hachtel augmented matrix .....	31
Chapter 3 Methodology .....	33

3.1 Network description.....	33
3.2 Proposed state estimation variables .....	35
3.3 Iterative process description .....	37
3.4 Description of Matlab code.....	39
Chapter 4 Analytical and Simulation Results .....	40
4.1 Matlab analysis .....	40
4.2 PowerWorld simulation results.....	48
4.3 Comparison between Matlab and PowerWorld methods.....	51
Chapter 5 Discussion and Conclusions.....	52
Appendix: Matlab code.....	54
References.....	76

## List of Tables

Table 4.1: First iteration Matlab results .....	40
Table 4.2: Seventh iteration Matlab results .....	41
Table 4.3: Estimation results for a 5-node network .....	42
Table 4.4: Errors CSE vs PSE 5-node network .....	43
Table 4.5: Errors CSE vs PSE 69-node network .....	44
Table 4.6: Errors CSE vs PSE 85-node network .....	44
Table 4.7: Average number of iterations between methods .....	44
Table 4.8: Number of iterations vs base power – 5 node network .....	45
Table 4.9: Number of iterations vs base power – 69 node network .....	46
Table 4.10: Number of iterations vs base power – 85 node network .....	46
Table 4.11: Maximum residues per iteration – 5 node network .....	46
Table 4.12: Maximum residues per iteration – 69 node network .....	47
Table 4.13: Maximum residues per iteration – 85 node network .....	47
Table 4.14: Iterations per convergence values and method – 5 node network .....	47
Table 4.15: Iterations per convergence values and method – 69 node network .....	48
Table 4.16: Iterations per convergence values and method – 85 node network .....	48
Table 4.17: Transmission Line Data NR vs SE – 5 node network .....	49
Table 4.18: Injected Power and Loads NR vs SE – 5 node network .....	50
Table 4.19: PowerWorld Results – 5 node network .....	50
Table 4.20: Comparison of results NR – Matlab and PowerWorld -Voltage .....	51
Table 4.21: Comparison of results NR – Matlab and PowerWorld -Phase .....	51

## List of Figures

Figure 1.1: Functional diagram of energy management system .....	11
Figure 3.1: 5-node radial network.....	33
Figure 3.2: 69-node radial network.....	34
Figure 3.3: 85-node radial network.....	34
Figure 4.1: PowerWorld NR vs SE system– 5 node system.....	49
Figure 4.2: PowerWorld simulation graphic for a 5 node network .....	50

## **Abbreviations**

CSE: Conventional State Estimation

PSE: Proposed State Estimation

NR: Newton Raphson

SE: State Estimation

## Chapter 1 Introduction

### 1.1 Introduction

State estimation algorithms for transport networks are based in both active and reactive power values. Line current measurements can be used to implement these algorithms, and when doing so, malfunctioning problems and non-convergence issues may occur [1]. On the other hand, in distribution networks the majority of available existent measurements are line currents and node voltages [2]. It is also found that the solution of the problem of state estimation by the least squares method sometimes presents a bad conditioning of the gain matrix. Solving a badly conditioned problem results in a proximity to the singularity of the coefficient matrix.

The use of line current measurements in the state estimation process leads to numeric and observability problems in the systems, as for example the cancelation of elements in the jacobian matrix [3] in the plain state, which means that those measures are useless when starting from plain state. These equations are non-linear, which can cause convergence difficulties in the iterative process.

In order to efficiently plan and operate electric power network, it is necessary to monitor the operating states of the system. With the purpose of controlling the system, state estimation is developed and comprises an important tool for network observation. The state of a system is defined generally by a vector of the bus voltage and the phase angle. This definition procedure is extensively used in energy control centers to provide an estimation of what is occurring in real-time. Forecasting loads ensures security and allows safe critical operation such as opening or closing of substation switches, load frequency control and dispatch. To process data, the state estimator uses:

- Measurement of variables in the system
- Model of the system

-Previous measurements or inputs/outputs

The development of the state estimation theory is also motivated by the existence of smart grids, decentralized renewable energy production and temporary network congestions. A main difference between conventional load flow and state estimation is that in the conventional method the results are calculated with measurements that are completely correct. On the other hand, state estimation is done with real-time values taken in different places of the system and transferred to a main center that may eventually include a small number of incorrect values.

There are several challenges that make state estimation need the use of redundant measures, from which should be mentioned that the information may not always be available from different measuring equipment, that is located in disperse locations, leading to errors that cannot be always automatically eliminated. The topology of the system is also not always completely defined.

To be mentioned is also that the management of the system performs functions of automatic control of the generated power, security of the system, and economic dispatch, and this feature is achieved by means of interconnecting the electric systems. This interconnection increases the overall reliability of the network, and the interconnected transmission lines are known as tie-lines.

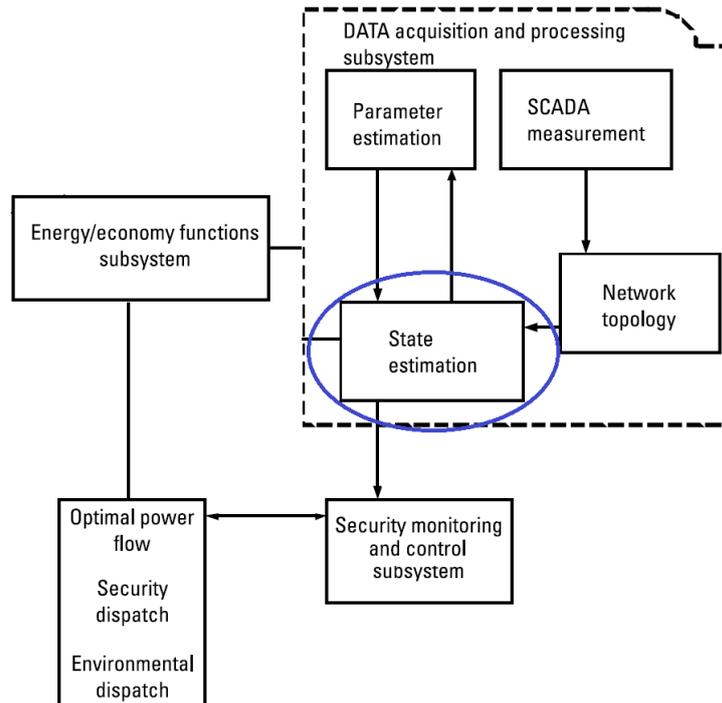


Figure 1.1: Functional diagram of energy management system [4]

A system is subjected to disturbances, and these disturbances cause undesirable frequency changes. Examples of such disturbances are changes in demand, losses in lines, outages and system failures. Renewable energy sources also cause load fluctuations when connected to conventional systems. The control operation intends to set and maintain this frequency as constant as possible throughout the entire electrical system. Therefore, disturbances and the unpredictability of renewable sources make state estimation the more important.

The notion of security in power systems is considered with regards of the prospect for the system to be working under contingent situations. Security will consist of how robust the system is towards imminent disturbances. Such notion is based on monitoring and control and is expressed in terms of the system state. This state will be a description that notes key information once it is known, and that will express variables of importance. Security is not defined towards every possible contingency, as it would unnecessarily oversize the protections, but towards a “feasible/possible” negative conditions. A normal state would fulfill security requirement if all reasonable contingencies result in normal operations. If a

contingence brings the system towards emergency, the state is no longer secure. A system that has a low probability of blackout is considered safe. The goal of the system control will be to stay within a safe state.

A modern-day control center is centralized and uses digital data, performing analyses.

- Load forecasting
- Planning
- Maintenance scheduling
- Monitoring
- State estimation
- Dispatch
- Frequency control

A security study is performed to verify robustness, as a typical power system is never in steady state as changes evolve constantly. In a restoration operation, the goal is to restore power after a loss, whereas in an emergency state, it is intended to decrease the pressure in the overcharged equipment. In order to assess the necessary measures to be taken is necessary to know the system states, by the placement of measurement devices on the system points.

The contingency analysis is another major function of the system, modeling problems before they arise. It is necessary as problems in the power system arise so quickly that they could not be solved unless previously predicted. It is valid for outage studies and serves as an indicator for annual investment plans. To achieve the optimum economic and safe operation, it is necessary to observe and control the operating states of the system. State estimation comprises then an important tool for this monitoring. It is traditionally described by the state vector of bus voltage magnitudes and phase angles, and it is used to provide an estimation of what occurs in the system and what will follow, based on the measurements and the model of

the system. State estimation will also assist the function of dispatch and load forecasting. Network applications need the current and voltage measurements in order to achieve an optimal performance, and gathering and treating data becomes of major importance due to the proliferation of deregulated systems in a renewable energy environment as well as smart grid implementation begins. The state estimator processes inputs following a particular algorithm, in a constant data acquisition process. The method is a modification of the original load flow calculations where data free from errors is used. What is more, state estimator may use a larger number of input data than the number of variables, which reduces the total number of measurements [4].

## **1.2 Purpose of this work**

Introducing line current measurements in the state estimation process constitutes an issue due to distribution networks characteristics. In order to overcome the difficulty, it is necessary to develop mathematical models that simulate the behavior of those networks. In a radial distribution network, when voltage magnitudes are available, a load flow solution always exists and is unique [5]. Reviewed techniques in order to solve the load flow problem based on radial networks include Newton –Raphson method, proposing a variable change. The problem appears when in the network are available, besides the voltage measurements, the line current magnitudes, which may lead to multiple solutions, losing then the unicity of the solution [6].

Based on Weighted Least Squares method, it will be intended to introduce alternative current line measurements, in order to reduce the multiplicity of available solutions and increase the convergence, by reducing the number of required iterations.

### **1.3 Scope of this work**

The following points have been treated throughout the present thesis work:

- Review published articles about state estimation that use line current measurements.
- Present several solving techniques for the WLS state estimator using the measurements.
- Formulate new state estimation models starting from those already published, seeking significant improvements.

To propose a state estimator, state variables were introduced, defining different radial networks, including nodes and taking measurements, intending to implement in MATLAB the method. Developed and previous methods were compared, also contrasting measured and estimated parameters. In order to solve the problem of cancelation of terms in the jacobian matrix, it was needed to start the estimation by non-plain profiles, or by not neglecting the shunt elements from the matrix in the first iteration [7]. To decrease the bad conditioning of the gain matrix, it was pursued numerical techniques that try to solve the problem, such using virtual or exact measurements. Throughout Lagrange methods or the Hachtel augmented matrix solutions were also pursued [8].

WLS state estimator solutions using line current magnitudes that avoid or decrease the problem of bad conditioning of the matrix were also developed throughout the thesis, as well as the formulation of new models for simulating distribution networks.

## **1.4 Assumptions**

The studied systems will be assumed to be in a normal state, without any contingencies or faults already existing on the nodes. No bad data will be purposely introduced in the model.

## **1.5 Definitions**

### State conditions

Emergency: Operating limit is violated. (Overload, under/over voltage and/or frequency)

Restorative: Load not fulfilled, causing a partial/total blackout in the system.

Normal: All load and operating conditions are satisfied.

### Type of measurements

Ordinary measurements: Defined as those that come from the state variables and from a real physical measurement from the system, by means of a current or voltage transformer.

Virtual measurements: Measures that are correct despite there is not a physical meter. They are treated as power injection measurements with small variance.

## Chapter 2 Review of Literature

### 2.1 Including current measurements in the generalized state estimation

The inclusion of line current measurements in the generalized state estimation is discussed in [9], creating a methodology that allows all current measurements to be included in the model. The generalized state estimator is based on a detailed representation of substations to the physical level. There exist two models to represent the substation: the complete model, also known as explicit, and the implicit model. The complete model increases the number of state variables with power flow through all the switches. The number of conventional state variables is very high, as a consequence of the detailed model adopted. Such number of state variables is compensated in a proportional way by the topological restrictions that represent the state of the switches, either on or off. This leads to a very large model that is the reason only reduced substations are modeled with it in detail. On the other hand, the implicit model of the substation reduces the size of the problem significantly, allowing all substations in the system to be represented, and from there, to include all the available internal measurements. Mathematically, the complete model can be represented by equations (2.1) – (2.4):

$$z_a = h_a(x_c, x_{CB}, x_a) + \varepsilon \quad (2.1)$$

$$c(x_c, x_a) = 0 \quad (2.2)$$

$$t(x_c, x_{CB}, x_a) = 0 \quad (2.3)$$

$$c_a(x_c, x_{CB}, x_a) = 0 \quad (2.4)$$

where:

$z_a$  : Magnitude measurements of voltage, power injections and power flows through the switches.

$h_a$  : Non-linear function measurements that relate  $z_a$  with the state variables  $x$ .

$x_c$  : State vector composed by voltage magnitudes and phase angles.

$x_{CB}$  : Flow through the switches

$x_a$  : Voltage magnitudes and phase angles

$\varepsilon$  : Error

$c$  : Structural restrictions

$t$  : Topological restrictions, (switch state on/off)

$c_a$  : Additional structural restrictions.

For the implicit model, the main goal is to minimize the number of additional variables that may be included in the state vector and avoid topological restrictions and also minimize the number of structural restrictions that remain in a model. An automatic way of choosing a critical state variable from the substation model is to define the own tree, which follows the following rules:

- Exclude all impedance branches different than zero.
- Exclude as many Open Loops as possible
- Include as many Closed Loops as possible
- For each electrical node, one of these incident injections or a null injection branch will be added to the tree when necessary.

The state vector is composed by the corresponding voltages of the tree branches and the power flows corresponding to the connections, excluding the lines/transformers. The extra set of state variables,  $x_e$ , added to the state vector by the implicit model it is exclusively made up by power injection and power flows through the switches, which are excluded from the own tree previously defined. Certain power measurements may be easily expressed in terms of  $x_c$  and  $x_e$ . On the other hand, due to the way of building the own tree, the topological restrictions consider  $x_{CB} = 0$ , variables that are eliminated from the model, reducing it to the following form in (2.5) and (2.6):

$$z_a = h_a(x_c, x_e) + \varepsilon \quad (2.5)$$

$$c(x_c) = 0 \quad (2.6)$$

In terms of minimizing the state vector, power flows (active or reactive),  $f_{ij}$ , in the substation can be expressed as shown in (2.7):

$$f_{ij} = \sum f_L(x_c) + \sum f_e \quad (2.7)$$

where  $f_L$  is the power flow through the regular branches with impedance different than zero, which is a function exclusively of the classic state variables  $x_c$ .  $f_e$  refers to the flows through connections of the own tree, included in  $x_e$ .

The generalized state estimator adopts an additional change with respect to the inclusion of current measurements associated with branches of null impedance or unknown added to the model, this is the case of currents that flow through the switches and also inject in different nodes of the external network, none of which may be solved by the conventional state estimator. On the other hand, current measurements associated with the switching devices may be expressed as shown in (2.8):

$$I_{ij} = \frac{\sqrt{P_{ij}^2 + Q_{ij}^2}}{V} \quad (2.8)$$

where  $I_{ij}$  is the current through the branch i-j inside of the substation  $P_{ij}$  and  $Q_{ij}$  are the active and reactive power and  $V$  is the voltage magnitude in the node. Note that the state variables can be expressed by means of (2.7). The jacobian elements can be obtained using one of the two methods.

(i) If  $P_{ij}$  and  $Q_{ij}$  are state variables, then use equations (2.9) – (2.11) as follows:

$$\frac{\partial I_{ij}}{\partial P_{ij}} = \frac{P_{ij}}{I_{ij}V^2} \quad (2.9)$$

$$\frac{\partial I_{ij}}{\partial Q_{ij}} = \frac{Q_{ij}}{I_{ij}V^2} \quad (2.10)$$

$$\frac{\partial I_{ij}}{\partial V} = -\frac{I_{ij}}{V} \quad (2.11)$$

(ii) If  $P_{ij}$  and  $Q_{ij}$  are not state variables, then use equations (2.12) – (2.13) as follows:

$$\frac{\partial I_{ij}}{\partial x} = \frac{1}{I_{ij}V^2} \left[ P_{ij} \frac{\partial P_{ij}}{\partial x} + Q_{ij} \frac{\partial Q_{ij}}{\partial x} \right] \quad (2.12)$$

$$\frac{\partial I_{ij}}{\partial v} = \frac{1}{I_{ij}v^2} \left[ P_{ij} \frac{\partial P_{ij}}{\partial v} + Q_{ij} \frac{\partial Q_{ij}}{\partial v} \right] - \frac{I_{ij}}{v} \quad (2.13)$$

It can be observed that there exists indetermination problems in the beginning of a plane profile. A way to avoid this would be to take  $I_{ij}^2$ , in which case the profiles of the jacobian will be determined using one of the following two methods.

(i) If  $P_{ij}$  and  $Q_{ij}$  are state variables, then use equations (2.14) – (2.16) as follows:

$$\frac{\partial I_{ij}^2}{\partial P_{ij}} = \frac{2P_{ij}}{v^2} \quad (2.14)$$

$$\frac{\partial I_{ij}^2}{\partial Q_{ij}} = \frac{2Q_{ij}}{v^2} \quad (2.15)$$

$$\frac{\partial I_{ij}^2}{\partial v} = -\frac{2I_{ij}^2}{v} \quad (2.16)$$

(ii) If  $P_{ij}$  and  $Q_{ij}$  are not state variables, then use equations (2.17) – (2.18) as follows:

$$\frac{\partial I_{ij}^2}{\partial x} = \frac{2}{I_{ij}v^2} \left[ P_{ij} \frac{\partial P_{ij}}{\partial x} + Q_{ij} \frac{\partial Q_{ij}}{\partial x} \right] \quad (2.17)$$

$$\frac{\partial I_{ij}^2}{\partial v} = \frac{2}{I_{ij}v^2} \left[ P_{ij} \frac{\partial P_{ij}}{\partial v} + Q_{ij} \frac{\partial Q_{ij}}{\partial v} \right] - \frac{2I_{ij}^2}{v} \quad (2.18)$$

Also the injected current in the node I can be expressed as shown in (2.19):

$$I_i = \frac{\sqrt{P_i^2 + Q_i^2}}{v} \quad (2.19)$$

The previous expressions are valid and may replace  $I_{ij}$ ,  $P_{ij}$ ,  $Q_{ij}$  with  $I_i$ ,  $P_i$  and  $Q_i$  respectively. The model was tried on different scenarios, having generated a set of simulations with measurements that contain Gaussian errors, having compared the proposed method against the conventional method and the proposed method using current measurements and also using squared current measurements. The trials were made in presence of topological error and its behavior towards wrong data when working with exact measurements. The generalized state estimator showed in [10] that the inclusion of certain current measurements which have been neglected until now or considered in use as redundant

measurements present satisfactory results and reinforces the capability of detecting and identifying topological and analogical errors.

## 2.2 State estimation based in a line current

This article [11] published in parallel with [12], presented a method for the state estimation in which the set of available measurements, consists of voltage magnitudes in the nodes and line currents. The addressed problem is analyzing the observability  $P - \theta$  that may be achieved by providing inequality restrictions that are added to the estimating process. These restrictions simply indicate a sign convention of  $P \geq 0$  for a generating node and  $P \leq 0$  for a load node. The simulation results confirmed that the method could be applicable for a distribution network of medium size, as long as the radial network has more than just one feeder node. Given the mentioned available measurements, the line current measurement between nodes  $ij$  can be expressed as shown in (2.20) – (2.24):

$$|I_{ij}| = |AV_i^2 + BV_j^2 - 2V_iV_j(C \cos\theta_{ij} - D \sin\theta_{ij})|^{1/2} \quad (2.20)$$

where:

$$A = g_{ij}^2 + (b_{ij} + b_{sh}/2)^2 \quad (2.21)$$

$$B = g_{ij}^2 + b_{ij}^2 \quad (2.22)$$

$$C = g_{ij}^2 + b_{ij}(b_{ij} + b_{sh}/2) \quad (2.23)$$

$$D = g_{ij}b_{sh}/2 \quad (2.24)$$

and  $g$  and  $b$  are the line parameters.

The observability analysis of the proposed method by this article, is intended for distribution networks, where the shunt parameters are neglected. Also it is necessary to know the active power in each node and that dependent if the node is either load or generation, can be assumed as positive or negative or even zero. Knowing these values may be included in the

estimation process in the form of inequality restrictions. The estimation problem is then converted into minimizing a non-linear function  $h$  subjected to non-linear inequality restrictions. The minimizing to non-linear statement can be mathematically expressed as shown in (2.25):

$$\text{Min } J(x) = [z - h(x)]^T W [z - h(x)] \quad (2.25)$$

$$\text{where } c(x) \geq 0$$

These inequalities  $c(x)$  are the known active power flow equations, existing in each of the sides of each of the network lines. The inequality restrictions are not useful when there are sufficient available power flows or injections. Nevertheless, certain areas where those measurements do not guarantee the observability, it may be made observable by using the described method. It is used in the article the squared line current measurements with a standard deviation of twice the normal. The expression of the measured current elevated to the square for line  $ij$ , neglecting now the shunt parameters is the following (2.26):

$$I_{ij}^2 = [(g_{ij}^2 + b_{ij}^2)(V_i^2 - V_j^2 - 2V_iV_j\cos\theta_{ij})] \quad (2.26)$$

From the previous expression (2.26), knowing the voltage magnitudes and the value of the line current can be obtained the absolute value of phase displacement between the sides of the line  $ij$ , as shown in (2.27).

$$\cos \theta_{ij} = \frac{V_i^2 + V_j^2 - I_{ij}^2(r_{ij}^2 + x_{ij}^2)}{2V_iV_j} \quad (2.27)$$

The determination of the power flow directions in the lines, will determine the true sign of the phase angle between the line sides, with which it could be stated that two possible solutions exist [13]. The article determines rules for the determination of the direction of the power flow in the line and uses exact and real measurements. The problem is to define the sign of the phase angle, and with it to define the inequality restriction, but some problems are presented when there appear similar phase angle values in some lines and is to be solved by adding more restrictions or measurements so it exists more redundancy in the estimation

process. The second problem appears when a line is so short or it has an insignificant load so the phase angle is so small that is in the same order of magnitude of the convergence values. That would be the case of the state estimator that cannot identify the correct state of the flow direction. One solution is to consider it a super-node. It can be concluded in this article, that the proposed model provides, on top of filtering redundant measurements, a real time load flow. Nevertheless, in order to increase reliability, it is strongly recommended to include as many restrictions as possible. The augmented Lagrange method is the solution taken for the proposed state estimator, for optimizing power flow, with satisfactory results [14].

### **2.3 State estimation based in branch current for distribution systems**

In article [15] there is a three-phase state estimator for distribution networks, based on current line measurements. It is a method adapted to low meshed distribution networks and more efficient than the conventional state estimator. The proposed method introduced the real and imaginary components of the branch currents as state variables in place of the conventional variables that are voltage and phase in the nodes. The line current defines the system state because if the branch current is known, loads and voltages in the nodes may be determined. The measurement function such as the power flows and load measurements in the nodes are expressed as a function of this new state variables, which have a linear behavior. The line current measurement expression is a linear function and the voltage magnitudes are ignored, with the exception of the voltage measurement in the substation node that is considered as a reference node. The measurement function of the line current presents indetermination problems in the jacobian as starting the plane profile due to its nonlinear behavior, a way of solving this is by excluding the measurements in the first iterations and then introducing them in the subsequent measurements. The article considers that the proposed method is computationally efficient due to the decoupling of phases and also given that the gain matrix

is independent from the line parameters. The observability analysis, in the numerical observability method is necessary to ensure that the gain matrix is not singular and that is insured by the available sufficient measurements in the LU factorization.

#### **2.4 Multiple solution detection in state estimation in presence of line current measurements**

Article [16] is based on the problem of observability when the networks contain line current measurement magnitudes. The use of current measurements may lead to multiple solutions, and also the unique observability of the network may be lost for specific measurement configurations. A simple method based on the residues covariance matrix is considered, that intends to detect the possibility of multiple solutions for a determined set of measurements. The method is based over the formulation of the conventional state estimator with inequality restrictions. The method is valid for systems that include voltage magnitudes, power flows and injections, and line current measurements. The use of inequality restrictions may guarantee that the unique observability is true. The method used indicates if such measurements are added or not to the calculations, as well as the use of inequality restrictions, or pseudo-measurements that will be added so that the given system has unique observability.

The algorithms for the observability analysis are based in the supposition that the system solution may be found with the state estimation problem and is unique. In order to do so is necessary to have as available measurements the active and reactive power in order to use the decoupled problems. Such natural uncoupling between active and reactive measurements is lost when the measurements are included in the set of available measurements, the line current magnitudes. When line current measurements are used it may be found that the system is observable, but there may be multiple solutions, a complete range of the Jacobian

matrix does not necessarily imply that the observability of the system is unique. In this case the set of measurements is made up entirely by voltage magnitudes and line currents as well as pairs of power flow (P-Q) and power injections. The initialization of the iterative process in the state estimation problem is important, as when there exist multiple solutions may converge to the solution that is closer to the initial point, especially when the voltage magnitude measurements may contain errors. For the determination of the unique solution, it is used a numerical method based on the measurement of the residues of the covariance matrix. Considering the linearized measurements given by (2.28):

$$\Delta z = H\Delta x + e \quad (2.28)$$

where  $H = \frac{\delta h(x)}{\delta x}$ ,  $h(x)$  are the non-linear measurement functions that relate  $z$  with the state variables  $x$ , and  $e$  represents the error measurements. Using the estimator Weighted Least Squares, the residues of the measurements may be related following (2.29):

$$r = Se \quad (2.29)$$

where  $S$  is as (2.30):

$$S = (I - H(H^T R^{-1} H)^{-1} H^T R^{-1}) \quad (2.30)$$

$R$ : covariance matrix of the measurement errors

$r = \hat{z} - z$  Residue of the measurements

$\hat{z}$  Estimated measurements

The covariance matrix of the residues can be written as (2.31):

$$C = cov(r) = SR \quad (2.31)$$

Starting from the previous expression it is presented the process to detect possible solutions.

It is assumed that the network has a solution and the possibility of multiple solutions depends on the set of available measurements. The process is as follows:

- Calculate the columns of  $C$ , corresponding on the current magnitudes.

- If one column k, contains an input different than zero that corresponds to a power flow measurement or injection, then skip that column. If not, mark the current measurement along with the other measurements with entries different than zero in that column as a component of the residue v-i capable of giving multiple solutions. If the column is completely zero, then indicate the current measurement as critical measurement.

## 2.5 Line current measurement in state estimation

From article [17] it is found the method to overcome the mathematical indetermination problems found when the current measurements are used in state estimation. The measurements used are the current measurements elevated to the square, also by modifying the values of the standard deviation of these measurements and the new values of the Jacobian are well mathematically defined for all the calculated conditions [18]. From the pi-equivalent model, the equation that relates the current magnitude with the state variable in a branch connecting nodes i and j follows (2.32) – (2.35):

$$|I_{ij}| = [AV_i^2 + BV_j^2 - 2CV_iV_j \cos(\theta_i - \theta_j)]^{1/2} \quad (2.32)$$

where:

$$A = (g_{ij} + g_{si})^2 + (b_{ij} + b_{si})^2 \quad (2.33)$$

$$B = g_{ij}^2 + b_{ij}^2 \quad (2.34)$$

$$C = g_{ij}(g_{ij} + b_{si}) + b_{ij}(b_{ij} + b_{si}) \quad (2.35)$$

By considering as a measurement of the square of the line current measurement, the terms corresponding to the Jacobian will be as (2.36) – (2.39):

$$\frac{\delta I_{ij}^2}{\delta V_i} = 2[AV_i - CV_j \cos(\theta_i - \theta_j)] \quad (2.36)$$

$$\frac{\delta I_{ij}^2}{\delta V_j} = 2[BV_j - CV_i \cos(\theta_i - \theta_j)] \quad (2.37)$$

$$\frac{\delta I_{ij}^2}{\delta \theta_i} = 2[CV_i V_j \sin(\theta_i - \theta_j)] \quad (2.38)$$

$$\frac{\delta I_{ij}^2}{\delta \theta_j} = 2[-CV_i V_j \sin(\theta_i - \theta_j)] \quad (2.39)$$

The detailed expressions may be evaluated under plane profile conditions and do not present indetermination problems in the state estimation [19]. In order to use the squared value as a measurement, it is necessary to determine their modified variances and that are necessities in the process of detection and identification of bad data [20]. It may be considered as the product of two random measurements  $z_k$  and  $z_m$  with their corresponding standard deviations  $\sigma_k$  and  $\sigma_m$  lead to a resulting new variable  $z_n$  with the standard deviation  $\sigma_n$ , as (2.40):

$$\begin{aligned} z_n \pm \sigma_n &= (z_k \pm \sigma_k) \cdot (z_m \pm \sigma_m) \\ &= z_k \cdot z_m \pm (z_k \cdot \sigma_m + z_m \cdot \sigma_k) + \sigma_k \cdot \sigma_m \end{aligned} \quad (2.40)$$

If it is considered that both random measurements are equal and belong to a line current measurement, leads to  $z_k = z_m = |I_{ij}|$  and the standard deviation is  $\sigma_k = \sigma_m = \sigma_{|I_{ij}|}$ . In this case, from equation (2.40) it is inferred that the standard deviation to the square be as (2.41):

$$\sigma_{|I_{ij}|^2} = 2|I_{ij}|\sigma_{|I_{ij}|} + \sigma_{|I_{ij}|}^2 \quad (2.41)$$

Simulations made in the article use equation (2.41) for the calculation of the standard deviation, used in the process of detecting and identifying wrong data with good results. Observability analysis through line currents are necessarily two measurements on each line and considers three possible cases that go together, such as the current flow with a flow of reactive flow, and lastly with a voltage magnitude [21]. Estimations done over a system considered as available measurements the voltage magnitudes in every node and the line currents, show that starting from a plane profile with correct direction of the power flow in the lines, the final estimated reached value is always correct. Otherwise it always converges to an incorrect solution, but in this case it shows higher sensitivity to the estimator, both

solutions make the network observable, plus the degree of redundancy of the used measurement set is almost minimal. The other case is presented when the available measurements are composed of a set of voltages, active and reactive flows, and line currents. The estimator led to correct results, even including incorrect measurements or even topological errors [22].

## **2.6 WLS state estimator using line current measurements**

In this section it will be detailed the solution techniques for the state estimator WLS using line current measurements, which will be the method to be used. The solution to the formulation of the problem of state estimation by minimum squares, uses the normal equations, but in some cases it presents a bad conditioning of the gain matrix. Solving a bad conditioned problem results in a proximity to the singularity of their matrix of coefficients [23]. Bad conditioning in the gain matrix may be due to different causes, in which it could be mentioned:

- Simultaneously using very high weights for the virtual measurements (transit nodes) and relatively low for the rest of measurements.
- Working with networks that are incident in their nodes with very long and very short lines simultaneously.
- Using as available measurements an elevated percentage of injection measurement instead of flow measurements.

In large radial distribution networks, it is produced a bad conditioning when the election of the base power is not adequate and especially if we use line current measurement magnitudes [24]. To reduce that bad conditioning of a gain matrix, several numerical techniques appear that intend to solve the problem, between these it may be mentioned: using exact or virtual

measurements, implementing equality restrictions through Lagrange multipliers and by applying the Hachtel augmented matrix.

### 2.6.1 Modeling of current measurements

For a branch connected between general nodes  $i$  and  $j$ , the equation that relates the current magnitude with the state variables is shown in the equation (2.32) which shows a non-linear relationship. If shunt parameters are discarded, it is obtained (2.42):

$$I_{ij} = [(g_{ij}^2 + b_{ij}^2)(V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij})]^{1/2} \quad (2.42)$$

It can be observed how the term  $\sin \theta_{ij}$  from equation (2.32) disappears, which was one of the main problems when using current magnitude measurements.

### 2.6.2 Difficulties in the use of current measurements

The use of current measurements in the process of state estimation leads to several numerical and/or observability issues, in which it could be mentioned that for the case of starting in the plain state, the elements of the jacobian that correspond to the current measurement are undefined if the value of  $I_{ij}$  is used directly, or else zero when it is used  $I_{ij}^2$ , which means that the current measurements are not useful when it is started by the plain state [25]. Therefore, to solve this issue it is necessary to initialize the estimation process with a non-plain profile or by not neglecting the shunt parameters in the first iteration [26]. The non-linearity of  $I_{ij}$  in the equation (2.42) causes convergence difficulties in the iterative process and is then more convenient to use  $I_{ij}^2$ . If current measurements are considered and there are no power measurements, starting by equation (2.39), the phase angles can be calculated with the expression (2.43):

$$\cos \theta_{ij} = \frac{V_i^2 + V_j^2 - I_{ij}^2(r_{ij}^2 + x_{ij}^2)}{2V_i V_j} \quad (2.43)$$

Starting by equation 3.2 it can be found two solutions for the displacement angle  $\pm\theta_{ij}$ . This leads to obtaining two different solutions for the power flows, due to the presence of the term “ $\sin \theta_{ij}$ ”. The expressions of the active and reactive power flow that neglect the shunt parameters are (2.44) – (2.45).

$$P_{ij} = V_i^2 g_{ij} - V_i V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) \quad (2.44)$$

$$Q_{ij} = -V_i^2 b_{ij} - V_i V_j (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}) \quad (2.45)$$

If expressions (2.44) – (2.45) are introduced in the equation (2.43), obtaining expressions (2.46) – (2.47):

$$P_{ij} = -b_{ij} V_i V_j \sin \theta_{ij} + \frac{1}{2} [g_{ij} (V_i^2 - V_j^2) + I_{ij}^2 r_{ij}] \quad (2.46)$$

$$Q_{ij} = -\frac{1}{2} [b_{ij} (V_i^2 - V_j^2) - I_{ij}^2 x_{ij}] - g_{ij} V_i V_j \sin \theta_{ij} \quad (2.47)$$

From the equations (2.46) – (2.47) it is shown the multiplicity of solutions, if only the current measurements are considered. A proper combination of available measurements from current magnitude as well as power flows allow to analyze the multiplicity of solutions problem, which would lead to the unique solution of the system [27].

## 2.7 Solution through virtual measurements

The problem of state estimation WLS through virtual measurements, is presented as (2.48):

$$\text{Min } J(x) = [z - h(x)]^T W(x) [z - h(x)] \quad (2.48)$$

where  $z$  is formed by ordinary measurements and measurements exact or virtual, which result from expressing the equation (2.42) as shown in (2.49):

$$f(x) = I_{ij}^2 - [(g_{ij}^2 + b_{ij}^2)(V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij})] = 0 \quad (2.49)$$

If  $f(x) = 0$  it is considered virtual measurements.

If  $H$  denotes the jacobian of the ordinary measurements and  $C$  the jacobian of virtual measurements, in that case the gain matrix will be as (2.50):

$$G = \begin{bmatrix} H \\ C \end{bmatrix}^T \begin{bmatrix} W & 0 \\ 0 & W_1 \end{bmatrix} \begin{bmatrix} H \\ C \end{bmatrix} \quad (2.50)$$

where:

$W$ : Weights assigned to ordinary measurements

$W_1$ : Weights assigned to the virtual measurements, of high value.

Then the state estimator leads to the iterative solution as following (2.51):

$$[G(x^k)][\Delta x^k] = \begin{bmatrix} H \\ C \end{bmatrix}^T \begin{bmatrix} W & 0 \\ 0 & W_1 \end{bmatrix} [\Delta z^k] \quad (2.51)$$

where the incremental steps are as (2.52) – (2.53):

$$\Delta x^k = x^{k+1} - x^k \quad (2.52)$$

$$\Delta z^k = z - h(x^k) \quad (2.53)$$

## 2.8 Solution through equality restrictions

This method is used when in the problem of state estimation, it is available a set of measurements that include exact values. In the case of distribution systems, it is presented the equation of line current magnitude as a function of the state variables and considered as an equality restriction, such as is the case in equation (2.49) as indicated by measure  $f(x) = 0$  [28]. The problem of WLS state estimation with equality restrictions, is presented on (2.54)

$$\text{Min } J(x) = \frac{1}{2} [z - h(x)]^T W(x) [z - h(x)] \quad (2.54)$$

subjected to  $f(x) = 0$

where  $f(x)$  represents the equality restrictions.

The problem presented can be solved by building the Lagrange function, as shown in (2.55):

$$\mathcal{L} = J(x) - \lambda^T f(x) \quad (2.55)$$

and the first order optimality conditions are obtained as (2.56) – (2.57):

$$\frac{\partial \mathcal{L}}{\partial x} = 0 \rightarrow H^T W [z - h(x)] + C^T \lambda = 0 \quad (2.56)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \rightarrow f(x) = 0 \quad (2.57)$$

where  $C$  is the jacobian of  $f(x)$

Applying the Gauss-Newton method, the solution of the non-linear system is obtained

iteratively through the following linear system, as shown in (2.58):

$$\begin{bmatrix} H^T W H & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ -\lambda \end{bmatrix} = \begin{bmatrix} H^T W \Delta z^k \\ -f(x^k) \end{bmatrix} \quad (2.58)$$

The solution to (2.58) presents several drawbacks when there exists a bad scaling between the values of the matrix coefficients  $H^T W H$  and the coefficients of the jacobian matrix  $C$ , and, as a consequence, it occurs a bad conditioning.

## 2.9 Hachtel augmented matrix

This formulation, as compared with the previous equality restrictions case, also considers as equality restrictions the equations of the residues  $r$ . The optimization problem is then proposed in the form as (2.59) – (2.60):

$$\text{Min } J(x) = \frac{1}{2} r^T W(x) r \quad (2.59)$$

$$\text{subjected to } f(x) = 0$$

$$r - z + h(x) = 0 \quad (2.60)$$

The Lagrange function that results has two subsets of Lagrange multiplications as (2.61):

$$\mathcal{L} = J(x) - \lambda^T c(x) - \mu^T (r - z + h(x)) \quad (2.61)$$

and the optimality conditions are expressed by equations (2.62) – (2.65):

$$\delta \mathcal{L} / \delta x = 0 \rightarrow C^T \lambda + H^T \mu = 0 \quad (2.62)$$

$$\delta\mathcal{L}/\delta\lambda = 0 \rightarrow f(x) = 0 \quad (2.63)$$

$$\delta\mathcal{L}/\delta r = 0 \rightarrow Wr - \mu = 0 \quad (2.64)$$

$$\delta\mathcal{L}/\delta\mu = 0 \rightarrow r - z + h(x) = 0 \quad (2.65)$$

where  $C$  is the jacobian of  $f(x)$  (virtual measurements) and  $H$  the jacobian of ordinary measurements. The equation  $Wr - \mu = 0$  allows to eliminate  $r$  ( $r = R\mu$ ) and the remainder of equations generates the system as shown in (2.66):

$$\begin{bmatrix} R & H & 0 \\ H^T & 0 & C \\ 0 & C & 0 \end{bmatrix} \begin{bmatrix} \mu \\ \Delta x \\ \lambda \end{bmatrix} = \begin{bmatrix} \Delta z^k \\ 0 \\ -f(x^k) \end{bmatrix} \quad (2.66)$$

The coefficients of the matrix in (2.66) are denominated Hachtel augmented matrix, and it presents a better conditioning with regards to (2.58) if a proper scaling in between the coefficients of  $H$  and  $C$  are achieved [29].

## Chapter 3 Methodology

### 3.1 Network description

The development of the present chapter introduces a new formulation of the problem of the state estimation taking into account the characteristics of a distribution system. In contrast with the conventional state estimator that uses as state variables voltage magnitudes and phase angles, in the proposed model it will be used new state variables in such way that a linear model subjected to quadratic restrictions is obtained. The main characteristic of the results of this model is that a set of own measurements is used. This set are the voltage magnitudes in the nodes and the line current measurement in the branches. This study is consistent with the distribution system model developed in chapter 2.

As an illustrative example a radial network of 5 nodes as shown in figure (3.1) will be used to show the new formulation of the state estimation problem. Similar studies will be done for larger networks with 69 and 85 nodes as shown in figure (3.2) and figure (3.3), respectively.

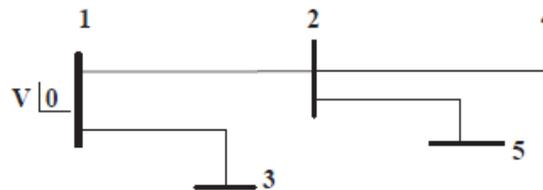


Figure 3.1: 5-node radial network

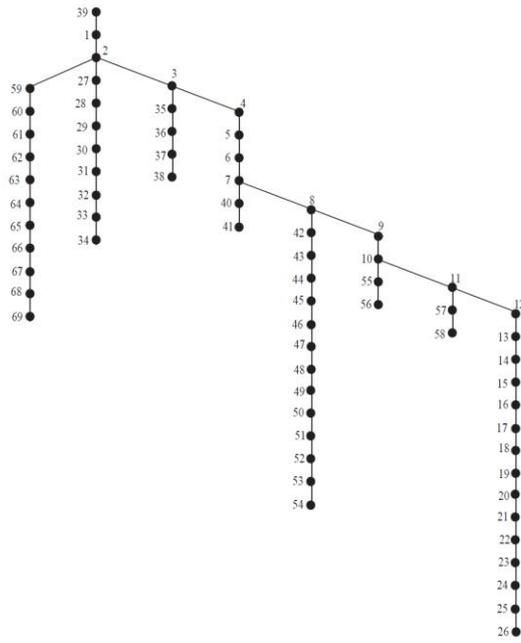


Figure 3.2: 69-node radial network

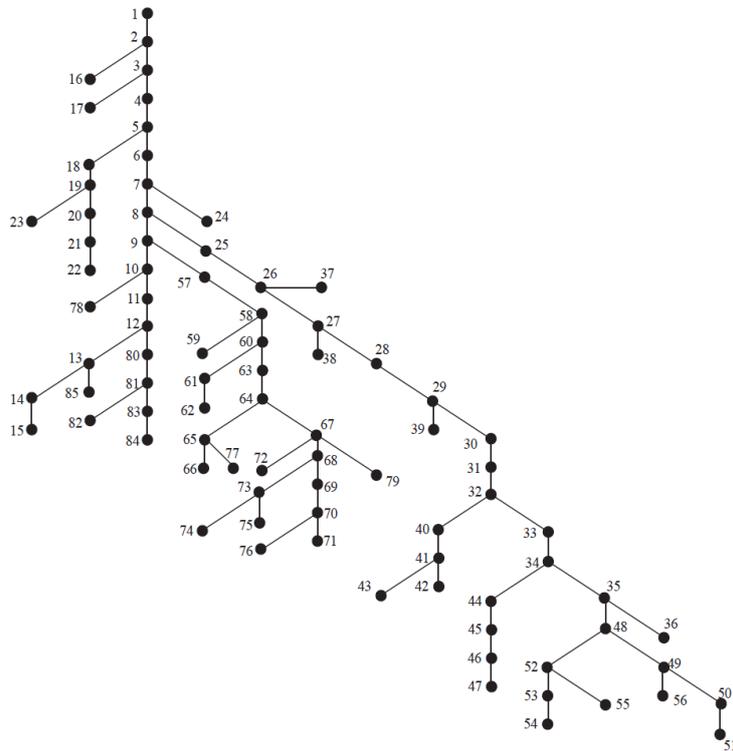


Figure 3.3: 85-node radial network

### 3.2 Proposed state estimation variables

Having into account previously shown equations (2.46) – (2.47) results in (3.1) – (3.2):

$$P_{ij} = -b_{ij}V_iV_j \sin \theta_{ij} + \frac{1}{2}[g_{ij}(V_i^2 - V_j^2) + I_{ij}^2r_{ij}] \quad (3.1)$$

$$Q_{ij} = -\frac{1}{2}[b_{ij}(V_i^2 - V_j^2) - I_{ij}^2x_{ij}] - g_{ij}V_iV_j \sin \theta_{ij} \quad (3.2)$$

In an electrical power distribution system, the flows of active and reactive power in the lines may be expressed as a function of the node voltages and the line current magnitudes, as shown in (3.3):

$$I_{ij} = [(g_{ij}^2 + b_{ij}^2)(V_i^2 + V_j^2 - 2V_i^2V_j^2 \cos \theta_{ij})]^{1/2} \quad (3.3)$$

Equation (3.3) relates the line current magnitude with the voltage magnitude in the nodes and the phase angles in the sides of the line. The state estimator proposed introduces the set of measurements as shown in (3.4) – (3.6):

$$U_i = V_i^2 \quad (3.4)$$

$$J_{ij} = I_{ij}^2 \quad (3.5)$$

$$W_{ij} = V_iV_j \sin \theta_{ij} \quad (3.6)$$

In terms of the new state variables, equations (2.46) – (2.47) can be expressed in a linear way for the second set of measurements as (3.7) – (3.8):

$$P_{ij} = -b_{ij}W_{ij} + \frac{1}{2}[g_{ij}(U_i - U_j) + J_{ij}r_{ij}] \quad (3.7)$$

$$Q_{ij} = -\frac{1}{2}[b_{ij}(U_i - U_j) - J_{ij}x_{ij}] - g_{ij}W_{ij} \quad (3.8)$$

On the other hand, from virtual measurements leads to (3.9):

$$f(x) = I_{ij}^2 - [(g_{ij}^2 + b_{ij}^2)(V_i^2 + V_j^2 - 2V_iV_j \cos \theta_{ij})] \quad (3.9)$$

and introducing the variable changes, it leads to the quadratic expression (3.10) for each branch:

$$W_{ij}^2 + \frac{[U_i + U_j - J_{ij}z_{ij}^2]^2}{4} - U_i U_j = f(x) = 0 \quad (3.10)$$

It can be observed then, that using the set of state variables proposed, any available measure is a linear function of the state vector. In a compact way, it is obtained the model shown in (3.11):

$$\begin{cases} z = Hx + e \rightarrow \text{linear} \\ f(x) = 0 \rightarrow \text{quadratic} \end{cases} \quad (3.11)$$

This leads to the following State Estimator WLS with equality restrictions shown in (3.12):

$$\begin{aligned} \text{Min } J(x) &= \frac{1}{2} [z - Hx]^T W [z - Hx] \\ \text{subjected to } &f(x) = 0 \end{aligned} \quad (3.12)$$

In the state estimator proposed, it can be observed that it is used state variables in the branches, and to develop that state estimation process is not necessary to have a reference value as it is the case with the Conventional State Estimator, that uses a phase angle as reference. For the solution of the problem it will be used two different techniques: Lagrange multipliers and virtual measurements, both techniques have been developed on chapter 1. For the case of solution through Lagrange multipliers, it will be obtained the first order optimality conditions as (3.13) – (3.14):

$$\delta \mathcal{L} / \delta x = 0 \rightarrow H^T W [z - Hx] + C^T \lambda \quad (3.13)$$

$$\delta \mathcal{L} / \delta \lambda = 0 \rightarrow f(x) = 0 \quad (3.14)$$

where C is the jacobian of f(x).

Applying the Gauss-Newton method, the solution of the non-linear system (3.12), is obtained iteratively through the following non-linear system, shown in (3.15):

$$\begin{bmatrix} H^T W H & C^T \\ C & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ -\lambda \end{bmatrix} = \begin{bmatrix} H^T W \Delta z^k \\ -f(x^k) \end{bmatrix} \quad (3.15)$$

where the increments are shown in (3.16) – (3.17):

$$\Delta x = x^{k+1} - x^k \quad (3.16)$$

$$\Delta z^k = z - H(x^k) \quad (3.17)$$

The structure of the equation (3.15) will be illustrated supported by a 5 nodes radial network as shown in figure (3.1).

### 3.3 Iterative process description

The measurements will be the voltage magnitudes in every node. The flows of active power and flows of line current will be the measurements in all the branches. Having that into account, the Jacobian of the measurement, will be as shown in equation (3.18):

$$H = \begin{matrix} & \begin{matrix} \partial U_1 & \partial U_2 & \partial U_3 & \partial U_4 & \partial U_5 & \partial J_{12} & \partial J_{13} & \partial J_{24} & \partial J_{25} & \partial W_{12} & \partial W_{13} & \partial W_{24} & \partial W_{25} \end{matrix} \\ \begin{matrix} \partial U_1 \\ \partial U_2 \\ \partial U_3 \\ \partial U_4 \\ \partial U_5 \\ \partial J_{12} \\ \partial J_{13} \\ \partial J_{24} \\ \partial J_{25} \\ \partial W_{12} \\ \partial W_{13} \\ \partial W_{24} \\ \partial W_{25} \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{\partial J_{12}}{\partial W_{12}} & \frac{g_{12}}{2} & -\frac{g_{12}}{2} & 0 & 0 & 0 & \frac{r_{12}}{2} & 0 & 0 & 0 & -\frac{b_{12}}{2} & 0 & 0 \\ \frac{\partial J_{13}}{\partial W_{13}} & -\frac{g_{13}}{2} & 0 & \frac{g_{13}}{2} & 0 & 0 & 0 & \frac{r_{13}}{2} & 0 & 0 & 0 & -\frac{b_{13}}{2} & 0 \\ \frac{\partial J_{24}}{\partial W_{24}} & 0 & \frac{g_{24}}{2} & 0 & -\frac{g_{24}}{2} & 0 & 0 & 0 & \frac{r_{24}}{2} & 0 & 0 & 0 & -\frac{b_{24}}{2} \\ \frac{\partial J_{25}}{\partial W_{25}} & 0 & \frac{g_{25}}{2} & 0 & 0 & -\frac{g_{25}}{2} & 0 & 0 & 0 & \frac{r_{25}}{2} & 0 & 0 & -\frac{b_{25}}{2} \end{pmatrix} \end{matrix} \quad (3.18)$$

It can be observed that the Jacobian is a function only of the network parameters, as a consequence of the linearity of the model, and the inclusion of current measurements is direct. The jacobian associated to the quadratic restrictions (3.12) is obtained by applying the expressions shown in (3.19) – (3.22).

$$\frac{\partial f(x)}{\partial U_i} = \frac{(U_i + U_j - J_{ij} z_{ij}^2)}{2} - U_j \quad (3.19)$$

$$\frac{\partial f(x)}{\partial U_j} = \frac{(U_i + U_j - J_{ij} z_{ij}^2)}{2} - U_i \quad (3.20)$$

$$\frac{\partial f(x)}{\partial J_{ij}} = \frac{z_{ij}^2 (U_i + U_j - J_{ij} z_{ij}^2)}{2} \quad (3.21)$$

$$\frac{\partial f(x)}{\partial W_{ij}} = 2W_{ij} \quad (3.22)$$

In order to initialize the iterative process, two different ways have been tried:

1. Start as  $U_i^0 = 1, J_{ij}^0 = 0; W_{ij}^0 = 0$  which is equivalent to the plain profile of the starting point in the CSE ( $\theta_i^0 = 0, V_i^0 = 1$ ). In this initialization it is obtained  $f(x^0) = 0$  and all the columns of matrix C are zero, with the exception of those that are associated with the state variable  $J_{ij}$ . This leads to a bad scaling of the iterative solution from equation (3.15) and the number of iterations to converge to the final estimated value is higher with respect to the CSE.
2. If there are enough available measurements, it is possible to obtain a first solution by solving a linear problem by starting by the plain profile without considering the equality restrictions. Equation (3.15) is then simplified, resulting in (3.23):

$$[H^TWH][\Delta x^1] = [H^TW\Delta z^0] \quad (3.23)$$

In a single iteration of the method shown in (3.23), it is obtained values that represent an initial state closer to the final estimated value. This option allows to decrease the number of iterations needed in order to get to the final solution. After applying the iterative process from equation (3.15) it is obtained an approximate solution. In the first obtained solution it can be observed how the problem of bad scaling is solved and also the convergence to the final estimated value is faster.

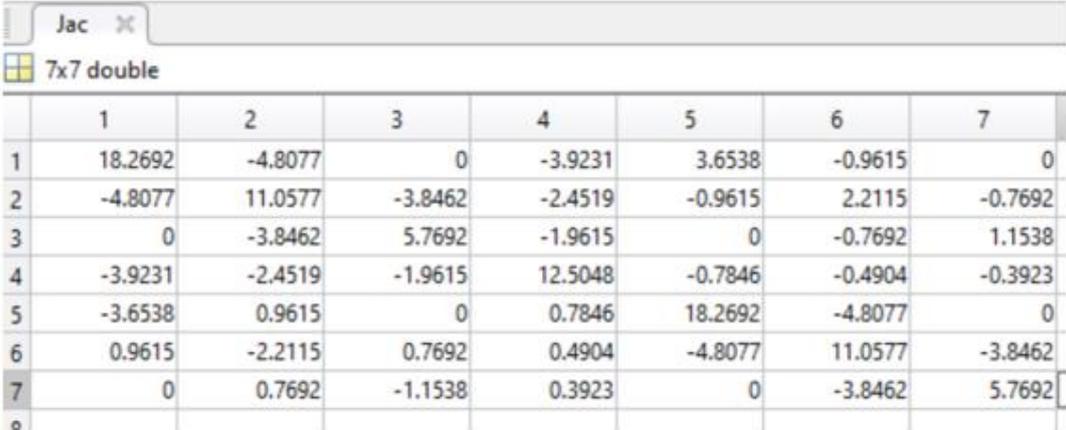
### **3.4 Description of Matlab code**

The process of creating the Matlab file will be briefly summarized in this section. The number of nodes will be an input, and for each one of the nodes, it will be asked whether there is an interaction between nodes or not. Active and reactive powers, as well as weights will be other inputs of the system. Voltage vector for the measured and as well for the calculated voltage will also be stored. Line impedances for those nodes in between which an interaction occurs will be required. Subsequently, admittance matrix will be calculated. Injected power to nodes will also be demanded to input user. Sample number were introduced as values for the line impedances and injected powers. The jacobian will then be calculated according to conventional state estimator and proposed conventional estimator. The jacobian matrix, altogether with the gain matrix, will lead to the state estimated solution and a counter for the number of iterations will be increased, indexes will be updated, and the number of iterations will continue increasing until the difference between the previously calculated iterative solution and the current last solution are smaller than an established order of convergence, for example error smaller than  $10^{-2}$ . State vector will then be calculated and will include all the state variables.

## Chapter 4 Analytical and Simulation Results

### 4.1 Matlab analysis

The power flow analysis on the proposed system calculates, as intermediate result, the jacobian matrix. Through this jacobian matrix the final state is calculated using the Newton Raphson method. The mentioned matrix is described in equation (4.1) for means of explanation and inner matrix dimension verification.



$$H = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 18.2692 & -4.8077 & 0 & -3.9231 & 3.6538 & -0.9615 & 0 \\ -4.8077 & 11.0577 & -3.8462 & -2.4519 & -0.9615 & 2.2115 & -0.7692 \\ 0 & -3.8462 & 5.7692 & -1.9615 & 0 & -0.7692 & 1.1538 \\ -3.9231 & -2.4519 & -1.9615 & 12.5048 & -0.7846 & -0.4904 & -0.3923 \\ -3.6538 & 0.9615 & 0 & 0.7846 & 18.2692 & -4.8077 & 0 \\ 0.9615 & -2.2115 & 0.7692 & 0.4904 & -4.8077 & 11.0577 & -3.8462 \\ 0 & 0.7692 & -1.1538 & 0.3923 & 0 & -3.8462 & 5.7692 \end{bmatrix} \end{matrix} \quad (4.1)$$

It can be explained, from what is shown in equation (4.1) that dimension will be  $7 \times 7$  when  $n = 5$  nodes.  $2n = 10$  for P, Q injections. Taking into account that the slack bus is  $2(n-1)$ , and then adding the PV node is equal to  $2(n-1) - 1$ , which leads to a number of variables equal to 7. The Matlab results after first iteration are shown in table (4.1):

Table 4.1: First iteration Matlab results



	1
1	-4.9100
2	-6.9100
3	-7.1900
4	-3.0900
5	0.9864
6	0.9817
7	0.9913

where the solutions of required parameters are shown in (4.2) – (4.3):

$$\begin{bmatrix} \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \end{bmatrix} = \begin{bmatrix} -4.91 \\ -6.95 \\ -7.19 \\ -3.09 \end{bmatrix} \quad (4.2)$$

$$\begin{bmatrix} V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0.9864 \\ 0.9817 \\ 0.9913 \end{bmatrix} \quad (4.3)$$

The results obtains after seven iterations are shown in table (4.2):

Table 4.2: Seventh iteration Matlab results

result	
7x1 double	
	1
1	-5.0124
2	-7.1322
3	-7.3705
4	-3.2014
5	0.9826
6	0.9777
7	0.9876

Table 4.2 results can be detailed as shown in (4.4) – (4.5):

$$\begin{bmatrix} \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \end{bmatrix} = \begin{bmatrix} -5.0124 \\ -7.1322 \\ -7.3705 \\ -3.2014 \end{bmatrix} \quad (4.4)$$

$$\begin{bmatrix} V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0.9826 \\ 0.9777 \\ 0.9876 \end{bmatrix} \quad (4.5)$$

The conventional method has to take the square voltage measurements in order to be directly comparable with the proposed method. It was conducted a simulation of the 5-node radial network from figure (3.1), using measurements with Gaussian noise in order to compare the PSE and the CSE, as well as to compare the two techniques from the PSE. Also, in order to compare by size it will be implemented as well with the 69 and 85 radial nodes, from figures (3.2) and (3.3), respectively.

The results detailed in table (4.3) include:

Measured value: It is the value corresponding to the measurement with Gaussian noise, and using a maximum weight of  $10^4$ .

CSE: It is the estimated value using the Conventional State Estimator.

PSE: It is the estimated value using the Proposed State Estimator, with equality restrictions in one column, and by considering virtual measurements with weights of  $10^6$ ,  $10^8$  and  $10^{10}$  respectively.

Table 4.3: Estimation results for a 5-node network

Results - 5 node network						
Measurement	Measured values	CSE	PSE Equality Restriction	PSE Weight $10^6$	PSE Weight $10^8$	PSE Weight $10^{10}$
$V_1$	1.0355	1.0347	1.0344	1.0345	1.0346	1.0347
$V_2$	1.0102	1.0189	1.0195	1.0186	1.0187	1.0189
$V_3$	1.001	0.9984	0.9973	0.9983	0.9982	0.9984
$V_4$	1.0031	1.0011	1.0023	1.0006	1.0008	1.0011
$V_5$	1.0147	1.0114	1.0136	1.0109	1.0111	1.0113
$P_{12}$	0.4424	0.4324	0.4312	0.4345	0.4335	0.4325
$P_{13}$	0.2601	0.2639	0.2643	0.2629	0.2636	0.2639
$P_{24}$	0.1626	0.1535	0.1541	0.1542	0.1538	0.1535
$P_{25}$	0.0633	0.0716	0.0711	0.0723	0.072	0.0717
$Q_{12}$	0.1383	0.1395	0.1383	0.1394	0.1397	0.1395
$Q_{13}$	0.113	0.1109	0.1116	0.1116	0.1108	0.1109
$Q_{24}$	0.0463	0.0541	0.0535	0.0537	0.0543	0.0542
$Q_{25}$	0.0163	0.0212	0.0223	0.022	0.0214	0.0212
$I_{12}^2$	0.1709	0.1928	0.1921	0.1724	0.181	0.192
$I_{13}^2$	0.1011	0.0765	0.0772	0.09	0.0768	0.0764
$I_{24}^2$	0.0098	0.0255	0.0248	0.0154	0.0253	0.0255
$I_{25}^2$	0.0163	0.0054	0.0059	0.0113	0.0058	0.0054
$P_1$	0.685	0.6964	0.6969	0.6973	0.697	0.6964
$P_2$	-0.209	-0.2035	-0.2029	-0.2047	-0.2042	-0.2036
$P_3$	-0.2711	-0.2598	-0.2621	-0.258	-0.2594	-0.2598
$P_4$	-0.1498	-0.152	-0.1536	-0.1533	-0.1524	-0.1521
$P_5$	-0.0851	-0.0713	-0.0721	-0.0716	-0.0716	-0.0714
$Q_1$	0.2592	0.2504	0.2521	0.251	0.2505	0.2504
$Q_2$	-0.0462	-0.0528	-0.0536	-0.0535	-0.0532	-0.0528
$Q_3$	-0.0814	-0.0938	-0.0951	-0.0916	-0.0937	-0.0938
$Q_4$	-0.0518	-0.0496	-0.051	-0.051	-0.0499	-0.0497
$Q_5$	-0.0191	-0.0203	-0.0231	-0.0201	-0.0204	-0.0203

Table (4.3) shows the results of the estimation after developing the simulation for the 5-node network, it can be noted both methods converge to the same estimated value, as can be seen between the PSE Equality restriction and the PSE Weight  $10^6$ . There is a first difference between both methods, which is that the CSE needs a measurement of angle reference, ( $\theta_i = 0$ ), whereas for the PSE it is not needed that measurement, as it uses branch state measurements.

In table (4.3) it can be checked the problem of bad conditioning is solved when using the second technique of resolution. It can be compared in the last three columns the results of the simulation in the radial network of 5 nodes as it is being considered the virtual measurements with weights very high with respect to the maximum weight of the ordinary measurements ( $10^4$ ), the results are more precise as the weight of the virtual measurement is increased. This leads to a bad conditioning of the gain matrix. From tables (4.4) – (4.6) it can be calculated the errors with respect to the conventional state:

5-node network shown in table (4.4):

Table 4.4: Errors CSE vs PSE 5-node network

Errors compared to Conventional State Estimator			
5-node network			
PSE Equality Restriction	PSE Weight $10^6$	PSE Weight $10^8$	PSE Weight $10^{10}$
1.73%	7.23%	0.77%	0.05%

69-node network shown in table (4.5):

Table 4.5: Errors CSE vs PSE 69-node network

Errors compared to Conventional State Estimator			
69-node network			
PSE Equality Restriction	PSE Weight $10^6$	PSE Weight $10^8$	PSE Weight $10^{10}$
1.79%	9.21%	1.23%	0.21%

85-node network shown in table (4.6):

Table 4.6: Errors CSE vs PSE 85-node network

Errors compared to Conventional State Estimator			
85-node network			
PSE Equality Restriction	PSE Weight $10^6$	PSE Weight $10^8$	PSE Weight $10^{10}$
1.89%	10.24%	1.66%	0.38%

To be noted is the importance of giving a high weight to virtual measurements in order to have a more accurate model with small errors and values within a 0.5% error compared to the conventional state estimation. In order to obtain statistical results and evaluate, if the proposed model, on top of allowing a different set of measurements does also have a computational advantage, the necessary of number of iterations of both methods will be compared as shown in table (4.7):

Table 4.7: Average number of iterations between methods

	5-node network		69-node network		85-node network	
	CSE	PSE	CSE	PSE	CSE	PSE
Exact measurements	3	<b>2</b>	3	<b>2</b>	3	<b>2</b>
Measurements with noise (3% distortion)	3	3	3	3	4	<b>3</b>

A Gaussian noise is added in table (4.7) in order to compare with exact measurements.

The developed method that takes into account the line current measurements does indeed have faster convergence, leading to a computational advantage. It has also been observed in the simulations that the scaling problem of the PSE can be decreased by varying the value of the base power in the radial network of study, this allows to obtain network measurements values that are more adequate and that improve the conditioning of the matrix that is used in the solution of the state estimation problem. To check this problem, it has been performed several simulations over the radial network of 85 nodes with measurements of Gaussian noise and in table (4.7), it is shown how the number of iterations vary to arrive to the final estimated value if we vary the base power. To show the convergence speed of both methods, it has been taken the radial network of 85 nodes with Gaussian node measurements and compared the average results with the maximum residue in each iteration over the set of measurements done with the PSE and the CSE. In tables (4.5) - (4.7), it is shown the maximum average residues in each iteration for the radial network of 85 nodes until arriving to the final estimated value, using a convergence limit of  $10^{-3}$ , the proposed method converges faster and it is done in less iterations than in the conventional estimator. The number of iterations was also compared with regards to different base powers in tables (4.8) – (4.10):

Table 4.8: Number of iterations vs base power – 5 node network

5 node network	
Base Power (MVA)	Iterations
0.1	2
0.5	2
1	2

Table 4.9: Number of iterations vs base power – 69 node network

69 node network	
Base Power (MVA)	Iterations
0.1	3
0.5	3
1	2

Table 4.10: Number of iterations vs base power – 85 node network

85 node network	
Base Power (MVA)	Iterations
0.1	4
0.5	3
1	2

Simulations of the 85-node network were conducted using exact measurements of voltage, active power flows and line current magnitudes and varying the convergence limit it can be noted that the proposed method always converges to the exact value in two iterations, whereas in the conventional method it happens when the convergence limit is higher than  $10^{-4}$ , and in the double of iterations than the proposed method. When it uses the convergence limit of  $10^{-3}$  it converges to a solution that is not the exact value.

The remaining residues after each iteration of the method was also compared between methods, to determine the extent to which the new method improves the convergence in finding a solution, as shown in tables (4.11) – (4.13).

Table 4.11: Maximum residues per iteration – 5 node network

	Maximum Residues Error margin (order of magnitude)	
	5 node network	
Iteration	CSE	PSE
1	$10^1$	$10^1$
2	$10^0$	$10^{-1}$
3	$10^{-2}$	$10^{-2}$

Table 4.12: Maximum residues per iteration – 69 node network

	Maximum Residues Error margin (order of magnitude)	
	69 node network	
Iteration	CSE	PSE
1	$10^1$	$10^1$
2	$10^0$	$10^{-1}$
3	$10^{-2}$	$10^{-2}$

Table 4.13: Maximum residues per iteration – 85 node network

	Maximum Residues Error margin (order of magnitude)	
	85 node network	
Iteration	CSE	PSE
1	$10^1$	$10^1$
2	$10^0$	$10^{-1}$
3	$10^{-1}$	$10^{-2}$
4	$10^{-2}$	$10^{-2}$

Finally, it will be compared how different convergence values affect the needed number of iterations between both methods, as shown in tables (4.14) – (4.16). It is shown how the proposed estimator performs equally well for stricter convergence values, whereas in the case of conventional estimator the number of iterations needed is augmented.

Table 4.14: Iterations per convergence values and method – 5 node network

Error margin		Convergence values 5-node network		
		$10^{-3}$	$10^{-4}$	$10^{-5}$
Iterations	CSE	3	3	4
	PSE	2	2	2

Table 4.15: Iterations per convergence values and method – 69 node network

Error margin		Convergence values 69-node network		
		$10^{-3}$	$10^{-4}$	$10^{-5}$
Iterations	CSE	3	4	4
	PSE	2	2	2

Table 4.16: Iterations per convergence values and method – 85 node network

Error margin		Convergence values 85-node network		
		$10^{-3}$	$10^{-4}$	$10^{-5}$
Iterations	CSE	3	4	4
	PSE	2	2	2

#### 4.2 PowerWorld simulation results

As detailed in chapter 3, a Matlab program was coded that implemented the conventional state estimation and the proposed state estimation with different network sizes and weights for virtual measurements. In order to verify that the state estimation follows a Newton Raphson problem solution, and is comparable with the conventional power flow study, the initial solution, (iteration number 1) will be compared, by means of the PowerWorld software.

The sample system from figure (4.1) will be simulated:

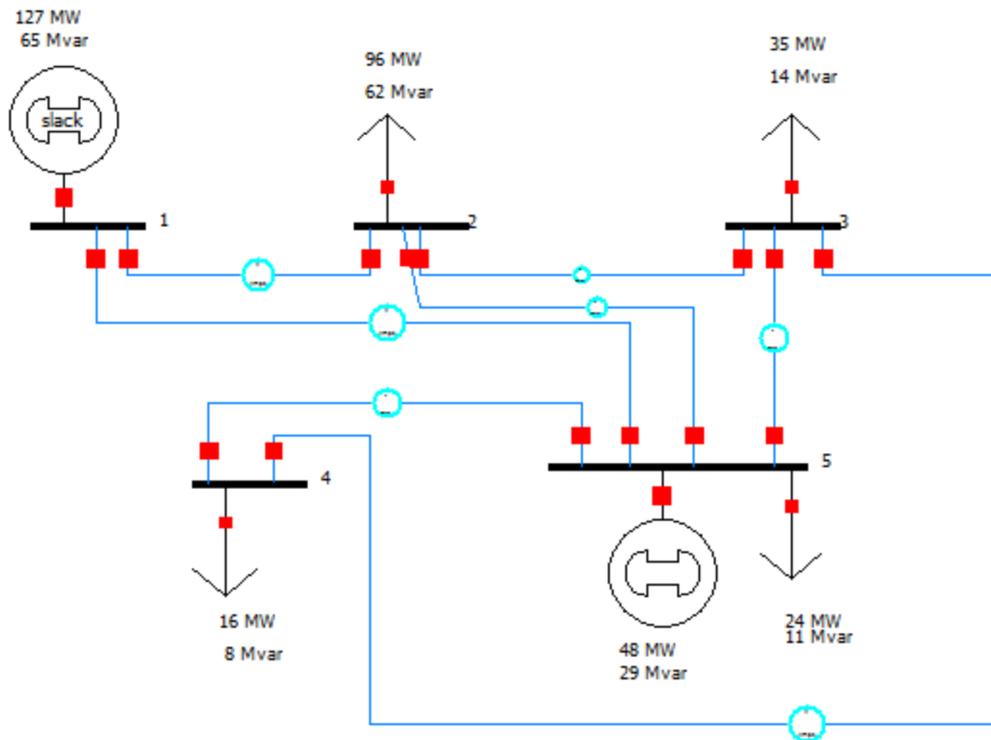


Figure 4.1: PowerWorld NR vs SE system – 5 node system

For means of replicating the results, the input data and characteristic parameters will be consistent in both Matlab and PowerWorld, and can be seen in tables (4.17) – (4.19):

Table 4.17: Transmission Line Data NR vs SE – 5 node network

Bus no.	Impedance	Line Charging
1-2	$0.02+j0.10$	$j0.03$
1-5	$0.05+j0.25$	$j0.02$
2-3	$0.04+j0.20$	$j0.025$
2-5	$0.05+j0.25$	$j0.02$
3-4	$0.05+j0.25$	$j0.02$
3-5	$0.08+j0.40$	$j0.01$
4-5	$0.10+j0.50$	$j0.075$

Table 4.18: Injected Power and Loads NR vs SE – 5 node network

Bus No.	Generation MW	Generation MVar	Load MW	Load MVar
1	-	-	0	0
2	0	0	96	62
3	0	0	35	14
4	0	0	16	8
5	48	-	24	11

The graphical simulation after PowerWorld can be found on figure (4.2):

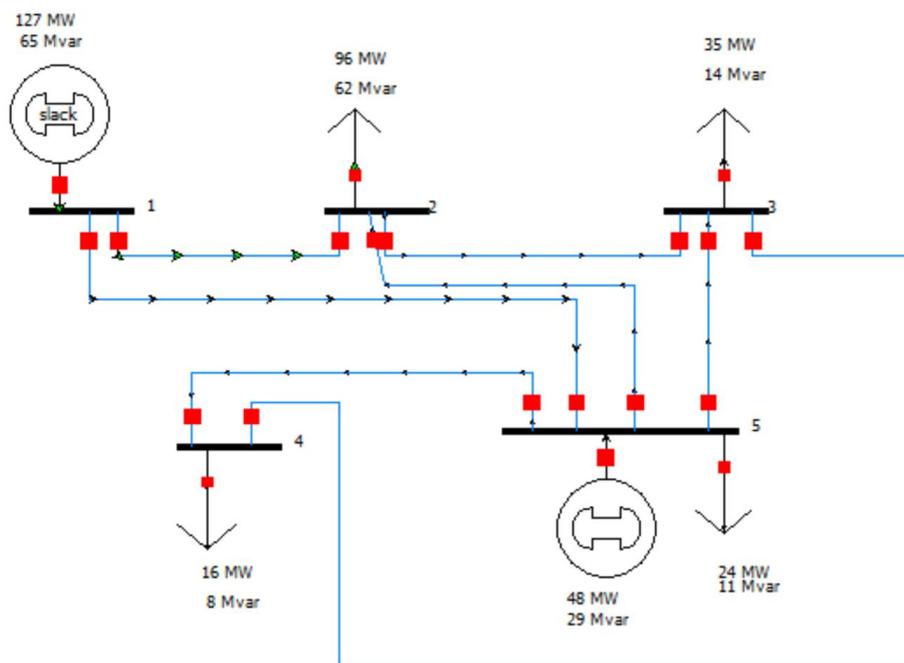


Figure 4.2: PowerWorld simulation graphic for a 5 node network

Table 4.19: PowerWorld results – 5 node network

	Number	Name	Area Name	Nom kV	PU Volt	Volt (kV)	Angle (Deg)	Load MW	Load Mvar	Gen MW	Gen Mvar
1	1	1	1	230,00	1,05000	241,500	0,00			126,78	64,65
2	2	2	1	230,00	0,97800	224,940	-4,99	96,00	62,00		
3	3	3	1	230,00	0,96808	222,659	-7,08	35,00	14,00		
4	4	4	1	230,00	0,97306	223,804	-7,27	16,00	8,00		
5	5	5	1	230,00	1,02000	234,601	-3,21	24,00	11,00	48,00	29,34

### 4.3 Comparison between Matlab and PowerWorld methods

The comparison between the Matlab and the PowerWorld results can be found in tables (4.20)

– (4.21).

Table 4.20: Comparison of results NR – Matlab and PowerWorld -Voltage

	PowerWorld (V p.u.)	Matlab Method (V p.u)	Error (%)
$V_2$	0.97801	0.9826	<b>0.47</b>
$V_3$	0.96808	0.9777	<b>0.99</b>
$V_4$	0.97306	0.9876	<b>1.49</b>

Table 4.21: Comparison of results NR – Matlab and PowerWorld -Phase

	PowerWorld (°)	Matlab Method (°)	Error (%)
$\delta_2$	-4.99	-5.01	<b>0.45</b>
$\delta_3$	-7.08	-7.13	<b>0.74</b>
$\delta_4$	-7.27	-7.37	<b>1.38</b>
$\delta_5$	-3.21	-3.20	<b>0.27</b>

Comparing the solutions of voltage and delta using Matlab and PowerWorld, it shows that they vary with less than a 2% error. This minor difference is due to a slightly modified version of the Newton Raphson method used by the PowerWorld software. It is also affected the different number of iterations required in both methods. For the comparison studies, real measurements and measurements with Gaussian noise (approximate 2% variation) were used. This measurement with approximate values was enforced in order to simulate a more realistic and non-ideal network.

## Chapter 5: Discussions and Conclusions

This chapter presents the major contributions of this work and summarizes this work with discussions on the findings. It also recommends some future work on this topic. The major contributions of this work are: (i) developed a state estimator for a determined radial network, (ii) introduced state variables of the developed method, (iii) compared them with previously published work, (iv) determined the influence of estimating parameters instead of using measured values, and (v) verified the validity of developed model using PowerWorld simulation software.

This work proposed a method with new state variables that allowed to incorporate a linear model for the line current measurements of distribution networks. Additionally, the proposed method allowed to incorporate the measurements to a set of quadratic restriction. The proposed method integrated in the model voltage magnitude measurements, and line current flows. These line current flows are usually available measurements in the distribution networks, and that was the reason that the line current variable was chosen for the developed method. Recurring to an appropriate initialization of the iterative process of the numerical solution, the proposed model showed a faster convergence with respect to the conventional method as shown in Tables (4.14) – (4.16). For large and bad conditioned radial networks, it was taken into account the selection of the base power, in order to avoid problems from bad scaling. An additional solution could have been using a new formulation that depended only on the base voltage. A problem that was considered and that must be observed is that the presence of measures of magnitude of line current measurements leads to multiple solutions of the system. This multiplicity would give as a result that the solution of the system is not unique and that would cause a problem that is out of the scope of this work.

As a continuation of the work done, in future, it could be conducted a detailed analysis of the nature of the line current measurements that were the purpose of this work. Determining

which order of magnitude and sign is appropriate for each step of the iteration method that would improve the detection of bad measures. Also, estimating the acceptable values would improve the detection of bad measures and help avoid the problem of the multiplicity of solutions. It is also suggested to formulate a modification of the proposed model, where all state variables are branch variables, which would also improve the conditioning of the gain matrix.

## Appendix: Matlab code

Detailed Matlab .m code for state estimation under new variables, including data input as well as convergence limit is presented in this Appendix.

```
%Hugo Vicente Barrera
%Master Thesis GVSU 2016

% WLS Method

clc %clearing previous window
clear %clearing previous existing variables

n = input('Insert total number of nodes in the network\n'); % input total
number of nodes
slack = input('Slack node is node number...'); % input total number of
nodes
%no error checking

P(1:n)=0; %creating and initializing vector for real injected power
Q(1:n)=0; %creating and initializing vector for imaginary injected power
sig_r(1:n)=0; %creating and initializing vector for weight of real injected
power
sig_y(1:n)=0; %creating and initializing vector for weight of imaginary
injected power

p(1:n,1:n)=0; %creating and initializing vector for real flowing power
between nodes
q(1:n,1:n)=0; %creating and initializing vector for imaginary flowing power
between nodes
sig_p(1:n,1:n)=0; %creating and initializing vector for weight of real
flowing power between nodes
sig_q(1:n,1:n)=0; %creating and initializing vector for weight of imaginary
flowing power between nodes

Vm(1:n)=1; %creating and initializing vector for given measured voltage
V(1:n)=1; %creating and initializing vector for WLS voltage

sig_v(1:n)=0; %creating and initializing vector for weight of given voltage

for k=1:n

fprintf('Is the voltage on node %d given? \n',k); %can be changed to input
only instead of looping through all nodes
aux = input('1/0 \n'); %auxiliary variable
if aux~= 0
    fprintf('Insert the voltage in p.u. in node %d \n',k);
        Vm(k)= input(' \n');

        fprintf('Insert the variance in p.u. associated to voltage
measured in node %d \n',k);
        sig_v(k)= input(' \n');
    end
end
```

```

fprintf('Is there an injected power on node %d ? \n',k); %can be changed to
input only instead of looping through all nodes
aux = input('1/0 \n'); %auxiliary variable
if aux~= 0
    fprintf('Insert the real injected power in p.u. in node %d \n',k);
        P(k)= input(' \n');
    fprintf('Insert the variance for the measurement of real injected power
in node %d \n',k);
        sig_r(k)= input(' \n');

    fprintf('Insert the imaginary injected power in p.u. in node %d
\n',k);
        Q(k)= input(' \n');
    fprintf('Insert the variance for the measurement of real injected power
in node %d \n',k);
        sig_y(k)= input(' \n');
    end
end

th(1:n)=1; %creating and initializing vector with angle values, the slack
is always one

x(1:(2*n-1))=0; %initializing state vector

int(1:n,1:n)=0; % matrix for storing if there is an interaction between
nodes
z(1:n,1:n)=0; % reserving memory for the impedances between nodes
y(1:n,1:n)=0; % reserving memory for the admittance between nodes
y_bus(1:n,1:n)=0; % reserving memory for the Y_bus matrix

for ii=1:n
    for j=1:n
        if ii<=j %for not inputting twice the connections (1 - 2 is the
same as 2 - 1) only upper diagonal
            if ii~=j % to determine if there is a relationship between two
nodes
                fprintf('Is there a connection between nodes %d and %d ?
\n',ii,j);
                int(ii,j) = input('1/0 \n');
                if int(ii,j)~= 0
                    fprintf('Insert the real impedance (r) in p.u. between
nodes %d and %d ? \n',ii,j);
                    aux_r= input(' \n');
                    fprintf('Insert the imaginary impedance (x) in p.u. between
nodes %d and %d ? \n',ii,j);
                    aux_i= input(' \n');
                    z(ii,j)=aux_r+1i*aux_i; %composing the complex number
                    y(ii,j)= 1/z(ii,j); %inverting the impedance for the
admittance

                    %completing the lower diagonal
                    z(j,ii)=z(ii,j);
                    y(j,ii)=1/z(j,ii);

                    fprintf('Is there a power flow between nodes %d and %d ?
\n',ii,j);
                    aux2 = input('1/0 \n');
                    if aux2~= 0

```

```

        fprintf('Insert the real power flow in p.u. between
nodes %d and %d \n',ii,j);
        p(ii,j)= input(' \n');
        fprintf('Insert the variance for the real power flow
between nodes %d and %d \n',ii,j);
        sig_p(ii,j)= input(' \n');

        fprintf('Insert the imaginary power flow in p.u.
between nodes %d and %d \n',ii,j);
        q(ii,j)= input(' \n');
        fprintf('Insert the variance for the imaginary power
flow between nodes %d and %d \n',ii,j);
        sig_q(ii,j)= input(' \n');

        end

    end

end

end

end

end

%g (reactance matrix)
g=real(y);
%b (susceptance matrix)
b=imag(y);

for ii=1:n %calculating the Y bus matrix (admittance between nodes
for j=1:n
    aux=0; %initializing an auxiliary variable for the addition
    if ii==j %if we are in the diagonal
        for k=1:n
            aux=aux+y(ii,k); %then add all the values
        end
        y_bus(ii,j)=aux; %passing the auxiliary value
    end
    if ii~=j
        y_bus(ii,j)=-y(ii,j); %assigning the non-diagonal value
    end
end
end

%calculate Jacobian terms

%calculate the number of given measurements, it will be the number of rows
%in the jacobian
aux1=0; %counter for voltage nodes

aux2=0; %counter for active injected powers
aux3=0; %counter for reactive injected powers

aux4=0; %counter for active flux powers
aux5=0; %counter for reactive flux powers

for i=1:n

    if Vm(i)~=1 %if the voltage in the node is given
        aux1=aux1+1; %increment the counter
    end
end

```

```

end

if P(i)~=0 %if the injected real power in the node is given
aux2=aux2+1; %increment the counter
end

if Q(i)~=0 %if the injected real power in the node is given
aux3=aux3+1; %increment the counter
end
end

%after it has been counted

for ii=1:n
for j=1:n

if p(ii,j)~=0
aux4=aux4+1; %increment the counter
end

if q(ii,j)~=0
aux5=aux5+1; %increment the counter
end

end
end

AUX=aux1+aux2+aux3+aux4+aux5;
H(1:AUX,1:(2*n-1))=0; %initializing the jacobian matrix with the
appropriate size

%all derivatives with respect to angles except for the slack and for all
%the voltages from all the given values that we have been gathering

H(1:AUX,1:(2*n-1))=0;

%all derivatives with respect to angles except for the slack and for all
%the voltages
%from all the given values that we have been gathering

intjac1(aux4,2)=0; %indexes for iteration in active power
index1=1;
for i=1:n
for j=1:n

if p(i,j)~=0
intjac1(index1,1)=i;
intjac1(index1,2)=j;
index1=index1+1;
end

end
end

for k=1:n %for all the angles except the slack
if k~=slack %if we are not in the slack angle
for l=1:aux4 %for as many as flow powers there exist
ii=intjac1(l,1);

```

```

        jj=intjac1(1,2);
        %p(ii,jj)~=0 is fulfilled

        if ii==k
            jac=(V(ii)*V(jj))*(g(ii,jj)*sin(th(ii)-th(jj))-
b(ii,jj)*cos(th(ii)-th(jj)));

            if k<slack %for placing properly in the matrix
                H(1,k)=jac;
            end

            if k>slack && slack~=1
                H(1,k-slack+1)=jac;
            end

            if k>slack && slack==1
                H(1,k-slack)=jac;
            end

        end

        if jj==k
            jac=-(V(ii)*V(jj))*(g(ii,jj)*sin(th(ii)-th(jj))-
b(ii,jj)*cos(th(ii)-th(jj)));

            if k<slack %for placing properly in the matrix
                H(1,k)=jac;
            end

            if k>slack && slack~=1
                H(1,k-slack+1)=jac;
            end

            if k>slack && slack==1
                H(1,k-slack)=jac;
            end

        end

    end

end

l=l+1; %updating the index

intjac2(aux2)=0;
index2=1;
for i=1:n
    if P(i)~=0
        intjac2(index2)=i;
        index2=index2+1;
    end
end

for k=1:n %for all the angles except the slack
    if k~=slack %if we are not in the slack angle
        for m=1:(1+aux2-1) %for as many as flow powers there exist, minus 1
            because the advancement of index was already done

```

```

ii=intjac2(m-1+1); %accessing the index

if ii==k
    aux=0;
    for j=1:n
        aux=aux+(V(ii)*V(j)*(-g(ii,j)*sin(th(ii)-
th(j))))+b(ii,j)*cos(th(ii)-th(j))-V(ii)*V(ii)*b(ii,ii);
    end
    jac=aux;

    if k<slack %for placing properly in the matrix
        H(m,k)=jac;
    end

    if k>slack && slack~=1
        H(m,k-slack+1)=jac;
    end

    if k>slack && slack==1
        H(m,k-slack)=jac;
    end

else
    jac=(V(ii)*V(k)*(g(ii,k)*sin(th(ii)-th(k))))-
b(ii,k)*cos(th(ii)-th(k));

    if k<slack %for placing properly in the matrix
        H(m,k)=jac;
    end

    if k>slack && slack~=1
        H(m,k-slack+1)=jac;
    end

    if k>slack && slack==1
        H(m,k-slack)=jac;
    end

end

end

end

end

l=m+1; %updating index for rows

intjac3(aux5,2)=0;
index1=1;
for i=1:n
    for j=1:n

        if p(i,j)~=0
            intjac3(index1,1)=i;
            intjac3(index1,2)=j;
            index1=index1+1;
        end
    end
end

```

```

end
end

for k=1:n %for all the angles except the slack
    if k~=slack %if we are not in the slack angle
        for m=1:(1+aux5-1) %for as many as flow powers there exist
            ii=intjac3(m-1+1,1);
            jj=intjac3(m-1+1,2);
            %q(ii,jj)~=0 is fulfilled

                if ii==k
                    jac=-(V(ii)*V(jj))*(g(ii,jj)*cos(th(ii)-
th(jj))+b(ii,jj)*sin(th(ii)-th(jj)));

                    if k<slack %for placing properly in the matrix
                        H(m,k)=jac;
                    end

                    if k>slack && slack~=1
                        H(m,k-slack+1)=jac;
                    end

                    if k>slack && slack==1
                        H(m,k-slack)=jac;
                    end

                end

                if jj==k
                    jac=(V(ii)*V(jj))*(g(ii,jj)*cos(th(ii)-
th(jj))+b(ii,jj)*sin(th(ii)-th(jj)));

                    if k<slack %for placing properly in the matrix
                        H(m,k)=jac;
                    end

                    if k>slack && slack~=1
                        H(m,k-slack+1)=jac;
                    end

                    if k>slack && slack==1
                        H(m,k-slack)=jac;
                    end

                end

            end

        end

    end

end

l=m+1; %updating the index

intjac4(aux3)=0;
index2=1;
for i=1:n
    if Q(i)~=0

```

```

        intjac4(index2)=i;
        index2=index2+1;
    end
end

for k=1:n %for all the angles except the slack
    if k~=slack %if we are not in the slack angle
        for m=1:(l+aux3-1) %for as many as flow powers there exist, minus 1
            because the advancement of index was already done

                ii=intjac4(m-1+1); %accessing the index

                if ii==k
                    aux=0;
                    for j=1:n
                        aux=aux+(V(ii)*V(j)*(g(ii,j)*cos(th(ii)-
th(j))))+b(ii,j)*sin(th(ii)-th(j))-V(ii)*V(ii)*g(ii,ii);
                    end
                    jac=aux;

                    if k<slack %for placing properly in the matrix
                        H(m,k)=jac;
                    end

                    if k>slack && slack~=1
                        H(m,k-slack+1)=jac;
                    end

                    if k>slack && slack==1
                        H(m,k-slack)=jac;
                    end

                else
                    jac=(V(ii)*V(j)*(-g(ii,j)*cos(th(ii)-th(j))))-
b(ii,j)*sin(th(ii)-th(j));

                    if k<slack %for placing properly in the matrix
                        H(m,k)=jac;
                    end

                    if k>slack && slack~=1
                        H(m,k-slack+1)=jac;
                    end

                    if k>slack && slack==1
                        H(m,k-slack)=jac;
                    end

                end

            end
        end
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

l=1; %reinitializing indexes
m=1;

```

```

for k=1:n %for all the voltages

    for l=1:aux4 %for as many as flow powers there exist
        ii=intjac1(l,1);
        jj=intjac1(l,2);
        %p(ii,jj)~=0 is fulfilled

        if ii==k
            jac=-V(jj)*(g(ii,jj)*cos(th(ii)-
th(jj))+b(ii,jj)*cos(th(ii)-th(jj)))+2*(g(ii,jj))*V(ii);

            H(l,n-1+k)=jac;

            end

            if jj==k
                jac=-V(ii)*(g(ii,jj)*cos(th(ii)-
th(jj))+b(ii,jj)*sin(th(ii)-th(jj)));

                H(l,n-1+k)=jac;

                end
            end
        end

        l=l+1;

for k=1:n %for all the voltages
    for m=1:(1+aux2-1) %for as many as flow powers there exist, minus 1
because the advancement of index was already done

        ii=intjac2(m-1+1); %accessing the index

        if ii==k
            aux=0;
            for j=1:n
                aux=aux+(V(j)*(g(ii,j)*cos(th(ii)-
th(j))))+b(ii,j)*sin(th(ii)-th(j))+V(ii)*g(ii,ii);
            end
            jac=aux;

            H(m,n-1+k)=jac;

        else
            jac=(V(ii)*(g(ii,k)*cos(th(ii)-th(k))))+b(ii,k)*sin(th(ii)-
th(k));

            H(m,n-1+k)=jac;

            end
        end
    end
end
end

```

```

l=m+1; %updating index for rows

for k=1:n %for all the angles except the slack
    for m=1:(1+aux5-1) %for as many as flow powers there exist
        ii=intjac3(m-1+1,1);
        jj=intjac3(m-1+1,2);
        %q(ii,jj)~=0 is fulfilled

                if ii==k
                    jac=-(V(jj))*(g(ii,jj)*sin(th(ii)-th(jj))-
b(ii,jj)*cos(th(ii)-th(jj)))-2*V(ii)*b(ii,jj);

                    H(m,n-1+k)=jac;

                end

                if jj==k
                    jac=-(V(ii))*(g(ii,jj)*sin(th(ii)-th(jj))-
b(ii,jj)*cos(th(ii)-th(jj)));

                    H(m,n-1+k)=jac;

                end

            end

        end

l=m+1; %updating the index
for k=1:n %for all the voltages
    for m=1:(1+aux3-1) %for as many as flow powers there exist, minus 1
        because the advancement of index was already done

            ii=intjac4(m-1+1); %accessing the index

            if ii==k
                aux=0;
                for j=1:n
                    aux=aux+(V(j)*(g(ii,j)*sin(th(ii)-th(j)))-
b(ii,j)*cos(th(ii)-th(j)))-V(ii)*b(ii,ii);
                end
                jac=aux;

                H(m,n-1+k)=jac;
            else
                jac=(V(ii)*(g(ii,k)*sin(th(ii)-th(k)))-b(ii,k)*cos(th(ii)-
th(k)));

                H(m,n-1+k)=jac;
            end

        end

    end

end

```

```

l=m+1; %updating the index

intjac5(aux1)=0;
index=1;
for i=1:n
    if Vm(i)~=0
        intjac5(index)=i;
        index=index+1;
    end
end

for m=1:(l+aux1-1)

for k=1:n %for all the voltages
    ii=intjac5(m-1+1);

    if ii==k
        H(m,n-1+k)=1;
    end
end

end

%%calculating matrix R

R(AUX,AUX)=0; %allocating space for matrix
sigma(AUX)=0; %allocating vector for sigmas
index=1;

for i=1:n %storing values for active power flows
    for j=1:n
        if p(i,j)~=0
            sigma(index)=sig_p(i,j);
            index=index+1;
        end
    end
end

for i=1:n %storing values for injected active power
    if P(i)~=0
        sigma(index)=sig_r(i);
        index=index+1;
    end
end

for i=1:n %storing values for reactive power flows
    for j=1:n
        if q(i,j)~=0
            sigma(index)=sig_q(i,j);
            index=index+1;
        end
    end
end

for i=1:n %storing values for injected reactive power

```

```

        if Q(i)~=0
            sigma(index)=sig_y(i);
            index=index+1;
        end
    end

for i=1:n %storing values for measured voltages
    if Vm(i)~=0
        sigma(index)=sig_v(i);
        index=index+1;
    end
end

ro=(1./sigma).^2; %calculating the standard deviation due to weights

for i=1:AUX %setting only the diagonal value
    R(i,i)=ro(i);
end

for i=1:8
    for j=1:5

        if i==3 || i==6
            H(i,j)=-H(i,j);
        end
    end
end

G=H'*R*H;

%%%%%%%%%%%%%%

x(2*n-1)=0;

index_fi(n-1)=0; %it will store which index for voltage
index_v(n)=0; %it will store which index for voltage

aux=1; %auxiliary index

for i=1:n
    if i~=slack
        x(aux)=th(i);
        index_fi(aux)=i;
        aux=aux+1;
    end
end
aux2=1;
for i=n:(2*n-1)
    x(aux)=Vm(i-n+1);
    index_v(aux2)=i-n+1;
    aux=aux+1;
    aux2=aux2+1;
end

%%%%%%%%%%%%%% xcomp vector

```

```

xcomp(AUX)=0;
index3=1;

for i=1:n
    if Vm(i)~=1 %if the voltage in the node is given
        xcomp(index3)=Vm(i); %increment the counter
        index3=index3+1;
    end
end

for ii=1:n
    for j=1:n

        if p(ii,j)~=0
            xcomp(index3)=p(ii,j); %increment the counter
            index3=index3+1;
        end
    end
end

for ii=1:n
    for j=1:n

        if q(ii,j)~=0
            xcomp(index3)=q(ii,j); %increment the counter
            index3=index3+1;
        end
    end
end

for i=1:n
    if P(i)~=0 %if the voltage in the node is given
        xcomp(index3)=P(i); %increment the counter
        index3=index3+1;
    end
end

for i=1:n
    if Q(i)~=0 %if the voltage in the node is given
        xcomp(index3)=Q(i); %increment the counter
        index3=index3+1;
    end
end

%%%%%%%%%%%%%%
%%%

x0 = x; %passing the value

%%%%%%%%%%%%%%

%solution of the system

counter=0;
z=zeros(AUX,1)';

for i=1:n

```

```

    if Vm(i)~=1
        counter=counter+1; %in order to know how many voltages are known
    end
end

pl=1; %index for storing in z vector,

for i=1:aux1

    z(i)=x(n-1+i)-xcomp(i);
    pl=pl+1;
end

%Flowing active powers
for k=1:aux4 %for all the power flows

    ii=intjac1(k,1); %extracting the indexes
    jj=intjac1(k,2);

    if ii==slack
        z(pl)=x(ii+n-1)*x(ii+n-1)*g(ii,jj)-x(ii+n-1)*x(jj+n-
1)*(g(ii,jj)*cos(0-x(jj))+b(ii,jj)*sin(0-x(jj)))-xcomp(pl);
        pl=pl+1;
    end

    if jj==slack
        z(pl)=x(ii+n-1)*x(ii+n-1)*g(ii,jj)-x(ii+n-1)*x(jj+n-
1)*(g(ii,jj)*cos(x(ii))+b(ii,jj)*sin(x(ii)))-xcomp(pl);
        pl=pl+1;
    end

    if ii~=slack && jj~=slack
        z(pl)=x(ii+n-1)*x(ii+n-1)*g(ii,jj)-x(ii+n-1)*x(jj+n-
1)*(g(ii,jj)*cos(x(ii)-x(jj))+b(ii,jj)*sin(x(ii)-x(jj)))-xcomp(pl);
        pl=pl+1;
    end

end

end

%Flowing reactive powers
for k=1:aux5 %for all the power flows

    ii=intjac3(k,1); %extracting the indexes
    jj=intjac3(k,2);

    if ii==slack
        z(pl)=x(ii+n-1)*x(ii+n-1)*b(ii,jj)-(x(ii+n-1)*x(jj+n-1))*(-
b(ii,jj)*cos(-x(jj)))-xcomp(pl);
        pl=pl+1;
    end

    if jj==slack
        z(pl)=x(ii+n-1)*x(ii+n-1)*b(ii,jj)-(x(ii+n-1)*x(jj+n-1))*(-
b(ii,jj)*cos(x(ii)))-xcomp(pl);
    end
end

```

```

        pl=pl+1;
    end

    if ii~=slack && jj~=slack
        z(pl)=x(ii+n-1)*x(ii+n-1)*b(ii,jj)-(x(ii+n-1)*x(jj+n-1))*(-
b(ii,jj)*cos(x(ii)-x(jj)))-xcomp(pl);
        pl=pl+1;
    end

end

%Injected active powers
for k=1:aux2

    i=intjac2(k); %extracting the index

    if i==slack

        suma=0; %auxiliary index for the addition
        for j=1:n

            if j==slack
                suma=suma+x(j+n-1)*(g(i,j)*cos(0)+b(i,j)*sin(0));
            else
                suma=suma+x(j+n-1)*(g(i,j)*cos(x(j))+b(i,j)*sin(x(j)));
            end

        end

        z(pl)=x(i+n-1)*suma-xcomp(pl);
        pl=pl+1;

    end

    if i~=slack

        suma=0; %auxiliary index for the addition
        for j=1:n

            if j==slack
                suma=suma+x(j+n-1)*(g(i,j)*cos(x(i))+b(i,j)*sin(x(i)));
            else
                suma=suma+x(j+n-1)*(g(i,j)*cos(x(i)-x(j))+b(i,j)*sin(x(i)-
x(j)));
            end

        end

        z(pl)=x(i+n-1)*suma-xcomp(pl);
        pl=pl+1;

    end

end
end

```

```
%Injected reactive powers
```

```
for k=1:aux3
```

```
    i=intjac4(k); %extracting the index
```

```
    if i==slack
```

```
        suma=0; %auxiliary index for the addition
```

```
        for j=1:n
```

```
            if j==slack
```

```
                suma=suma+x(j+n-1)*(g(i,j)*sin(0)-b(i,j)*cos(0));
```

```
            else
```

```
                suma=suma+x(j+n-1)*(g(i,j)*sin(-x(j))-b(i,j)*cos(-x(j)));
```

```
            end
```

```
        end
```

```
        z(pl)=x(i+n-1)*suma-xcomp(pl);
```

```
        pl=pl+1;
```

```
    end
```

```
    if i~=slack
```

```
        suma=0; %auxiliary index for the addition
```

```
        for j=1:n
```

```
            if j==slack
```

```
                suma=suma+x(j+n-1)*(g(i,j)*sin(x(i))-b(i,j)*cos(x(i)));
```

```
            else
```

```
                suma=suma+x(j+n-1)*(g(i,j)*sin(x(i)-x(j))-b(i,j)*cos(x(i)-x(j)));
```

```
            end
```

```
        end
```

```
        z(pl)=x(i+n-1)*suma-xcomp(pl);
```

```
        pl=pl+1;
```

```
    end
```

```
end
```

```
x2=x'+((H'*R*H)^(-1))*H'*R*z';
```

```
xmodif=x2';
```

```
it=it+1;
```

```
end
```

```
function F = state2(x,REF)
```

```

n=REF(1);
Vm=REF((2*n+1):(3*n));
AUX=REF(3*n+1);
xcomp=REF((3*n+2):(3*n+AUX+1));
aux1=REF(3*n+AUX+2);
aux2=REF(3*n+AUX+3);
aux3=REF(3*n+AUX+4);
aux4=REF(3*n+AUX+5);
aux5=REF(3*n+AUX+6);

auxintjac1=REF((3*n+AUX+7):(3*n+AUX+7+2*aux4-1));
intjac2=REF((3*n+AUX+7+2*aux4):(3*n+AUX+7+2*aux4+aux2-1));
auxintjac3=REF((3*n+AUX+7+2*aux4+aux2):(3*n+AUX+7+2*aux4+aux2+2*aux5-1));
intjac4=REF((3*n+AUX+7+2*aux4+aux2+2*aux5):(3*n+AUX+7+2*aux4+aux2+2*aux5+aux3-1));
slack=REF(3*n+AUX+7+2*aux4+aux2+2*aux5+aux3);
auxg=REF(3*n+AUX+7+2*aux4+aux2+2*aux5+aux3+1:3*n+AUX+7+2*aux4+aux2+2*aux5+aux3+n*n);
auxb=REF(3*n+AUX+7+2*aux4+aux2+2*aux5+aux3+n*n+1:3*n+AUX+7+2*aux4+aux2+2*aux5+aux3+2*n*n);
counter=0;

F=zeros((2*n-1),1)'; %initializing the return values of the function

%%%%%assigning voltage x variables%%%%%%%%%
V=zeros(n,1)';
for i=1:n
    V(i)=x(i+n-1);
end

%%%%%assigning theta x variables%%%%%%%%%
th=zeros(n,1)';

auxs=1; %index for storing index

for i=1:n
    if i~=slack
        th(i)=x(auxs);
        auxs=auxs+1;
    end
end

%%%%%%%%%

%intjac1 and intjac3 must be converted back to matrices from vectors

%intjac1
intjac1(aux4,2)=0;
aux7=1; %auxiliary variable to advance storing index
for i=1:aux4
    intjac1(i,1)=auxintjac1(aux7);
    aux7=aux7+1;
    intjac1(i,2)=auxintjac1(aux7);
    aux7=aux7+1;
end

%intjac3
intjac3(aux5,2)=0;

```

```

aux7=1; %auxiliary variable to advance storing index
for i=1:aux5
    intjac3(i,1)=auxintjac3(aux7);
    aux7=aux7+1;
    intjac3(i,2)=auxintjac3(aux7);
    aux7=aux7+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%g and b must be converted back to matrices from vectors

%g
g=zeros(n);
auxs=1;
for i=1:n
    for j=1:n
        g(i,j)=auxg(auxs);
    end
end

%b
b=zeros(n);
auxs=1;
for i=1:n
    for j=1:n
        b(i,j)=auxb(auxs);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

for i=1:n
    if Vm(i)~=1
        counter=counter+1; %in order to know how many voltages are known
    end
end

%aux1 counter for voltage nodes (n -1)
%aux2 %counter for active injected powers
%aux3 %counter for reactive injected powers

%aux4 %counter for active flux powers
%aux5 %counter for reactive flux powers

%%Voltages in the nodes
Vcalc(aux1,counter)=0;

for i=1:aux1
    for j=1:counter

        %Vcalc(i,j)=x(n-1+j)-xcomp(i);

    end
end
end

```

```

for i=1:aux1
    for j=1:counter

        if i==j
            %F(i)=Vcalc(i,j);
            end

        end

    end

end

pl=1; %index for placing the function, it will start with these
measurements

%Flowing active powers
for k=1:aux4 %for all the power flows

    ii=intjac1(k,1); %extracting the indexes
    jj=intjac1(k,2);

    if ii==slack
        if pl <= (2*n-1)
            F(pl)=x(ii+n-1)*x(ii+n-1)*g(ii,jj)-x(ii+n-1)*x(jj+n-
1)*(g(ii,jj)*cos(0-x(jj))+b(ii,jj)*sin(0-x(jj)))-xcomp(pl);
            pl=pl+1;
            end
        end

        if jj==slack
            if pl <= (2*n-1)
                F(pl)=x(ii+n-1)*x(ii+n-1)*g(ii,jj)-x(ii+n-1)*x(jj+n-
1)*(g(ii,jj)*cos(x(ii))+b(ii,jj)*sin(x(ii)))-xcomp(pl);
                pl=pl+1;
                end
            end

            if ii~=slack && jj~=slack
                if pl <= (2*n-1)
                    F(pl)=x(ii+n-1)*x(ii+n-1)*g(ii,jj)-x(ii+n-1)*x(jj+n-
1)*(g(ii,jj)*cos(x(ii)-x(jj))+b(ii,jj)*sin(x(ii)-x(jj)))-xcomp(pl);
                    pl=pl+1;
                    end
                end

            end

        end

    end

    %Flowing reactive powers
    for k=1:aux5 %for all the power flows

        ii=intjac3(k,1); %extracting the indexes
        jj=intjac3(k,2);

        if ii==slack
            if pl <= (2*n-1)
                F(pl)=x(ii+n-1)*x(ii+n-1)*b(ii,jj)-(x(ii+n-1)*x(jj+n-1))*(-
b(ii,jj)*cos(-x(jj)))-xcomp(pl);
            end
        end
    end

```

```

        pl=pl+1;
    end
end

    if jj==slack
        if pl <= (2*n-1)
            F(pl)=x(ii+n-1)*x(ii+n-1)*b(ii,jj)-(x(ii+n-1)*x(jj+n-1))*(-
b(ii,jj)*cos(x(ii)))-xcomp(pl);
            pl=pl+1;
        end
    end

    if ii~=slack && jj~=slack
        if pl <= (2*n-1)
            F(pl)=x(ii+n-1)*x(ii+n-1)*b(ii,jj)-(x(ii+n-1)*x(jj+n-1))*(-
b(ii,jj)*cos(x(ii)-x(jj)))-xcomp(pl);
            pl=pl+1;
        end
    end

end

%Injected active powers
for k=1:aux2

    i=intjac2(k); %extracting the index

    if i==slack

        suma=0; %auxiliary index for the addition
        for j=1:n

            if j==slack
                suma=suma+x(j+n-1)*(g(i,j)*cos(0)+b(i,j)*sin(0));
            else
                suma=suma+x(j+n-1)*(g(i,j)*cos(x(j))+b(i,j)*sin(x(j)));
            end

        end

        if pl <= (2*n-1)
            F(pl)=x(i+n-1)*suma-xcomp(pl);
            pl=pl+1;
        end

    end

    if i~=slack

        suma=0; %auxiliary index for the addition
        for j=1:n

            if j==slack
                suma=suma+x(j+n-1)*(g(i,j)*cos(x(i))+b(i,j)*sin(x(i)));

```

```

        else
            suma=suma+x(j+n-1)*(g(i,j)*cos(x(i)-x(j))+b(i,j)*sin(x(i)-
x(j)));
        end

    end

    if pl <= (2*n-1)
        F(pl)=x(i+n-1)*suma-xcomp(pl);
        pl=pl+1;
    end

end

end

%Injected reactive powers

for k=1:aux3

    i=intjac4(k); %extracting the index

    if i==slack

        suma=0; %auxiliary index for the addition
        for j=1:n

            if j==slack
                suma=suma+x(j+n-1)*(g(i,j)*sin(0)-b(i,j)*cos(0));
            else
                suma=suma+x(j+n-1)*(g(i,j)*sin(-x(j))-b(i,j)*cos(-x(j)));
            end

        end

        if pl <= (2*n-1)
            F(pl)=x(i+n-1)*suma-xcomp(pl);
            pl=pl+1;
        end

    end

    if i~=slack

        suma=0; %auxiliary index for the addition
        for j=1:n

            if j==slack
                suma=suma+x(j+n-1)*(g(i,j)*sin(x(i))-b(i,j)*cos(x(i)));
            else
                suma=suma+x(j+n-1)*(g(i,j)*sin(x(i)-x(j))-b(i,j)*cos(x(i)-
x(j)));

```

```
        end

    end

    if p1 <= (2*n-1)
        F(p1)=x(i+n-1)*suma-xcomp(p1);
        p1=p1+1;
    end

end

end
```

## References

- [1] Abur, A.; Exposito, A. G. "Observability and bad data identification when using ampere measurements in state estimation", *Circuits and Systems*, 1993. ISCAS '93, 1993 IEEE International Symposium on, On page(s): 2668 - 2671 vol.4
- [2] Abur, A.; Exposito A. G.; "Multiple Solutions and Unique Observability in State Estimation," *Power System Computation Conference*, Dresden, Germany, 1996, pp 1200-1206
- [3] Abur, A.; Exposito, A. G. "Detecting multiple solutions in state estimation in the presence of current magnitude measurements", *Power Systems*, IEEE Transactions on. On page(s): 370 - 375 Volume: 12, Issue: 1, Feb 1997
- [4] Ahmad, M.; *Power system state estimation*. Boston: Artech House, 2013.
- [5] Chen J.; Liao Y.; Gou B.; "Study of WLS state estimation convergence characteristics under topology errors," in *Southeastcon, 2013 Proceedings of IEEE* , vol., no., pp.1-6, 4-7 April 2013
- [6] Chen, L.; Bi, T.S.; Xue, A.C.; Yang, Q.X., "Approach for linear state estimation considering zero-injected currents," in *Renewable Power Generation Conference (RPG 2013)*, 2nd IET , vol., no., pp.1-5, 9-11 Sept. 2013

- [7] Choi S.; Meliopoulos, A.P.; Ratnesh, K., "Autonomous state estimation based diagnostic system in smart grid," in Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES , vol., no., pp.1-6, 24-27 Feb. 2013
- [8] Conejo, A.J., "Power System State Estimation-Theory and Implementations [Book Review]," in Power and Energy Magazine, IEEE, vol.3, no.2, pp.64-65, March-April 2005
- [9] Dowi, S.A.; Gengyin Li, "A new approach for including synchronized phasor measurements in dynamic state estimation," in Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on , vol., no., pp.389-394, 23-24 Dec. 2013
- [10] Exposito A. G., Abur A., De la Villa A.; Quiles C., "A multilevel state estimation paradigm for smart grids," Proceedings of the IEEE, vol. 99, pp. 952-976, 2011.
- [11] Exposito A. G., Abur A., De la Villa A.; Quiles C., "Bilinear Power System State Estimation," in Power Systems, IEEE Transactions on , vol.27, no.1, pp.493-501, Feb. 2012
- [12] Fusco, F. "General bad data identification and estimation in the presence of critical measurement sets", PES General Meeting | Conference & Exposition, 2012 IEEE
- [13] Gol, M.; Abur, A. "Observability and Criticality Analyses for Power Systems Measured by Phasor Measurements", Power Systems, IEEE Transactions on. On page(s): 3319 - 3326  
Volume: 28, Issue: 3, Aug. 2013

[14] Hassanzadeh, M.; Evrenosoglu, C. Y. "Use of PMUs in regression based power system dynamic state estimation" in North American Power Symposium (NAPS), 2012, 2012, pp. 1-5.

[15] Jaen, A.V.; Exposito, A. "Including ampere measurements in generalized state estimators", Power Systems, IEEE Transactions on, On page(s): 603 - 610 Volume: 20, Issue: 2, May 2005

[16] Krizan, P.; Magnago, F.H.; Abur, A., "A graphical tool for the analysis of network observability and state estimation algorithms" in Power Engineering Society 1999 Winter Meeting, IEEE , vol.1, no., pp.502-507 vol.1, 31 Jan-4 Feb 1999

[17] Marinho, J.L., P.A. Machado; C. Bongers, "On the Use of Line Current Measurements for Reliable State Estimation in Power Systems" Power Industry Computer Applications Conference, PICA-79, Cleveland, OH, 1979.

[18] Nguyen N. V.; Shin, V.; Shevlyakov, G. "Power system state estimation with fusion method" in Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on, vol.5, no., pp.71-76, 26-28 Feb. 2010

[19] Nuqui R. ; Phadke A.; "Hybrid linear state estimation utilizing synchronized phasor measurements" in Power Tech, 2007 IEEE Lausanne, 2007, pp. 1665-1669.

[20] Pau, M.; Pegoraro, P.A.; Sulis, S. "Branch current state estimator for distribution system based on synchronized measurements", Applied Measurements for Power Systems (AMPS), 2012 IEEE International Workshop on, On page(s): 1 - 6

[21] Shivakumar, N. R.; Jain, A. "A Review of Power System Dynamic State Estimation Techniques" in Power System Technology and IEEE Power India Conference, 2008. POWERCON 2008. Joint International Conference on, 2008, pp. 1-6.

[22] Shivakumar, N. R.; Jain, A. "Phasor measurements in dynamic state estimation of power systems" in TENCON 2008-2008 IEEE Region 10 Conference, 2008, pp. 1-6.

[23] Sijs, J.; Hanebeck, U.; Noack, B., "An empirical method to fuse partially overlapping state vectors for distributed state estimation" in Control Conference (ECC), 2013 European, vol., no., pp.1615-1620, 17-19 July 2013

[24] De la Villa, A., Exposito, A. G. "Including Ampere Measurements in Generalized State Estimators," IEEE Transactions on Power Systems, vol. 20, no. 2, pp. 603–610, 2005.

[25] Yang Xiaoli; Hu Zongshuai; Fan Rusen; Xu Haotian; Yang Heng; Wang Yong; Tian Xiuxia, "Weighted least squares state estimation based on the optimal weight," in Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2015 Third International Conference on , vol., no., pp.12-16, April 29 2015-May 1 2015

- [26] Yibing, S.; Minyue, F.; Bingchang W.; Huanshui Z., "Dynamic state estimation in power systems using a distributed MAP method," in Control Conference (CCC), 2015 34th Chinese , vol., no., pp.47-52, 28-30 July 2015
- [27] Zhou, M.; Centeno, V. A.; Thorp, J. S.; Phadke, A. G.; "An alternative for including phasor measurements in state estimators" Power Systems, IEEE Transactions on, vol. 21, pp. 1930-1937, 2006.
- [28] C. A. Fantin, M. R. C. Castillo, B. E. B. de Carvalho and J. B. A. London, "Using pseudo and virtual measurements in distribution system state estimation," Transmission & Distribution Conference and Exposition - Latin America (PES T&D-LA), 2014 IEEE PES, Medellin, 2014, pp. 1-6.
- [29] Abur, A.; Expósito, A. G.; Power system state estimation: theory and implementation. New York, NY: Marcel Dekker, 2004