IRis

Northeastern University

Electrical and Computer Engineering Master's Theses

Department of Electrical and Computer Engineering

January 01, 2008

Practical meter placement algorithms for improving state estimation performance

Bilgehan Donmez
Northeastern University

Recommended Citation

Donmez, Bilgehan, "Practical meter placement algorithms for improving state estimation performance" (2008). *Electrical and Computer Engineering Master's Theses.* Paper 20. http://hdl.handle.net/2047/d10017756

This work is available open access, hosted by Northeastern University.

NORTHEASTERN UNIVERSITY

Graduate School of Engineering

Thesis Title: Practical Meter Placement Algorithms for Improving State Estimation Performance

Author: Bilgehan Donmez

Department: Electrical and Computer Engineering

Approved for Thesis Requirements of the Master of Science Degree

_____ _____

Thesis Adviser: Professor Ali Abur Date

_____ _____

Thesis Reader: Professor Alexandar Stankovic Date

_____ _____

Thesis Reader: Professor Bahram Shafai Date

_____ _____

Department Chair: Ali Abur Date

Graduate School Notified of Acceptance:

_____ _____

Director of the Graduate School: Yaman Yener Date

PRACTICAL METER PLACEMENT ALGORITHMS FOR IMPROVING STATE ESTIMATION PERFORMANCE

A Thesis Presented

by

Bilgehan Donmez

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements

for the degree of

Master of Science

in

Electrical Engineering

in the field of

Power Systems

Northeastern University

Boston, Massachusetts

September 2008

© Copyright 2008 by Bilgehan Donmez

All Rights Reserved

Abstract

In the early 1990s, Congress and the Federal Energy Regulatory Commission (FERC), which oversees the transmission and wholesale of electricity, decided to deregulate the electric power industry to bring more competition to the market. The traditional vertically integrated structure of the utilities was broken into separate entities of generation, transmission, and distribution. These recent changes resulted in a competitive market where a reliable, secure, and cost efficient operation has become more important. In an effort to improve the performance of their energy management functions, the utilities are upgrading their measurement sets by installing new meters, such as phasor measurement units.

In this thesis, methods to improve the functionality of state estimation by observability analysis and bad data detection are discussed. Identification of unobservable branches and detection of observable islands are studied as part of the observability problem. Critical measurement elimination by placing additional meters is investigated to provide the system with bad data detection capability.

Two measurement placement algorithms are developed to upgrade the measurement configuration and to eliminate critical measurements. For each algorithm, two types of measurements, either real power injections or phasor measurement units (PMUs), can be placed as candidate measurements. The simulations of IEEE 14 and

30 bus systems illustrate that the algorithms are effective in finding an optimal set of additional measurements.

The methods developed in this thesis are not only used to install new meters, but they are also used to detect the weak spots in a power system. These algorithms work for both designing a measurement configuration for a new system or upgrading the measurement design of an existing system.

Acknowledgements

I wish to acknowledge and thank those people who contributed to this thesis:

Prof. Ali Abur, my academic advisor, who gave me the chance to do research in this field and guided me through my studies. It was an honor and pleasure to be his research assistant.

Roozbeh Emami, my research partner, for his friendship and help.

Gülçin Puşuoğlu, my aunt, Salim and Gülseren Dönmez, my parents, and Can Dönmez, my brother, for their love and support.

Contents

Abstract **iv**

Acknowledgements **vi**

1 Introduction **1**

1.1 Motivation 1

1.2 Contribution of This Thesis 3

1.3 Chapter Organization 4

2 Weighted Least Squares State Estimation **5**

2.1 Introduction 5

2.2 Transmission Lines 7

2.3 Transformers 8

2.3.1 Tap Changing Transformer 8

2.3.2 Phase Shifting Transformer 10

2.4 Admittance Matrix 12

2.5 Measurement Model 15

2.5.1 The Measurement Jacobian 17

2.6 WLS Algorithm 19

2.7 DC Approximation 22

3 Observability Analysis and Special Cases 25

3.1 Introduction 25

3.2 Observability Check 26

3.3 Finding Unobservable Branches 31

3.4 Identification of Observable Islands 35

4 Meter and PMU Placement for Observability 37

4.1 Introduction 37

4.2 Assessment of the Measurements 39

4.2.1 Measurement Jacobian 39

4.2.2 Sparse Factorization 41

4.3 Optimal Meter Placement 42

4.3.1 Identifying Candidates 42

4.3.2 Selecting Candidates 44

4.4 Numerical Example 47

4.4.1 Injection Solution 47

4.4.2 PMU Solution 52

4.5 Simulation Results 53

4.5.1 IEEE 14 Bus System 53

4.5.2 IEEE 30 Bus System 54

4.6 Conclusions 55

5 Meter and PMU Placement for Bad Data Detection 58

5.1 Introduction 58

5.2 Identification of Critical Measurements 60

5.3 Measurement Placement 64
5.3.1 Conventional Measurement Placement 64
5.3.2 PMU Placement 68
5.4 Simulation Results 73
5.4.1 IEEE 14 Bus System 73
5.4.2 IEEE 30 Bus System 75
5.5 Conclusions 76

6 Conclusions **79**

Bibliography **81**

Appendices **87**

A Measurement Placement Algorithm for Observability **87**

B Measurement Placement Algorithm for Critical Measurement Elimination **94**

C Functions Used in the Measurement Placement Algorithms **101**
C.1 Read Network Parameters and Calculate Y matrix 101
C.2 Read Measurement Data 104
C.3 Build Measurement Jacobian 107
C.4 Matrices Required for PMU Optimization 112
C.5 Display Chosen PMUs 113

List of Figures

2.1 2-port π model of a transmission line 8

2.2 Circuit of a tap changing transformer 9

2.3 Tap changing vs. phase shifting transformer 11

2.4 One-line diagram for a 3 bus system 13

2.5 2 port π model used for measurement equations 16

3.1 4 bus unobservable system example 30

3.2 6 bus unobservable system example 32

3.3 Islands in 6 bus system 35

4.1 6 bus system example 48

4.2 IEEE 14 bus system 54

4.3 IEEE 30 bus system 56

5.1 5 bus system analyzed for critical measurements 62

5.2 IEEE 14 bus system analyzed for critical measurements 74

5.3 IEEE 30 bus system analyzed for critical measurements 76

List of Tables

2.1 Network parameters of the 3 bus system 13

3.1 Summary of observable and unobservable cases 29

3.2 Unobservable branches in Figure 3.2 35

4.1 Measurements in the IEEE 30 bus system 57

5.1 Phasor measurements of all possible PMUs at bus 4 69

5.2 4 optimal candidate PMUs 72

5.3 Critical measurements in the IEEE 14 bus system 75

5.4 Optimal placement of candidate PMUs for 14 bus system 75

5.5 Critical measurements in the IEEE 30 bus system 78

5.6 Optimal placement of candidate PMUs for 30 bus system 78

Chapter 1

Introduction

1.1 Motivation

The deregulation of power markets in the United States changed the structure of the electric power industry. Before deregulation, generation, distribution, and transmission were owned by a single utility company within a certain region. The restructuring mandated that these three subsystems were to be owned and operated by private companies [1] while coordinated by an independent system operator. The functions of these three companies and the operator can be summarized as:

Generation Company (GENCO) is responsible for generating electricity, which is sold (delivered) to the customers according to the contracts.

Transmission Company (TRANSCO) is responsible for reliable transmission of the power from generator to customer. TRANSCOs are the links between generation and distribution.

Distribution Company (DISTCO) is responsible for providing power to the customers. The power generated by GENCOs are consumed by DISTCOs.

Independent System Operator (ISO) is responsible for coordination of the market within its region. The operations of GENCOs, TRANSCOs, and DISTCOs are governed by ISOs.

State estimation is one of the most important tools that TRANSCOs and ISOs use to analyze their systems. The operating conditions of a power system is monitored by state estimation. Measurement data coming from all the meters in the system are processed to find the best estimate of voltage phasors at every node in the system. The success of state estimation is directly related to the redundancy in metering. In the new structure of the power industry, companies such as TRANSCOs and ISOs need more meters in order to have a better understanding of their system behavior.

Another improvement in the system operation has come from the increased popularity of phasor measurement units (PMUs). More PMUs are being installed throughout the country. The voltage and current phasors are directly measured by PMUs, which have a positive effect on system monitoring. Thus, an optimal placement of PMUs in a system has recently become a significant task.

These recent changes in the power industry have resulted in a more competitive market. System operators are trying harder to provide a reliable and uninterrupted service by preventing blackouts, matching the generation capacity with load demand, and meeting the security constraints of the system [2]. State estimation plays a key role in the reliable operation of a system given that there are enough measurements placed strategically throughout the system.

1.2 Contribution of This Thesis

The main goal of this thesis is to improve the state estimation performance by studying and upgrading the measurement design. The first task is checking the observability of the system to determine whether or not the available measurement set is enough to make the state estimation possible [3]. The observability analysis is followed by a measurement placement procedure to bring the system into an observable condition in an optimal way.

An observable system does not ensure the detection of errors in the measurements. In addition, loss of a measurement can endanger the observability. These concerns can be addressed by placing additional meters at critical locations. Redundancy in measurements enables detection and identification of topology changes and bad data [4]. The large interconnected systems, which consist of thousands of buses, expand regularly, therefore, it is important to upgrade the measurement configuration by installing new meters in a cost efficient way.

A number of algorithms to address the questions regarding measurement design are developed for:

- testing observability
- identification of unobservable branches and detection of observable islands
- measurement placement for observability
- identification of critical measurements whose loss result in an unobservable system
- measurement placement to eliminate critical measurements

1.3 Chapter Organization

The discussion in this thesis starts with a background in state estimation given in Chapter 2, where the weighted least squares (WLS) method is explained. Chapter 2 wraps up with the linear decoupled measurement model, which is widely used in observability studies in later chapters.

Chapter 3 describes observability analysis. Algorithms for testing observability and detecting unobservable branches are explained. Identification of observable islands is illustrated by a simple example.

In Chapter 4, a practical measurement placement algorithm to restore observability with either conventional measurements or PMUs is developed. This algorithm is tested with IEEE 14 and 30 bus systems.

Chapter 5 is a discussion of another measurement placement algorithm where meters (conventional or PMU) are placed to eliminate critical measurements. The simulation results show that the IEEE 14 and 30 buses are critical measurement free with the addition of suggested meters.

Finally, in Chapter 6, the summary of the main contributions of this thesis is given. In the future work section, some possible studies related to measurement placement are discussed.

Chapter 2

Weighted Least Squares State Estimation

2.1 Introduction

State estimation is a tool system operators use to evaluate the present condition of their system. A set of measurements and network topology are processed to get the best estimate of the state. The information extracted from state estimation also helps predict future conditions of the system. The state consists of voltage phasors of all the buses in a system. A voltage phasor can be represented as:

$$\hat{V} = V\angle\theta^{\circ} = Ve^{j\theta} = V\cos\theta + jV\sin\theta \tag{2.1}$$

where V is the voltage magnitude and θ is the phase angle. The state x is usually formed as a row or column vector where all the phase angles are listed first, followed by voltage magnitudes:

$$x = [\theta_1\theta_2\cdots\theta_n V_1 V_2 \cdots V_n]^T \tag{2.2}$$

It is a common practice to choose a reference bus such that all other phasors are calculated with respect to the reference angle. If the first bus is chosen as reference, then its phase angle is set equal to zero, $\theta_1 = 0$.

Although there are time-varying variables in power systems such as frequency and load, power systems have a slow response time to changes under normal operating conditions. A dynamic state estimation can be implemented with additional information and variables, but a steady-state approximation is practical and its performance is satisfactory for most cases [5]. Therefore, state estimation is assumed to be in steady-state and it is sometimes called static-state estimator [4].

There are three main parts in a power system: generation, transmission, and distribution. The lines, where voltage levels are higher than a certain value, usually 115kV, are labeled as transmission lines. In state estimation, the transmission part of the power grid represents a network, where generation and distribution are modeled as net power injections. An injection refers to the net flow (of current or power) in or out of a bus. Positive injection means that flow goes out of the bus or provided by the bus. This kind of a bus is called generation bus. On the other hand, negative injection implies flow being drawn to the bus, thus it is a load bus.

State estimation results are processed to identify errors in the meters and parameters given that there are enough redundant measurements. In addition, failure or loss of a meter can be detected. Topology of a network changes when a line is lost due to overloading or equipment failure. In such a case, the security of the system might be in danger since some other lines might get overloaded as well. State estimation results, analyzed carefully, can give early warnings of such cases.

Once the state is estimated, all flows can be calculated using admittance matrix, derived from network parameters, and Kirchoff's laws [4]. If an operator knows the

present state of the system, then future flows can be predicted. Therefore, an accurate estimation of the current state improves the security.

This chapter is organized such that background information about power system network is given before the state estimation algorithm is explained. The first couple of sections cover the modeling of transmission lines and transformers. In addition, the formulations of admittance and measurement Jacobian matrices, which are essential parts of the state estimation, are described. This chapter concludes with the linearized measurement model, which serves as the foundation of observability analysis in the subsequent chapters.

2.2 Transmission Lines

Transmission lines in power systems are three phase high voltage circuits. It is assumed that each phase is balanced. The generation and load at each phase is also assumed to be even. Thus, single phase analysis can be used to simplify the transmission line model, which is represented as a two-port circuit. The series impedance and total line charging susceptance of a transmission line from bus k to m are shown in Figure 2.1.

R is the resistance of the conductors.
X is the inductance associated with the line.
B is the susceptance between the line and ground.
i_k is the injection current at bus k.
i_{ks} is the current from bus k to ground.
i_{km} is the current from bus k to m.

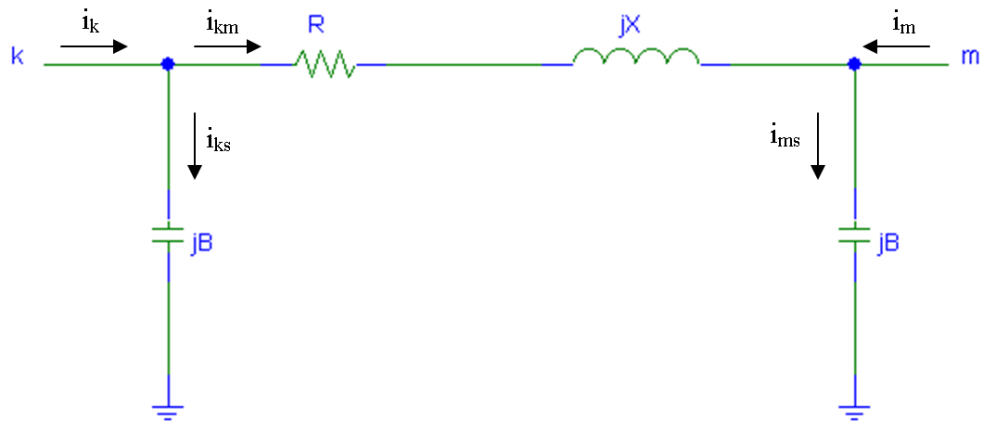

Figure 2.1: 2-port π model of a transmission line

Implementing the Kirchoff's laws, the current injections for buses k and m can be written as:

$$\begin{bmatrix} i_k \\ i_m \end{bmatrix} = \begin{bmatrix} y + jB & -y \\ -y & y + jB \end{bmatrix} \begin{bmatrix} v_k \\ v_m \end{bmatrix} \tag{2.3}$$

where

$$y = \frac{1}{R + jX} \tag{2.4}$$

2.3 Transformers

Tap changing and phase shifting transformers are two types of widely used transformers in power systems.

2.3.1 Tap Changing Transformer

The most common transformer in power systems is the *tap changing* transformer. It is used to step up or down the voltage levels without changing the phase angle of the

voltage. Tap value a of a tap changing transformer is a scalar value, and it changes the magnitudes of the voltage and current on the secondary side of the transformer with respect to the primary side. The voltage induced on the secondary side of the transformer is a times the voltage on the primary side. For the currents, the ratio is reversed, $1/a$.

Once the power is generated in generating stations, it is stepped up by tap changing transformer to higher voltage to decrease the I^2R losses in transmission. Then, it is stepped back down to lower voltage for distribution systems. The tap changing transformer can be modeled by a tap ratio and series impedance shown in Figure 2.2.

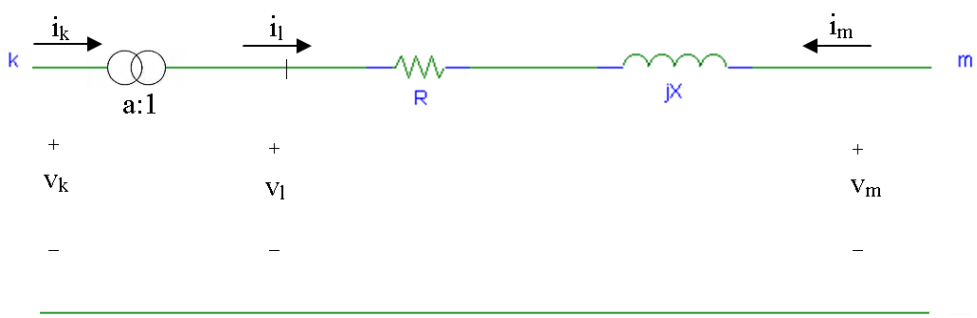

Figure 2.2: Circuit of a tap changing transformer

The admittance of the branch $l-m$, which is the inverse of the series impedance, is used in current calculations. The current from bus k going into the transformer is i_k and it is $1/a$ times the current on the other side of the transformer, i_l. For the voltages, the ratio is reversed:

$$i_k = \frac{1}{a} \cdot i_l \tag{2.5}$$

$$v_k = a \cdot v_l \tag{2.6}$$

$$i_l = y \cdot (v_l - v_m) \tag{2.7}$$

Equations (2.5) and (2.6) are used to replace v_l and i_l in equation (2.7):

$$\begin{aligned} a \cdot i_k &= y \cdot (\frac{v_k}{a} - v_m) \\ i_k &= \frac{y}{a^2} \cdot v_k - \frac{y}{a} \cdot v_m \end{aligned} \tag{2.8}$$

An equation for i_m is derived similar to (2.8). The equations for i_k and i_m are written in matrix form as:

$$\begin{bmatrix} i_k \\ i_m \end{bmatrix} = \begin{bmatrix} y/a^2 & -y/a \\ -y/a & y \end{bmatrix} \begin{bmatrix} v_k \\ v_m \end{bmatrix} \tag{2.9}$$

The transformer current equations derived in this section will be used to write measurement equations for state estimation.

2.3.2 Phase Shifting Transformer

The second type of transformer is the *phase shifting* transformer. Unlike tap changing transformer, phase shifters have complex tap ratio, denoted by a^*. A phase difference between primary and secondary side voltages can be created by a complex tap ratio. The main purpose of a phase shifter is to control the flow of power to prevent congestion. Figure 2.3 shows how voltages are changed by tap changing (Figure 2.3a) and phase shifting transformers (Figure 2.3b).

V_{in} and V_{out} in Figure 2.3 represent the primary and secondary side voltages respectively, and α is the phase angle. The equations for phase shifters are derived similar to the equations for tap changing transformer (2.9). Figure 2.2 used for tap changing transformers can be adjusted to represent phase shifters with complex tap

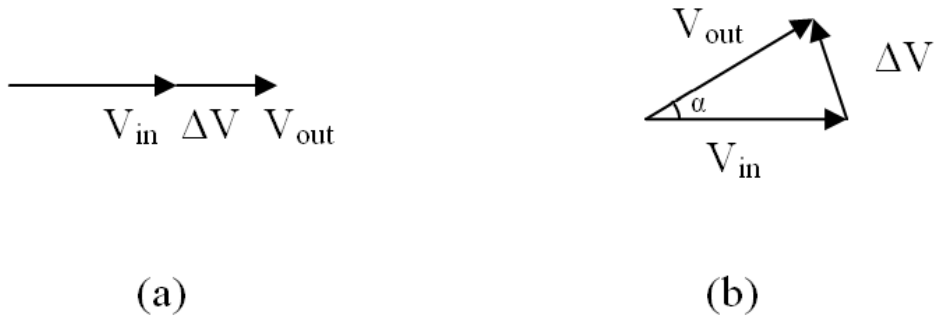

Figure 2.3: Tap changing vs. phase shifting transformer

value in this case.

$$i_k = \frac{1}{a^*} \cdot i_l \tag{2.10}$$

$$v_k = a \cdot v_l \tag{2.11}$$

$$i_l = y \cdot (v_l - v_m) \tag{2.12}$$

Combining these three equations above, i_k can be written in terms of v_k and v_m:

$$\begin{aligned} a^* \cdot i_k &= y \cdot (\frac{v_k}{a} - v_m) \\ i_k &= \frac{y}{|a|^2} \cdot v_k - \frac{y}{a^*} \cdot v_m \end{aligned} \tag{2.13}$$

Similarly, the equation for i_m is derived, and i_k and i_m in matrix form are:

$$\begin{bmatrix} i_k \\ i_m \end{bmatrix} = \begin{bmatrix} y/|a|^2 & -y/a^* \\ -y/a & y \end{bmatrix} \begin{bmatrix} v_k \\ v_m \end{bmatrix} \tag{2.14}$$

2.4 Admittance Matrix

The admittance matrix, usually denoted by Y, is structured such that it contains all the network parameter information, which are line impedances, shunt elements (capacitors and inductors), and transformer taps. The Y matrix for regular lines and transformers are given in equations (2.3), (2.9), and (2.14). Y matrix is used in calculation of bus current injections and therefore bus power injections. The current injections are calculated by Kirchoff's laws, $I = Y \cdot V$. All the current injections for every node in a system are represented as:

$$\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \tag{2.15}$$

where

i_x is the injection current at bus x in phasor form.
v_x is the phasor voltage at bus x.
Y_{ij} is the element of Y in i^{th} row and j^{th} column.

The Y matrix can be formed by treating each line separately. Once the Y matrix is formed for a line between 2 buses, then the same process can be repeated for other lines. Each branch has an admittance value denoted by y_{ij}, where i and j are the buses on each side of the branch. The steps to form the Y matrix are as follows:

1. Y_{ii} is formed by adding all line admittances (y_{ij}) connected to bus i, and all shunt elements such as line charging susceptance and capacitors.

2. Y_{ij} is the negative of the line admittance y_{ij}.

3. Y_{ji} is equal to Y_{ij} since Y matrix is symmetric.

The steps described above hold true for the regular lines with some exceptions for the transformers. The tap ratio of a transformer changes the calculations. A good way to handle transformers is to write their equations independently by (2.9) and (2.14) depending on the type of the transformer, and then incorporate this independent Y matrix by adding its values to the system's Y matrix. Here is an example explaining how to construct the Y matrix for a 3 bus system with a transformer.

Example 2.1**:** Y matrix is calculated for the 3 bus system shown in Figure 2.4.

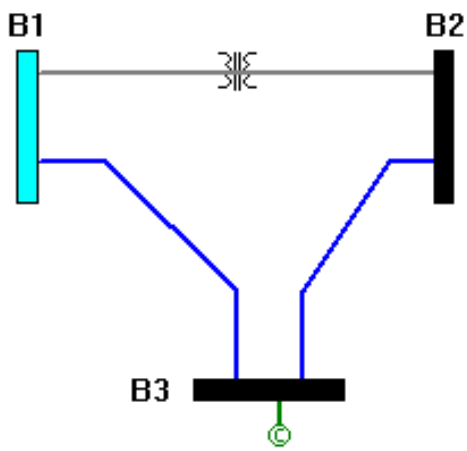

Figure 2.4: One-line diagram for a 3 bus system

The network data for the system in Figure 2.4 is given in Table 2.1.

Table 2.1: Network parameters of the 3 bus system

From Bus	To Bus	R (p.u.)	X (p.u.)	Total Line Charging Suscept.	Tap a	Tap Side Bus
1	2	0.0	0.07	0.0	0.98	1
1	3	0.02	0.05	0.16	N/A	N/A
2	3	0.05	0.08	0.20	N/A	N/A
The shunt capacitor at bus 3 has susceptance 0.5 p.u.						

Note that total line charging susceptance is equal to $j2B$. So, half of this value is added to the diagonal entries of Y. First branch admittance of each line is obtained:

$$\begin{aligned}
y_{12} &= \frac{1}{0+j0.07} = -j14.2857 \\
y_{13} &= \frac{1}{0.02+j0.05} = 6.8966 - j17.2414 \\
y_{23} &= \frac{1}{0.05+j0.08} = 5.6180 - j8.9888
\end{aligned}$$

Since the branch between buses 1 and 2 is a transformer, equation (2.9) is used before the elements of the Y matrix is calculated:

$$\begin{aligned}
\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} &= \begin{bmatrix} (-j14.2857)/(0.98^2) & j14.2857/0.98 \\ j14.2857/0.98 & -j14.2857 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \\
&= \begin{bmatrix} -j14.8748 & j14.5773 \\ j14.5773 & -j14.2857 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}
\end{aligned}$$

Now, the elements of Y are calculated using the values above:

$$\begin{aligned}
Y_{11} &= -j14.8748 + 6.8966 - j17.2414 + j\frac{0.16}{2} = 6.8966 - j32.0361 \\
Y_{12} &= Y_{21} = j14.5773 \\
Y_{13} &= Y_{31} = -6.8966 + j17.2414 \\
Y_{22} &= -j14.2857 + 5.6180 - j8.9888 + j\frac{0.20}{2} = 5.6180 - j23.1745 \\
Y_{23} &= Y_{32} = -5.6180 + j8.9888 \\
Y_{33} &= 6.8966 - j17.2414 + 5.6180 - j8.9888 + j\frac{0.16}{2} + j\frac{0.20}{2} + j0.5 \\
&= 12.5146 - j25.5502
\end{aligned}$$

The final Y is a 3x3 matrix:

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} 6.8966 - j32.0361 & j14.5773 & -6.8966 + j17.2414 \\ j14.5773 & 5.6180 - j23.1745 & -5.6180 + j8.9888 \\ -6.8966 + j17.2414 & -5.6180 + j8.9888 & 12.5146 - j25.5502 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

2.5 Measurement Model

The measurements are modeled as nonlinear functions of voltage magnitude and phase angle. Power injections, power flows, and voltage magnitudes are conventional measurements, which measure real and reactive power in various nodes and branches in the network. Then, they are telemetered to control center by remote terminal units (RTUs) to be processed by functions such as power flow, state estimation, and contingency analysis.

The measurements, represented by a column vector z:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} h_1(x_1, x_2, \ldots, x_n) \\ h_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ h_m(x_1, x_2, \ldots, x_n) \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix} = h(x) + e \tag{2.16}$$

where,

z is the measurement vector.

$h(x_i)$ is the nonlinear measurement function for state i, determined by Y matrix and Kirchoff's laws.

e is the measurement error vector.

m is the total number of measurements.

n is the total number of states.

The values recorded by the meters might have some errors, which can be related to the accuracy of the unit. Some meters might have a slight bias on their readings. Furthermore, there is a room for error in the wireless communication and transfer of the meter readings to the control center. All these errors are modeled as independent random variables with zero mean. Thus, the standard deviation of measurement i is:

$$E[e_i e_i^T] = \sigma_i^2 \tag{2.17}$$

where, E represents the expected value.

The measurement equations are written using Kirchoff's laws with the assumptions that all the line parameters are known and given by the Y matrix, and two port π model is used to represent branches as show in Figure 2.5.

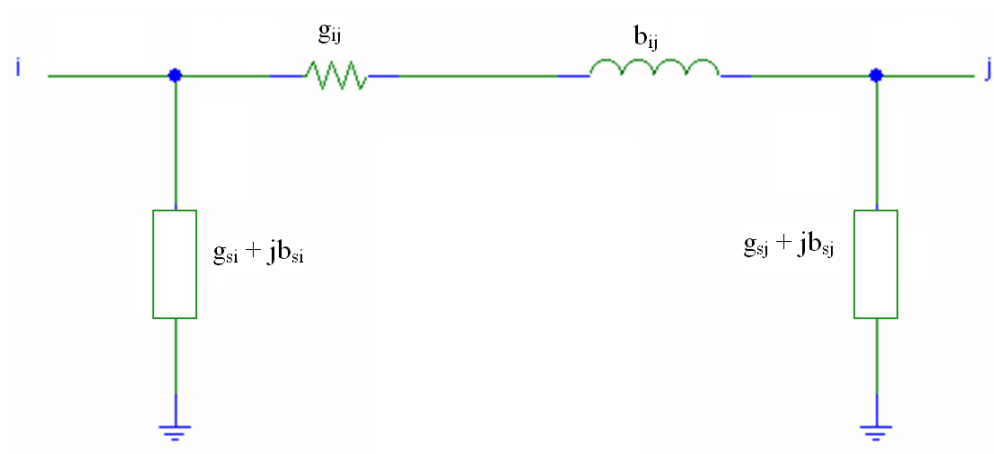

Figure 2.5: 2 port π model used for measurement equations

Using the node voltage method for circuit analysis, the real and reactive power

injections at bus i are: $\hat{V_i} = V_i(\cos\theta_i i + j\sin\theta_i$ $\hat{I_{ij}} = I_{ij}(\cos\theta_i + j\sin\theta_i)$

$$P_i = V_i \sum_{j \in N_i} V_j(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}) \tag{2.18}$$

$$Q_i = V_i \sum_{j \in N_i} V_j(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}) \tag{2.19}$$

Similarly, real and reactive power flows from bus i to j are:

$$P_{ij} = V_i^2(g_{si} + g_{ij}) - V_iV_j(g_{ij}\cos\theta_{ij} + b_{ij}\sin\theta_{ij}) \tag{2.20}$$

$$Q_{ij} = -V_i^2(b_{si} + b_{ij}) - V_iV_j(g_{ij}\sin\theta_{ij} - b_{ij}\cos\theta_{ij}) \tag{2.21}$$

where,

V_i is the voltage magnitude at bus i.

θ_i is the phase angle at bus i.

θ_{ij} is equal to $(\theta_i - \theta_j)$.

N_i is the set of neighboring buses directly connected to bus i.

G_{ij} is the real part of the ij^{th} element of Y matrix.

B_{ij} is the imaginary part of the ij^{th} element of Y matrix.

g_{ij} is the real part of the series branch connecting buses i and j.

b_{ij} is the imaginary part of the series branch connecting buses i and j.

g_{si} is the real part of the shunt branch from bus i to ground.

b_{si} is the imaginary part of the shunt branch from bus i to ground.

2.5.1 The Measurement Jacobian

The measurement Jacobian, denoted by H, is a matrix of first-order partial derivatives of measurements, and it is used in least squares algorithm for estimating the state.

The first order partial derivative of a cost function is set equal to zero to find a minimum, which is explained in the next section. The measurement Jacobian is obtained by taking derivatives of each measurement in the system with respect to the elements in the state vector (voltage magnitudes and phase angles).

Each measurement will have a row in H, so there will be m rows, where m is the total number of measurements. There are two entries, one voltage magnitude and one phase angle, for a bus in the state vector. Therefore, there will be $2n$ elements in the state vector, where n is the total number of buses in the network. Note that one of the phase angles is chosen as reference, which is implemented as taking out that phase angle of the state vector. Thus, the total number of columns in H will be $(2n-1)$. H matrix can be built in the following way:

$$H = \left[\begin{array}{c|c} \frac{\partial P_{inj}}{\partial \theta} & \frac{\partial P_{inj}}{\partial V} \\ \hline \frac{\partial P_{flow}}{\partial \theta} & \frac{\partial P_{flow}}{\partial V} \\ \hline \frac{\partial Q_{inj}}{\partial \theta} & \frac{\partial Q_{inj}}{\partial V} \\ \hline \frac{\partial Q_{flow}}{\partial \theta} & \frac{\partial Q_{flow}}{\partial V} \\ \hline 0 & \frac{\partial V_{mag}}{\partial V} \end{array}\right] \tag{2.22}$$

The rows of the H matrix can be in any order. The order given in the equation above helps group the measurements. Each block in the H is a matrix itself. Consider the first block:

$$\frac{\partial P_{inj}}{\partial \theta} = \begin{bmatrix} \frac{\partial P_1}{\partial \theta_2} & \frac{\partial P_1}{\partial \theta_3} & \cdots & \frac{\partial P_1}{\partial \theta_n} \\ \frac{\partial P_2}{\partial \theta_2} & \frac{\partial P_1}{\partial \theta_3} & \cdots & \frac{\partial P_2}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_k}{\partial \theta_2} & \frac{\partial P_k}{\partial \theta_3} & \cdots & \frac{\partial P_k}{\partial \theta_n} \end{bmatrix} \tag{2.23}$$

where k is the number of real power measurements and n is the number of buses.

Note that θ_1 is chosen as a reference and its corresponding column is omitted.

2.6 WLS Algorithm

The previous sections in this chapter provide the background for the state estimation algorithm. Weighted least squares method is implemented to find the best estimate of the state, where weight is defined as the standard deviation of a measurement. Measurement errors are assumed to be independent, therefore their covariance matrix would only have non-zeros in the diagonals:

$$R = \begin{bmatrix} \sigma_{11} & 0 & \cdots & 0 \\ 0 & \sigma_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{mm} \end{bmatrix}$$

The optimal estimate will be the one where all residuals are minimized. The residual of a measurement is the difference between its observed and calculated values. State estimation minimizes the following objective function by implementing weighted least squares method:

$$\text{minimize} \qquad J(x) = \sum_{i=1}^{m} (z_i - h_i(x))^2 / R_{ii} \tag{2.24}$$

$$\text{subject to} \qquad z_i = h_i(x) + r_i, \quad i = 1, \ldots, m. \tag{2.25}$$

where m is the number of measurements and n is the number of states. In vector from equation (2.24) is written as:

$$J(x) = [z - h(x)]^T R^{-1} [z - h(x)] \tag{2.26}$$

The value of x that minimizes the above equation is defined as the estimated state [4]. $J(x)$ in equation (2.26) is a non-linear function, thus the first derivative is set equal to zero to find a minimum.

$$\begin{aligned} g(x) = \frac{\partial J(x)}{\partial x} &= -H^T(x)R^{-1}[z - h(x)] = 0 \\ \text{where, } H(x) &= \frac{\partial h(x)}{\partial x} \end{aligned} \tag{2.27}$$

Since $g(x)$ in equation (2.27) is non-linear, x cannot be directly calculated. An iterative approach with Taylor series expansion around the state vector x_k gives:

$$\begin{aligned} g(x) &= g(x_k) + \frac{\partial g(x_k)}{\partial x}(x - x_k) + \cdots = 0 \\ \text{where, } \frac{\partial g(x_k)}{\partial x} &= G(x_k) = H^T(x_k)R^{-1}H(x_k) \end{aligned} \tag{2.28}$$

Iterations are carried out by rearranging equation (2.28) until the value of $|x_{k+1} - x_k|$ is less than a small value, determined by the operator:

$$x_{k+1} = x_k - G(x_k)^{-1}g(x_k) \tag{2.29}$$

where k is the iteration number. $G(x)$ is called the gain matrix. $J(x)$ might have local minimums or it might not even converge. In order to help convergence, the state vector is initialized to flat start, where all voltage magnitudes are set to 1 p.u. and all phase angles are set equal to 0 degrees.

It is not a trivial task to take the inverse of a large matrix directly as in equation (2.29). When the number of buses increase, the number and size of the measurement equations get larger making state estimation computationally challenging. Cholesky decomposition and lower-upper (LU) factorization are fast algorithms that can be

used with forward and backward substitution to replace matrix inversion. In this chapter, the Cholesky decomposition is used:

$$G = L \cdot L^* \tag{2.30}$$

where L is a lower triangular matrix called the Cholesky factor. A modified version of equation (2.29) is used to prevent matrix inversion:

$$\begin{aligned} LL^T(x_{k+1} - x_k) &= -g(x_k) \\ \text{where } \ g(x_k) &= -H^T(x_k)R^{-1}[z - h(x_k)] \end{aligned} \tag{2.31}$$

The steps of forward/back substitutions are explained in [6]. The forward substitution is used to find the elements of a temporary vector u:

$$L^T(x_{k+1} - x_k) = u \tag{2.32}$$

$$Lu = -g(x_k) \tag{2.33}$$

In equation (2.33), L is a lower triangular matrix. So, starting from the top, the first row of u can be found by scalar division, $u_1 = -g(x_k)/L_{11}$. Then, replacing the value for u_1 in the second row, u_2 can be found in a similar manner. This is the forward substitution.

Once u is obtained, equation (2.32) is used to find the difference $\Delta x_k = x_{k+1} - x_k$. L^T in equation (2.32) is an upper triangular matrix. The scalar division starts from the last row this time, and the rows of Δx_k are calculated one by one. Since the last row is obtained first, this process is called backward substitution.

2.7 DC Approximation

The measurement models used in the WLS algorithm are non-linear, therefore the state can be estimated only after a number of iterations. If the measurement equations can be linearized somehow, estimate can be calculated directly [7] making state estimation a faster process. On the other hand, the state obtained by using linearized measurements will not be as accurate as the full non-linear ones.

The main purpose of using linearized equations is to analyze inherent properties of a power system. For example, observability analysis checks the measurement placement in a network. The actual meter readings have no role in observability analysis. Only the location and type of a meter affects the observability of a system. Therefore, there is no need to implement full non-linear measurement equations for observability analysis. Since the actual meter reading and parameters are not necessary, further simplification can be achieved with the following approximations:

1. All voltage magnitudes are set equal to 1.0 p.u., $|V| = 1.0$.
2. The branch resistance is usually much smaller than the reactance for a given branch, thus the resistance is ignored ($R \approx 0$), and the branch impedance is set equal to $j1.0$ p.u.
3. All shunt elements such as the line charging susceptance are ignored.

With these assumptions, a real power flow equation (2.20) will look like:

$$P_{ij} = \frac{V_i V_j}{1.0} \sin(\theta_i - \theta_j) = \sin(\theta_i - \theta_j) \tag{2.34}$$

The linearization of measurements is done by Taylor series expansion, ignoring higher

order terms, around $x_0 = 0$, where x is the state vector [8]:

$$h(x) = h(x_0) + \frac{\partial h(x)}{\partial x}(x - x_0) \tag{2.35}$$

The linearized version of equation (2.34) will be:

$$\begin{aligned} P_{ij}(x) &= P_{ij}(x_0) + \tfrac{\partial P_{ij}(x)}{\partial x}(x - x_0) \\ P_{ij}(x) &= 0 + \cos(0)[(\theta_i - \theta_j) - 0] \\ P_{ij}(x) &= \theta_i - \theta_j \end{aligned} \tag{2.36}$$

The state estimation algorithm needs to be modified when linearized measurements are used. The new measurement model is:

$$z = Hx + e \tag{2.37}$$

where H is not same as the Jacobian matrix used for non-linear model. It is a matrix of constants multiplying x. WLS will minimize the new objective function:

$$[z - Hx]^T R^{-1}[z - Hx] \tag{2.38}$$

The first order derivative is set equal to zero to find a minimum:

$$g = -H^T R^{-1}[z - Hx] = 0 \tag{2.39}$$

x can be directly solved in equation (2.39):

$$\hat{x} = (H^T R^{-1} H)^{-1} H^T R^{-1} z \tag{2.40}$$

where $(H^T R^{-1} H)^{-1}$ is defined as the gain matrix G. The linearized measurements with the other assumptions given earlier in this section comprise the DC approximation model. The analogy is that full non-linear model is the AC model, and the linearized one is the DC model. Observability analysis using the DC model is explained in detail in chapter 3.

Chapter 3

Observability Analysis and Special Cases

3.1 Introduction

Observability needs to be checked before running state estimation because it is not possible to estimate the state of an unobservable system. The importance of the observability analysis is that the operators can locate the unobservable parts of the system where additional meters can be placed to improve the system. The initial design and later upgrades of a system can be accomplished by observability analysis. In the literature there are two approaches, numerical and topological, for network observability analysis. The algorithms in this thesis are based on the numerical methods due to its ease of implementation. However, topological method can be used to get similar results. Detailed discussions of topological analysis can be found in [9]–[14].

If the result of the analysis shows that a system is unobservable, then its network

is split into smaller observable areas, which are called **islands** [3]. A state estimation solution can be obtained for the buses within these islands, but branches tying observable islands will remain unobservable. Once islands are identified, a meter placement procedure to merge these islands can be implemented. Optimal meter placement is described in detail in chapter 4, where a practical implementation is also developed. In this chapter, observability of a system and identification of unobservable branches are explained. In addition, unobservable branch information is used to find observable islands. At the end of the chapter an observability algorithm is given, which can be used to identify unobservable branches and determine observable islands.

3.2 Observability Check

The minimum number of real power measurements to have a barely observable system is $(n-1)$, given that one of the buses is chosen as reference bus and n is the number of buses (or states) in the system. Therefore, it is not possible to have an observable system with less than $(n-1)$ measurements.

If there are more than $(n-1)$ meters, gain matrix (G) or measurement Jacobian matrix (H) can be factorized to check observability. For an observable system, G and H are both non-singular. If these matrices are rank deficient, then there are unobservable branches in the system. In this chapter, the G matrix is used for observability analysis and island identification. In chapter 4, the H matrix is preferred due to its use in meter placement algorithm.

The linearized measurement model is used in this chapter since observability is related to the topology and the measurement configuration of the network. Observability analysis is similar to the **tree** concept in graph theory. A power system

network can be thought of as a graph, and the observability analysis is a method to check whether this graph is a tree or not. Furthermore, it is shown in earlier studies [15] that there is a weak coupling between $P - V$ and $Q - \theta$, which can be exploited to formulate decoupled state estimation [16]–[18]. This decoupling can be used to simplify observability analysis. Assuming that P and Q measurements come in pairs either $P - \theta$ or $Q - V$ observability tests can be carried out to get the same results. The $P - \theta$ model is preferred throughout this thesis since it is simpler than the $Q - V$ model. The measurements equations are:

$$z = \begin{bmatrix} z_P \\ z_{QV} \end{bmatrix} = \begin{bmatrix} H_{P\theta} & H_{PV} \\ H_{Q\theta} & H_{QV} \end{bmatrix} \begin{bmatrix} \theta \\ V \end{bmatrix} + \begin{bmatrix} e_P \\ e_{QV} \end{bmatrix}$$

where P and Q represent the real and reactive power measurements. V and θ represent the voltage magnitude and phase angle. H_{PV} and $H_{Q\theta}$ terms can be ignored due to weak coupling between $P - V$ and $Q - \theta$. The decoupled measurements are given as:

$$\begin{aligned} z_P &= H_{P\theta}\theta + e_P \\ z_{QV} &= H_{QV}V + e_{QV} \end{aligned}$$

Although $P - \theta$ test is implemented in this chapter, a voltage magnitude measurement is still needed for each observable island to prevent multiple solutions so that a unique state estimation solution can be found. Observability analysis as described in [3] is based on checking whether branch flows are zero or not. Therefore, the line parameters and the operating state of the system are irrelevant for observability analysis. All lines can be assumed to have $j1.0$ p.u. and all bus voltages can be set equal

to 1.0 p.u. for further simplification. A real power flow from bus k to m and injection at bus k are given as:

$$P_{km} = \frac{1}{X_{km}}(\theta_k - \theta_m) = \theta_k - \theta_m$$
$$P_k = \sum_{j=N_i} \frac{1}{X_{kj}} P_{kj} = \sum_{j=N_i} P_{kj}$$

where the impedances of all lines are set equal to $j1.0$ p.u. and N_i represents all the neighbor buses of bus k. Thus, flows are simply written as the angle differences. In matrix form, the flows can be written as:

$$P_b = A\theta \tag{3.1}$$

where,

P_b is the vector of branch flows.
A is the branch-bus incidence matrix.
θ is the vector of bus voltage phase angles.

The branch-bus matrix is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if bus } j \text{ is the from end of branch } i \\ -1 & \text{if bus } j \text{ is the to end of branch } i \\ 0 & \text{else} \end{cases}$$

The errors in the measurements, $e_{P\theta}$, are ignored since they have no effect on observability. Given the state vector θ, the measurements can be simplified to:

$$z_{P\theta} = H_{P\theta}\theta \tag{3.2}$$

Since only the $P-\theta$ model is used for observability, there is no need to repeat the subscript $P\theta$ in the remainder of the equations in this chapter. Note that similar to measurement errors, the measurement weights do not have any significance for observability either. Hence, all weights are set equal to 1.0. The least squares estimate of the state θ becomes:

$$\begin{aligned} \hat{\theta} &= (H^T H)^{-1} H^T z \\ \hat{\theta} &= G^{-1} t \end{aligned} \tag{3.3}$$

where, $t = H^T z$.

If all measurements are equal to zero in the system, all flows should be zero as well. If a non-zero state, $\theta^* \neq 0$, can be found even though all measurements are zero, then there will be non-zero flows in the system, and such a system will be called **unobservable**. Non-zero flows are on **unobservable branches**.

On the other hand, if all the phase angles are zero, $\theta = 0$, as well as the measurements, then all the flows have to be zero. This implies an **observable** system.

The observable and unobservable cases are summarized in Table 3.1.

Table 3.1: Summary of observable and unobservable cases

	State	Equations	Flows
Observable	$\hat{\theta} = 0$	$H\hat{\theta} = 0$	$P_b = A\hat{\theta} = 0$
Unobservable	$\hat{\theta} \neq 0$	$H\hat{\theta} = 0$	$P_b = A\hat{\theta} \neq 0$

Example 3.1**:** A simple four bus system shown in Figure 3.1 illustrates an unobservable system. There are only two flow measurements, which are equal to zero.

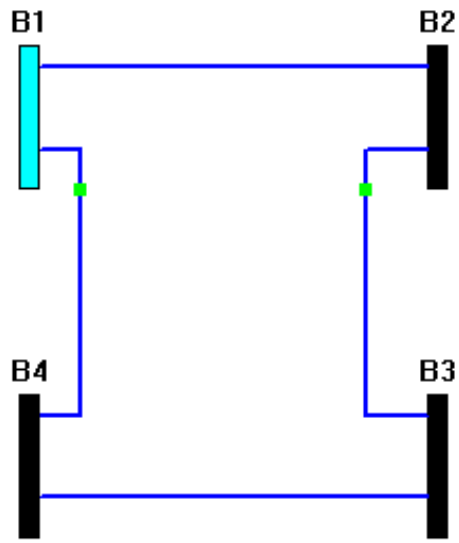

Figure 3.1: 4 bus unobservable system example

The measurements in this system can be written as:

$$P_{14} = \theta_1 - \theta_4 = 0$$

$$P_{23} = \theta_2 - \theta_3 = 0$$

Even though a zero state satisfies the measurement equations above, there are infinite number non-zero solutions. One of the unobservable states that satisfies the measurement equations is:

$$\theta^* = \begin{bmatrix} -3 \\ 1 \\ 1 \\ -3 \end{bmatrix}$$

Therefore, only two power flow measurements are insufficient to make this four bus system observable. One more flow measurement can be placed either on the

branch between buses 1 and 2 or 3 and 4 to restore observability.

3.3 Finding Unobservable Branches

The inverse of the G matrix is part of equation (3.3), which is used to get the estimate of θ. However, the G matrix of an unobservable system is singular and it is not possible to take the inverse of a singular matrix. Therefore, a systematic approach to modify the G matrix by keeping track of unobservable states is required. Determination of unobservable states and identification of unobservable branches are presented in [19].

As in chapter 2, the inverse of the G matrix is calculated by applying forward and backward substitution on the Cholesky factor of G. After Cholesky factorization and multiplication of equation (3.3) by G from the left, (3.3) will look like:

$$(LL^T)\theta = t = 0 \tag{3.4}$$

where L is the Cholesky factor of G. For an unobservable system, L will have at least one zero diagonal. The state vector is divided into two groups, $\theta = (\theta_a, \theta_b)$. θ_b represents the elements of the state corresponding to zero diagonals. θ_a is the vector of all the remaining states.

In order to find solution for unobservable state, the L matrix is modified such that when a zero diagonal is encountered in L, it is replaced by a 1.0. The corresponding element of the right hand side vector, t, in equation (3.4) is assigned a distinct integer. The element of t vector corresponding to the row of the first zero diagonal in L is assigned 1 and the next modified element in t will be incremented by 1 until all zero pivots are covered. For example, if the first 3 rows of t correspond to zero diagonals in L, then t will look like: $t = (1, 2, 3, 0, \cdot, 0)^T$. Note that one of the buses in the

system is picked as the reference bus and its state is not included in the formulation. Therefore, the sizes of G and L reduce to $(n-1)$ by $(n-1)$, where n is the number of states.

Once G and t are modified, the estimate for the state, $\hat{\theta}$, can be calculated by equation (3.3). The next step then is to obtain the flows from equation (3.1), $P_b = A\hat{\theta}$. Any non-zero flows in P_b will identify an unobservable branch. A simple 6 bus example is used to illustrate this method.

Example 3.2**:** An unobservable 6 bus system has 3 measurements, which are two power injections at buses 3 and 4, and a flow measurement between buses 3 and 4, as shown in Figure 3.2.

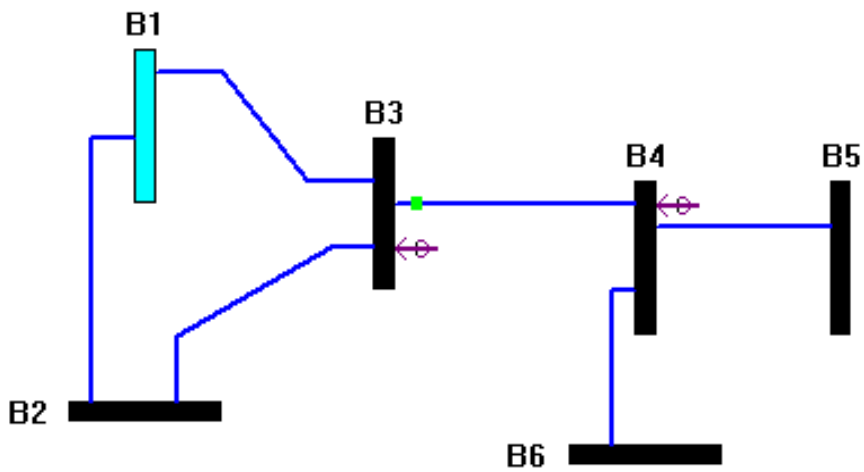

Figure 3.2: 6 bus unobservable system example

The H matrix for 6 bus system is:

$$H = \begin{bmatrix} 0 & -1 & 3 & -1 & -1 \\ -1 & 3 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \end{bmatrix}$$

and branch-bus incidence matrix is:

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

Note that bus 1 is chosen as reference and its corresponding column in H and A matrices is taken out. The columns of the H and A matrices are the states. The rows of H and A matrices represent measurement equations and branches respectively. The gain matrix is $G = H^T H$:

$$G = \begin{bmatrix} 1 & -3 & 1 & 0 & 0 \\ -3 & 11 & -7 & 1 & 1 \\ 1 & -7 & 11 & -3 & -3 \\ 0 & 1 & -3 & 1 & 1 \\ 0 & 1 & -3 & 1 & 1 \end{bmatrix}$$

The Cholesky factor of G is:

$$L = \begin{bmatrix} 1.0000 & 0 & 0 & 0 & 0 \\ -3.0000 & 1.4142 & 0 & 0 & 0 \\ 1.0000 & -2.8284 & 1.4142 & 0 & 0 \\ 0 & 0.7071 & -0.7071 & 0 & 0 \\ 0 & 0.7071 & -0.7071 & 0 & 0 \end{bmatrix}$$

The fourth and fifth diagonals of L are replaced by 1.0. The right hand side vector is also updated such that:

$$t = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 \end{bmatrix}^T$$

Now, equation (3.3) will give the estimated state:

$$\hat{\theta} = (LL^T)^{-1}t = \begin{bmatrix} 3.0000 \\ 1.5000 \\ 1.5000 \\ 1.0000 \\ 2.0000 \end{bmatrix}$$

The estimated state is used to calculate the flows in equation (3.1):

$$P_b = A\hat{\theta} = \begin{bmatrix} -3.0000 \\ 1.5000 \\ 0 \\ 0.5000 \\ -0.5000 \\ -1.5000 \end{bmatrix}$$

Only third element in P_b is zero and the rest of the elements are non-zeros, which correspond to unobservable branches. Since each element of P_b matches a row of A and each row of A is a branch, the unobservable branches correspond to all rows of A except row 3:

Table 3.2: Unobservable branches in Figure 3.2

Non-zero rows of P_b	Unobservable Branches
1	Bus 1 to 2
2	Bus 2 to 3
4	Bus 4 to 5
5	Bus 4 to 6
6	Bus 1 to 3

3.4 Identification of Observable Islands

The observable islands in a system are separated by unobservable branches. If every branch in a system is found to be unobservable, then each bus is an island. The unobservable branches in Figure 3.2 from previous example are used to determine the observable islands. There are 5 islands which are shown in the Figure 3.3.

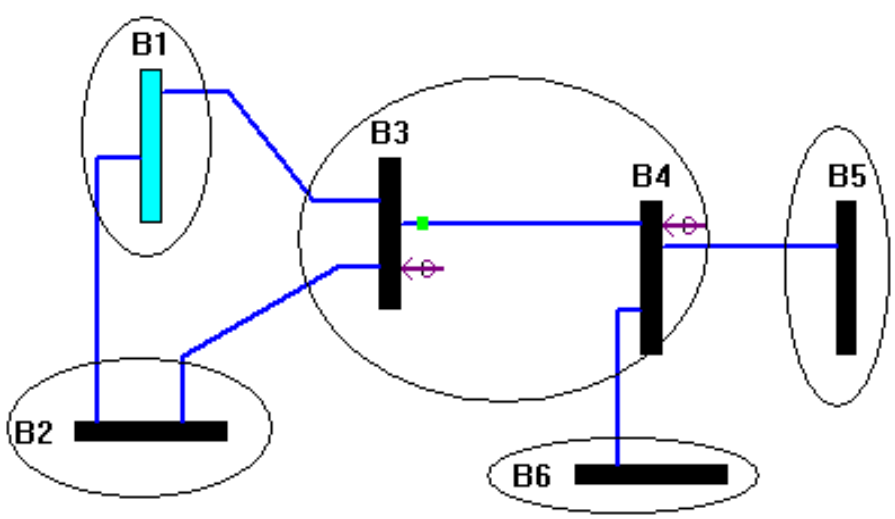

Figure 3.3: Islands in 6 bus system

The observability algorithm finding unobservable branches and determining observable islands is summarized below:

Step 1 Form the measurement Jacobian matrix H, and branch-bus incidence matrix A.

Step 2 Calculate the gain matrix $G = H^T H$, and check the rank of G. If it is full rank, the system is observable. Stop. Else, go to step 3.

Step 3 Find the zero diagonals in the Cholesky factor of G and replace them by 1's. Assign distinct integers to the corresponding rows of t vector in equation (3.4).

Step 4 Calculate the estimate of the state from equation (3.3), $\hat{\theta} = (LL^T)^{-1}t$.

Step 5 Find the non-zero rows of the flow vector P_b, using equation (3.1), which will give the unobservable branches.

Step 6 Take out the unobservable branches from the system and determine the islands which are connected only by the remaining observable branches.

Conventional measurements (power injections and flows) as well as PMUs can be added to the system to merge observable islands. In chapter 4, a practical measurement placement algorithm, which finds an optimal set of either conventional measurements or PMUs to restore observability, is explained.

Chapter 4

Meter and PMU Placement for Observability

4.1 Introduction

Network observability is directly related to the number and location of existing measurements. Due to variations in the system such as topology changes or meter failures, power system operators need to check observability prior to running state estimation. In the case that the available measurements are not sufficient to estimate the state, a meter placement procedure needs to be implemented to find additional measurements to render the system observable again.

The results of meter placement can be used in two different ways. The first approach provides an immediate fix to the system by finding the minimum number of pseudo measurements so that the state can be estimated again. Load forecasts and scheduled generation data are examples of sources that are used to generate pseudo measurements. The second possibility is to upgrade the measurement configuration

permanently by installing additional measurements at the locations identified by the meter placement procedure. The latter approach is preferable since real measurements will provide more accurate and reliable values than the pseudo measurement counterparts.

The basic solution to meter placement is to place measurements at the boundaries of observable islands [3]. When an injection measurement is installed at a boundary of two islands, it merges these two islands into one. Phasor measurement units (PMUs) can also be used to merge islands by establishing a common reference in the system. The advantage of PMU placement is that a PMU can be installed at any bus in an island to merge it to another island. In practice, power system operators do not have enough resources to place measurements at every boundary. Therefore, a systematic approach is required to find an optimal solution to select measurements in a fast and efficient way.

The optimal meter placement problem is addressed before in various studies [19]–[22]. The use of sparsity techniques is common in some of these studies [20, 22] due to sparse nature of measurement equations. The sparse triangular decomposition of the measurement Jacobian is used to eliminate unnecessary measurements whose addition to the system will result in no improvement in the network observability. Once the unnecessary measurements are eliminated, all the remaining measurements are candidates for merging observable islands. Reference [21] explains a non-iterative method for optimal meter placement, which also uses sparse factorization. One drawback to the algorithm in [21] is that the lower triangular factor of the measurement Jacobian is inverted, which is a computationally difficult operation for large systems. When a sparse matrix is inverted, the result is a nearly full matrix, which increases the time complexity of an algorithm.

The approach in this thesis has computational advantages since the running time mainly depends on the sparse triangular decomposition. Therefore, the proposed method can be implemented to solve measurement placement problems in large systems, which are in the order of thousands of buses.

The organization of this chapter is such that the triangular decomposition of the measurement Jacobian is described first. The lower and upper triangular matrices calculated at the end of the decomposition include all the information necessary to identify candidate measurements. There is an installation and maintenance cost associated with each candidate measurement. The next section explains the use of this cost information and binary integer programming to optimize the selection of the candidates. Finally, some simulations will be shown, in which the proposed method is applied to solve IEEE 14 and 30 bus systems.

4.2 Assessment of the Measurements

4.2.1 Measurement Jacobian

The measurements can be classified into two groups: conventional measurements and phasor measurement units (PMUs). Power injections, power flows, and voltage magnitudes are conventional measurements, which measure real and reactive power in various nodes and branches in the network. Then, they are telemetered to control center by remote terminal units (RTUs) to be processed by functions such as power flow, state estimation, and contingency analysis.

PMUs measure the voltage phasor and the line current phasors at a given bus [23]. There are channels in PMUs assigned for each phasor measurement. Depending on the number of channels in a given PMU, several or all line current phasors along

with the voltage phasor at a given bus can be measured [24]. The GPS technology enables the synchronization of PMUs. One of the reasons why the popularity of PMUs has increased within the last decade is that they improve the state estimation performance [25]. In this chapter, injection and PMU placements for observability are discussed. Chapter 5 describes a different placement algorithm for improving the measurement reliability.

Observability of a given network is determined by the type and location of the available measurements, and also by the topology of the network. In analyzing the inherent limitations of a measurement configuration, it is often helpful to work with a simplified DC approximation model of the measurement equations from chapter 2.

Consider the linear decoupled measurement model:

$$z = H\theta + e \tag{4.1}$$

where,

z is the vector of real power measurements.
H is the decoupled measurement Jacobian.
θ is the vector of phase angles.
e is the measurement error vector.

The column rank of H is an indicator of network observability, full rank representing an observable system. Note that one of the buses is selected as slack bus and its corresponding column in H is taken out. Therefore, the column rank of H will be at most $(n-1)$, where n is the number of buses. There is no need to select a slack bus if PMUs are used for observability. The PMU solution will have one more measurement than the injection solution since one of the PMUs can be thought as reference PMU. Each row of H correspond to a measurement equation.

Let the rows of H be ordered such that the essential measurements are listed first:

$$H = \begin{bmatrix} H_e \\ \cdots \\ H_r \end{bmatrix} \begin{array}{l} \{ \quad \text{essential measurements} \\ \\ \{ \quad \text{redundant measurements} \end{array}$$

where, H_e and H_r indicates the essential and redundant measurements respectively. A fully observable system needs to have $(n-1)$ essential measurements. Therefore, there remains $(m-n+1)$ redundant measurements, where m is the total number of measurements. Evaluation of network observability can be achieved by sparse triangular factorization. The essential, redundant, and candidate measurements can be identified by analyzing the lower and upper triangular matrices obtained at the end of the factorization. If a system is unobservable, then the measurement Jacobian will be rank deficient, and zero pivots will appear in the triangular matrices. Depending on the factorization method implemented, these zero pivots will be encountered in the lower or upper triangular matrices.

4.2.2 Sparse Factorization

The triangular factorization of H with row pivoting will yield the following:

$$H = \begin{bmatrix} L_e \\ \cdots \\ M_r \end{bmatrix} \begin{bmatrix} U_e \end{bmatrix}$$

where, L_e and U_e are the sparse lower and upper triangular matrices, and M_r is the sparse rectangular matrix corresponding to the redundant measurements. At the end

of factorization, any zero diagonals in U_e implies that the available measurements are not sufficient to observe the system. In other words, there are not $(n-1)$ essential measurements. Once zero diagonals are identified, an optimal meter placement procedure is required to select candidate measurements to restore observability.

4.3 Optimal Meter Placement

4.3.1 Identifying Candidates

The zero diagonals in U_e indicates that the system has more than one observable island. Placement of an injection measurement at a boundary of any two islands will merge these two islands [3]. Therefore, all boundary injection measurements can be treated as candidate measurements. A PMU can be used for same purpose. Only one voltage phasor measurement installed at any bus in an observable island is sufficient to merge these two islands. Therefore, a PMU represents one voltage phasor measurement in this chapter. The line current phasors are ignored, but they can be added to the measurement Jacobian if they are already available. The measurement Jacobian corresponding to the candidate measurements can be added to the bottom of H as if they were part of the measurement list as shown:

$$H^{mod} = \begin{bmatrix} H \\ \dots \\ H_c \end{bmatrix} \begin{array}{ll} \} & \text{available measurements} \\ & \\ \} & \text{candidate measurements} \end{array}$$

where, H^{mod} is the new measurement Jacobian, H and H_c represent the available and candidate measurements respectively.

Factorization of H^{mod} without row pivoting cannot be completed since zero diagonals will cause numerical problems such that some entries of H_c will be divided by zero. This numerical problem can be solved by adding 1's to the entries of H corresponding to the zero diagonals. Since the rows of zero diagonals are linearly independent, addition of 1's to the entries in these rows will not affect the rest of the system. The modified H is called H_0. Now, H_0 appears to have full column rank, and therefore triangular factorization can be carried out. The modified H^{mod} is given by:

$$H^{mod} = \begin{bmatrix} H_0 \\ \dots \\ H_c \end{bmatrix}$$

where, $H_0(i,j) = H(i,j) + 1$, if $H(i,j)$ corresponds to a zero diagonal (i and j are the row and column indices of H_0 respectively). Here is how lower triangular factor of H^{mod} as a result of triangular decomposition will look like:

$$L^{mod} = \begin{bmatrix} x & 0 & \cdots & \cdot & \cdot & 0 \\ & x & \cdots & \cdot & \cdot & 0 \\ & & \ddots & \cdot & \vdots & \vdots \\ & & & p & \cdot & \cdot \\ & & & & p & 0 \\ & & & & & x \\ & & & \vdots & \vdots & \vdots \\ & & & \cdot & \cdot & \cdot \\ & & & c & & \cdot \\ & & & \cdot & c & \cdot \\ & & & c & & \\ & & & & c & \cdot \end{bmatrix}$$

where, x's are non-zeros, p's are modified non-zeros, and c's are non-zeros representing the candidate measurements. Note that there were zeros in the places of p's before 1's were added to the entries of H. As seen in matrix L^{mod} above, there are usually more than one non-zero underneath a given zero diagonal. In other words, there are more than one candidate measurement to merge two or more islands. Therefore, an optimization procedure is required to select an optimal number of measurements.

4.3.2 Selecting Candidates

Let us assume that there is a set of candidate measurements for each zero diagonal, and these candidates are represented in a binary matrix A, where the rows and

columns of A stand for the zero diagonals and candidates respectively [20]. A is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if measurement } j \text{ is a candidate} \\ & \text{for zero pivot } i \\ 0 & \text{otherwise} \end{cases}$$

The objective is to optimize the selection of candidates not only by finding the smallest number of measurements, but also minimizing the installation and maintenance costs of these measurements. The information for installation and maintenance costs are stored in a cost vector. Then, the binary integer programming is employed to solve the optimization problem given the cost vector C, and the binary matrix A:

$$\begin{aligned} \text{minimize} \quad & C^T \cdot X \\ \text{subject to} \quad & A \cdot X \geq b \end{aligned}$$

where $b^T = [11 \cdots 1]$, is a vector of 1's. The output of integer programming will be a status vector X, as shown below, which stores the selected measurements as follows:

$$X(i) = \begin{cases} 1 & \text{if measurement } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

where, i is the measurement index.

The constraint $A \cdot X \geq b$ ensures that at least one candidate measurement is selected for each zero diagonal. If a measurement is assigned to more than one zero diagonal, then this measurement is favored by the constraint.

One measurement can take care of at least one zero diagonal. For example, a given measurement can be assigned to 3 zero diagonals representing 3 separate islands, and

if this measurement can only merge 2 of these 3 islands, leaving the last island unobservable, then extra measurements are needed. Therefore, the list of measurements selected in the first implementation of integer programming might not be sufficient for observability. It is necessary to recheck the observability of the system after each optimization scheme until observability is ensured. In fact, this iterative approach provides optimal solution by picking a minimum number of measurements in each iteration.

In the beginning of each repetition, similar to the initial observability check, the lower and upper triangular factors of the new measurement Jacobian are observed for zero diagonals. Since the number of measurements has increased with the addition of candidate measurements from the previous iteration, the triangular factorization will yield either no zero diagonals or a smaller number of zero diagonals. No zero diagonals imply that the system is observable. In the case that there are still zero pivots in the lower triangular matrix, the optimal meter placement procedure needs to be repeated until there are no more zero pivots. The following algorithm summarizes the proposed method:

Step 1 Form H of available measurements and perform triangular factorization. If H has less rows than columns, add rows of zeros to make it a square matrix. Additional zero rows acts as place holders for the missing essential measurements.

Step 2 Identify entries of H corresponding to zero diagonals and add 1 to each one of them, forming H_0.

Step 3 Form the candidate measurement Jacobian and add it to the bottom of H_0, forming H^{mod}.

Step 4 Perform triangular factorization of H^{mod} and identify candidate measurements.

Step 5 Use binary integer programming to select candidate measurements.

Step 6 Extend the original H by adding selected candidate measurements as the last rows of H.

Step 7 Repeat steps 2-6 until there is no more zero diagonals.

4.4 Numerical Example

A simple 6 bus system is used to illustrate the proposed method. The measurement configuration of the system is shown in Figure 4.1. Two solutions of this problem is presented in this example. First, the observability is restored by placing only power injections. Then, the same algorithm is implemented with PMUs to show the difference between injections and PMUs.

4.4.1 Injection Solution

Step 1 Form H of available measurements and perform triangular factorization. If H has less rows than columns, add rows of zeros to make it a square matrix:

$$H = \begin{bmatrix} -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

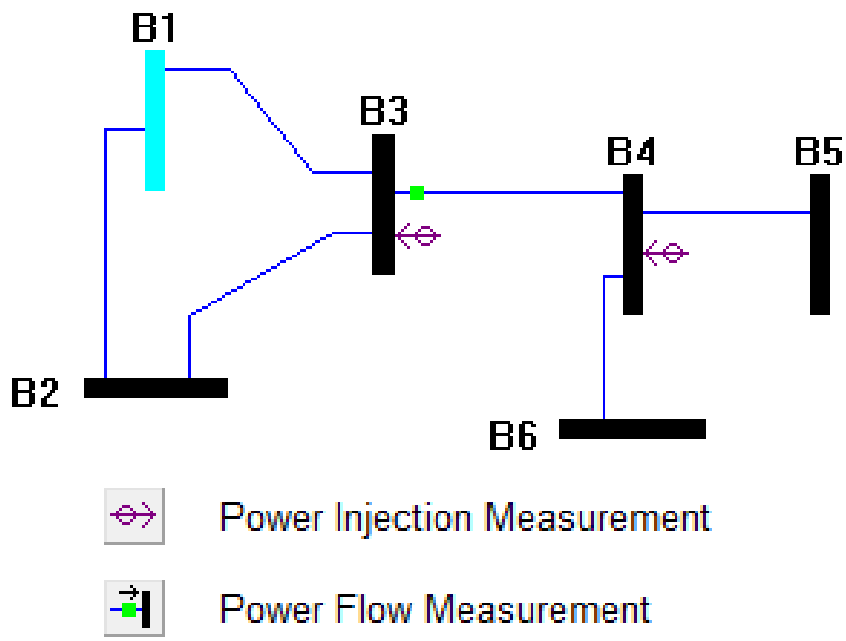

Figure 4.1: 6 bus system example

Note that first column is taken out of H as the reference bus and 2 rows of zeros are added to H. The lower triangular matrix L_e is:

$$L_e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The upper triangular matrix U_e is:

$$U_e = \begin{bmatrix} -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 2 Identify entries of H corresponding to zero diagonals and add 1 to each one of them, forming H_0:

$$H_0 = \begin{bmatrix} -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The fourth and fifth columns of U_e represents the zero diagonal entries of H. Thus, we add 1's to the fourth and fifth diagonals of H.

Step 3 Form the candidate measurement Jacobian and add it to the bottom of H_0,

forming H^{mod}:

$$H^{mod} = \begin{bmatrix} -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

The last 4 rows of H^{mod} represent the candidate injection measurements at buses 1, 2, 5, and 6.

Step 4 Perform triangular factorization of H^{mod} and identify candidate measurements.

$$L^{mod} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 4 & -5.5 & -1.5 & -1.5 \\ -2 & -5 & 6.5 & 1.5 & 1.5 \\ 0 & 0 & -0.5 & 0.5 & -0.5 \\ 0 & 0 & -0.5 & -0.5 & 0.5 \end{bmatrix}$$

Non-zeros underneath columns 4 and 5 are the candidate measurements.

Step 5 Use binary integer programming to select the candidate measurements.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The rows of A correspond to the zero diagonals and the columns indicates the candidate injections at buses 1, 2, 5, and 6. Note that any injection can be chosen since A is full of 1's. Injection at bus 1 is picked as the first candidate.

Step 6 Extend H by adding selected candidate measurements to the bottom of original H.

$$H = \begin{bmatrix} -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \end{bmatrix}$$

Note that there are 4 measurements in the system with the addition of injection 1.

Step 7 The system is still not observable since there are less than $n-1$ rows in H. Steps 2-6 are repeated until there is no more zero diagonals.

Step 2' Factorization of new H yields a zero diagonal at column 5.

Step 3' Since injection 1 was selected in the last iteration, there are 3 injections left as candidates (Inj. 2, 5, and 6).

Step 4' Triangular factorization of H^{mod} returns non-zeros at rows corresponding to injections 5 and 6.

Step 5' Binary integer programming picks injection 5 as the second candidate.

Step 6' Adding a row to the bottom of H designating injection 5 forms a measurement Jacobian of full rank, which represents an observable system and there

is no need to iterate anymore. So, the final candidates to render this system observable are injections at buses 1 and 5.

4.4.2 PMU Solution

Since there are no PMUs installed in this system, voltage phasors can be installed at any bus in the system. The Jacobian matrix with candidate PMUs will look like:

$$H^{mod} = \begin{bmatrix} -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that there are 2 differences between this solution and the injection solution. The candidate part of H^{mod} is an identity matrix of size 6x6 representing candidate voltage phasors. Also, the first column of H is not taken out since one of the PMUs in the solution will set the reference. The same placement algorithm iteratively adds PMUs until the system becomes observable. After two iterations, 3 voltage phasors at buses 1, 2, and 5 are found to merge all the islands. This solution has one more measurement than the injection solution as expected.

4.5 Simulation Results

In the following sections, the IEEE 14 and 30 bus systems used in Reference [21], are simulated and similar results are obtained in terms of the number of measurements identified as candidates. The difference between these two methods is that the approach explained in this thesis is computationally faster since no matrices are inverted. Inversion operation of a sparse matrix will yield a nearly full matrix. For small systems such as 14 and 30 buses, full matrices can be handled. However, once the system size approaches thousands of buses or larger, then it becomes a computational challenge to invert these large matrices. This challenge can be avoided by only using sparse factorization as the major operation on matrices. The PMU solutions of the same systems are also given followed by injection solution.

4.5.1 IEEE 14 Bus System

The IEEE 14 bus system and its measurement configuration is shown in Figure 4.2. There are a total of 7 injection measurements and 8 flow measurements. Injections are at buses 1, 2, 3, 7, 9, 12, and 14, and flows are at branches 1-2, 1-5, 2-3, 4-7, 4-9, 6-13, 7-8, and 7-9.

Injection Solution

The initial triangular factorization identifies 2 zero diagonals at columns 10 and 13. In the first iteration, an injection is placed at bus 6, which is not sufficient by itself to make the system observable. Therefore, one more iteration places another injection at bus 5, and finally the system becomes observable. For this measurement configuration, observability is restored by placing 2 injections at buses 5 and 6.

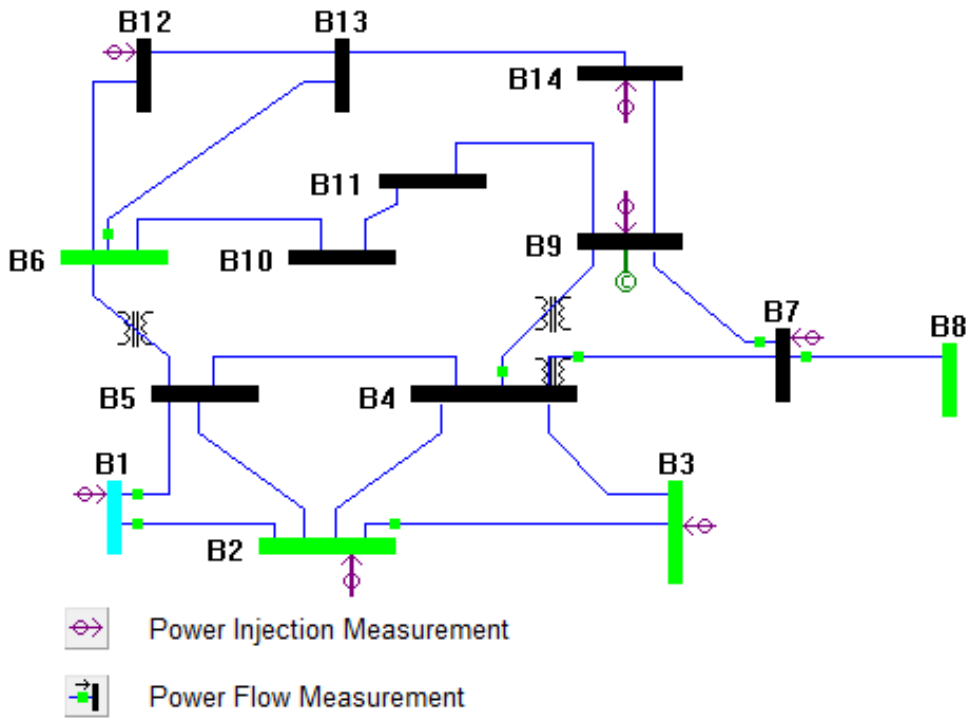

Figure 4.2: IEEE 14 bus system

PMU Solution

For the system in Figure 4.2, the observability is restored with 3 PMUs at buses 6, 9, and 10 in 2 iterations. There is one more measurement in the PMU solution than the injection solution because one of the PMUs is used to set the reference phase angle of the system.

4.5.2 IEEE 30 Bus System

The measurement configuration of IEEE 30 bus system is shown in Fig. 4.3. There are a total of 34 measurements, 10 injections and 24 flows, in the system. The list of measurements are given in Table 4.1.

Injection Solution

The triangular factorization finds 2 zeros at 26th and 29th pivots. An injection at bus 27 is placed at the end of first iteration, but it is not enough for observability. Another injection is placed at bus 22 and the system becomes observable. Therefore, the candidates for this system are injections at buses 22 and 27.

PMU Solution

The observability of the system in Figure 4.3 is restored by placing 3 voltage phasors at buses 1, 24, and 29 in only 2 iterations. The number of iterations is the same as the injection solution, but there is one more measurement in the PMU solution because a specific slack bus is not selected for this case.

4.6 Conclusions

In this chapter, placement of new meters in order to make an unobservable system fully observable is described. A numerically efficient method is proposed for this purpose. Reliable measurement configurations can be achieved by installing these additional meters rather than using pseudo measurements. The utilization of sparse triangular decomposition in the proposed algorithm leads to the better use of the computer resources such as memory and processor. Therefore, this algorithm can be applied to larger systems with thousands of buses to get a measurement placement solution.

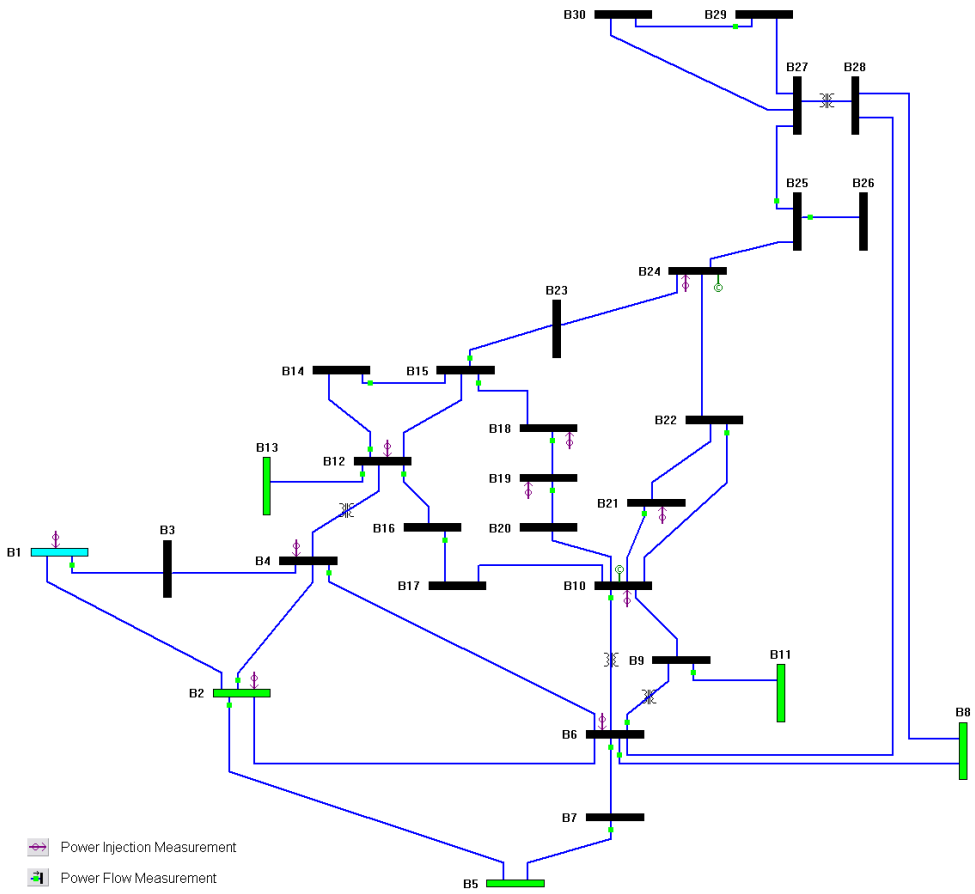

Figure 4.3: IEEE 30 bus system

Table 4.1: Measurements in the IEEE 30 bus system

	Injections	Flows
Bus Numbers	1	1-3
	2	2-4
	4	2-5
	6	4-6
	10	6-7
	12	6-8
	18	6-9
	19	7-5
	21	9-11
	24	10-6
		12-13
		12-14
		12-16
		14-15
		15-18
		15-23
		16-17
		18-19
		19-20
		21-10
		22-10
		25-26
		25-27
		29-30

Chapter 5

Meter and PMU Placement for Bad Data Detection

5.1 Introduction

Similar to chapter 4, this chapter describes meter placement, but for a different purpose. Unlike chapter 4, an observable system is given to start with so that there are enough meters to estimate the state of the system. Thus, this chapter explains a different problem, determining how reliable the state estimation solution is and eliminating the weaknesses if there are any. Being able to run state estimation does not guarantee an accurate solution. There might be bad measurements in the system affecting the results. Bad data can occur for a number of reasons such as telemetry errors, unreported topology changes, and inherent equipment biases [26]. There are well explained methods for detection and identification of bad data in the literature [27]–[31]. If there is not enough meter redundancy at certain locations, there is no way to detect bad measurements. In this chapter, the goal is to bring the system into

a condition by adding a minimum number of meters so that all the bad data can be detected and identified. The actual process of finding the bad data is not part of the algorithm.

If a system is barely observable, then there are critical measurements whose loss would make the system unobservable [20, 32]. In addition, it is not possible to detect any errors in critical measurements. Therefore, elimination of critical measurements will improve system's reliability and security. The goal in this chapter is to find the minimum number of additional measurements so that all the weaknesses are eliminated. In other words, none of the measurements remain critical.

Identification of critical measurements are well documented in [33] and [34]. An algorithm for elimination of critical measurements by meter placement is given in [35]. Different types of additional meters can be available. In this chapter, conventional measurements as well as PMUs, are implemented in the measurement placement algorithm. A PMU placed at a given bus can measure the voltage phasor at that bus and the phasor currents of a single or several lines depending on the type of the PMU used [36]. In this chapter, it is assumed that a given PMU measures the voltage phasor at a given bus and a single phasor current of one of the branches connected to that bus. If there is a need to measure other line current phasors branching out of the same bus additional PMUs can be installed. In the next section, a method to identify critical measurements is given followed by measurement placement algorithm to enable bad data detection. Last two sections consist of the illustrations of placement algorithm. IEEE 14 bus and 30 bus systems are used as examples.

5.2 Identification of Critical Measurements

An observable system needs to have at least $(n-1)$ measurements, which are linearly independent, where n is the number of states. In such a system, all the measurements will be critical since removal of any measurement will turn the system into an unobservable condition. On the other hand, if there are more than $(n-1)$ measurements, then some $(m-n+1)$ of those measurements, will be redundant or linearly dependent. Thus, the first task is to find a set of $(n-1)$ linearly independent measurements and call them *essential measurements*. The remaining measurements will be redundant. All these analyses are related to the location of meters and topology of the network. The actual parameters do not effect the results, therefore decoupled $(P-\theta)$ DC approximation model from chapter 3 is used in this chapter:

$$z_P = H_{P\theta} + e_P \tag{5.1}$$

$$H_P = \begin{bmatrix} H_{P,e} \\ H_{P,r} \end{bmatrix} \tag{5.2}$$

where P, e, and r stand for real power, essential, and redundant measurements respectively. Since only the DC model is used in this chapter, all P subscripts are ignored from this point on to simplify the notation. Applying LU factorization with row shifting will yield:

$$T \cdot H = \begin{bmatrix} L_e \\ M_r \end{bmatrix} \cdot U_e = L \cdot U_e \tag{5.3}$$

where L_e and U_e are lower and upper triangular factors. T is the permutation matrix that stores the row shifting information. M_r is the factorized version of redundant measurement Jacobian, H_r.

The rank of L_e is $(n-1)$ for an observable system given that first column of H is taken out because θ_1 is chosen as reference. The L matrix can be multiplied by L_e^{-1} from right to see the linear dependency relation better between the rows of L:

$$L' = L \cdot L_e^{-1} = \begin{bmatrix} L_e \cdot L_e^{-1} \\ M_r \cdot L_e^{-1} \end{bmatrix} = \begin{bmatrix} I \\ K_r \end{bmatrix} \tag{5.4}$$

where I is an identity matrix of size $(n-1)$. This transformation gives matrix K_r whose columns are checked to identify critical measurements. If a column of K_r has all zeros, then the measurement corresponding to the row index of that column will be critical. For example, if the only non-zero in the first column of L' is the entry $I(1,1)$, then the measurement corresponding to first row of L' is critical. On the other hand, if there are any other non-zeros in the first column of K_r, then the first row of L' is not linearly dependent, therefore not critical. Before a simple 5 bus example is used to illustrate this method, the algorithm can be summarized as:

Step 1 Form the measurement Jacobian, H, and apply LU factorization with row shifting as in equation (5.3)to get L, U_e, L_e, M_r and T matrices.

Step 2 Transform L matrix to L' by equation (5.4). Obtain I and K_r.

Step 3 Find columns of K_r that have all zeros. The corresponding rows of zero columns are critical measurements.

Example 5.1**:** A five bus system is given in Figure 5.1, which has 5 measurements. The algorithm identifying critical measurements is applied in the following way:

Step 1 Form the measurement Jacobian, H, and apply LU factorization in equation

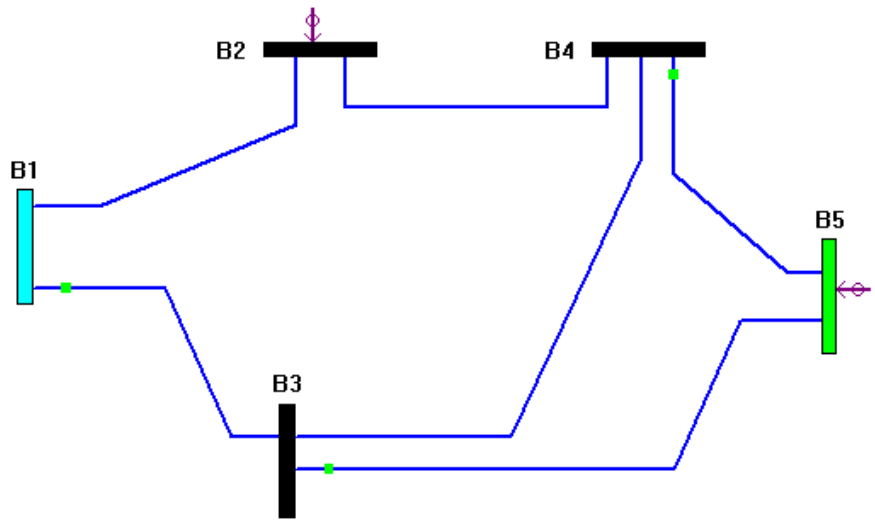

Figure 5.1: 5 bus system analyzed for critical measurements

(5.3):

$$H = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & -1 & -1 & 2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \tag{5.5}$$

$$
\begin{aligned}
T \cdot H &= \begin{bmatrix} L_e \\ M_r \end{bmatrix} \cdot U_e \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & -1 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & -1 & -1 & 2 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{5.6}
$$

Step 2 Transform L matrix to L' by equation (5.4). Obtain I and K_r:

$$
\begin{aligned}
L' = L \cdot L_e^{-1} &= \begin{bmatrix} L_e \cdot L_e^{-1} \\ M_r \cdot L_e^{-1} \end{bmatrix} \\
&= \begin{bmatrix} I \\ K_r \end{bmatrix} = \begin{array}{c} P_2 \\ P_5 \\ P_{13} \\ P_{45} \\ P_{35} \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \text{-1} & 0 & \text{-1} \end{bmatrix}
\end{aligned} \tag{5.7}
$$

Step 3 Find columns of K_r that have all zeros. The corresponding rows of zero columns are critical measurements:

$$
L' = \begin{array}{c} P_2 \\ P_5 \\ P_{13} \\ P_{45} \\ P_{35} \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \text{-1} & 0 & \text{-1} \end{bmatrix} \begin{array}{l} \leftarrow \text{critical} \\ \\ \leftarrow \text{critical} \\ \\ \leftarrow \text{redundant} \end{array} \tag{5.8}
$$

The results of equation (5.8) show that there are 2 critical measurements, which are injections at bus 2 (P_2), and flow from bus 1 to 3 (P_{13}). The power flow from bus 3 to 5 (P_{35}) is the only redundant measurement in the system.

5.3 Measurement Placement

The elimination of the critical measurements can be achieved by placing either conventional measurements or PMUs [35]. Although the theory used in both cases is the same, the implementation of PMUs is slightly more involved. In a Jacobian matrix, PMUs are represented by more than one row depending on how many channels it has. In this chapter, since a single PMU is assumed to measure only one voltage phasor and one current phasor, two rows represent one PMU in H. The first row is the voltage phasor, and the second row is the current phasor. The conventional case is considered before PMUs to explain the theory of placement.

5.3.1 Conventional Measurement Placement

The installed meters in a system can be called available measurements. If they are not sufficient to convert all critical measurement into redundant ones, more meters need to be installed. All the locations that do not have a meter already installed are possible places to put more measurements. The set of absent measurements can be called remaining measurements. Not all of the remaining measurements can eliminate a critical measurement. In this work, to reduce the number of the remaining measurements, which can be flows or injections, only absent power injections are considered.

With a few extra steps the algorithm for identification of the critical measurements

can be adjusted to find candidate measurements to eliminate critical ones. Here is a list of steps for the placement algorithm:

Step 1 Append the Jacobian matrix corresponding to the remaining measurements to the bottom of the original H matrix forming H'.

Step 2 Factorize the new Jacobian matrix H' as in equation (5.3) by disabling the row shifting of the rows corresponding to remaining measurements.

Step 3 The rows of L' corresponding to candidate measurements is labeled as matrix K_c (subscript c for candidates). Any non-zero entries of K_c in the columns of critical measurements are tagged as candidates to eliminate critical measurements.

Step 4 Optimize the selection of candidate measurements.

More than one measurement can be tagged as candidates to eliminate the same critical measurement. Also, a candidate can convert more than one critical measurement to redundant ones. In addition, each meter has an installation and maintenance cost associated with it. Therefore, an optimization scheme is required to minimize the cost and number of meters to be placed for handling all critical measurements. Binary integer programming can be used to solve the following optimization problem:

$$\begin{aligned} &\text{Minimize} && C^T \cdot X \\ &\text{Subject to} && A \cdot X \geq \hat{1} \end{aligned} \tag{5.9}$$

where

C is the cost vector associated with each candidate injection.

X is the binary vector defined as:

$$X(i) = \begin{cases} 1 & \text{if injection } i \text{ is a chosen candidate} \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

A is the constraint matrix whose rows and columns correspond to candidate injections and critical measurements respectively:

$$A_{ij} = \begin{cases} 1 & \text{if injection } j \text{ is a candidate for critical measurement } i \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

$\hat{1}$ is a vector of ones. The number of rows in $\hat{1}$ is equal to the number of critical measurements.

Example 5.2**:** Conventional measurement placement is illustrated using the system in Figure 5.1 from previous example. Note that there are two injections at buses 2 and 5, already installed in the system. So, the H matrix for all the remaining injections (at buses 1, 3, and 4) are attached to the bottom of the original H:

$$H' = \begin{array}{c} P_2 \\ P_5 \\ P_{13} \\ P_{45} \\ P_{35} \\ P_1 \\ P_3 \\ P_4 \end{array} \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & -1 & -1 & 2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & -1 \\ -1 & -1 & 0 & 0 \\ 0 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \end{bmatrix} \tag{5.12}$$

The factorization of H' and then transformation of L into L' yields:

$$L' = \begin{array}{c} P_2 \\ P_5 \\ P_{13} \\ P_{45} \\ P_{35} \\ P_1 \\ P_3 \\ P_4 \end{array} \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & -1.0 & 0 & -1.0 \\ -0.5 & -0.5 & 1.5 & -1.0 \\ 0 & -2.0 & -1.0 & -3.0 \\ -0.5 & 1.5 & -0.5 & 4.0 \end{bmatrix} \begin{array}{l} \leftarrow \text{critical} \\ \\ \leftarrow \text{critical} \\ \\ \leftarrow \text{redundant} \\ \leftarrow \text{candidate} \\ \leftarrow \text{candidate} \\ \leftarrow \text{candidate} \end{array} \tag{5.13}$$

The last 3 rows of L' correspond to candidates. The first and third columns are related to critical measurements. It turns out that candidate injections at buses 1 and 4 (P_1 and P_4) can convert both critical measurements, while P_3 is good for only one critical. Here is how the constraint matrix A looks like:

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{5.14}$$

There are 2 critical and 3 candidate measurements, therefore, A has 2 rows and 3 columns. Note that every element in the cost vector C is set equal to 1. This means all the candidate meters have the same installation cost. The solution implies that either P_1 or P_4 can be installed to eliminate both of the critical measurements. P_3 is cannot handle both of the critical measurements by itself. So, P_3 would not be in the optimal solution.

5.3.2 PMU Placement

The theory of measurement placement is the same for both conventional measurements and PMUs. The Jacobian matrix for PMUs is appended to the bottom of the original H. Then, H' is factorized. The transformation of lower factor of H is done by multiplying L by L_e^{-1} from right as in equation (5.4). The major difference of conventional measurements and PMUs comes in the optimization of candidate selection, where two rows of L' represent one PMU. An algorithm for PMU placement is described in [35]. The columns of the constraint matrix A needs to represent PMUs instead of rows of L'. This is achieved by:

$$A = R^T \cdot B^T \tag{5.15}$$

where

R is the absolute value of the entries of the critical columns of K_c (rows of L' corresponding to candidates).

$$R(i,j) = \begin{cases} |K_c(i,j)| & \text{if } K_c(i,j) \neq 0 \text{ and measurement } j \text{ is critical} \\ 0 & \text{otherwise} \end{cases} \tag{5.16}$$

B is the incidence matrix keeping track of the phasors a PMU is measuring:

$$B(i,j) = \begin{cases} 1 & \text{if PMU at bus } i \text{ includes the phasor } j \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

Note that more than one PMU can be installed at a bus. All the branches directly connected to a given bus are possible locations for PMUs. The system in Figure 5.1 is used to illustrate placement of PMUs. For example, here is how the measurement

Jacobian for all 3 PMUs installed at bus 4 is structured:

$$H_{PMU} = \begin{array}{c} \theta_4 \\ I_{42} \\ I_{43} \\ I_{45} \end{array} \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \tag{5.18}$$

The rows in the equation (5.18) represent 3 PMUs as shown in Table 5.1.

Table 5.1: Phasor measurements of all possible PMUs at bus 4

	PMU1	PMU2	PMU3
Voltage Phasor	θ_4	θ_4	θ_4
Current Phasor	I_{42}	I_{43}	I_{45}

Example 5.3**:** There are 5 buses and 6 branches in Figure 5.1. Therefore, one PMU can be installed on each side of a branch making the total number of candidate PMUs 12. Here is how the measurement Jacobian looks like, once all PMUs are

attached to the bottom of original H:

$$H' = \begin{array}{c} P_2 \\ P_5 \\ P_{13} \\ P_{45} \\ P_{35} \\ \theta_1 \\ I_{12} \\ I_{13} \\ \theta_2 \\ I_{21} \\ I_{24} \\ \theta_3 \\ I_{31} \\ I_{34} \\ I_{35} \\ \theta_4 \\ I_{42} \\ I_{43} \\ I_{45} \\ \theta_5 \\ I_{53} \\ I_{54} \end{array} \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & -1 & -1 & 2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{5.19}$$

First, H' is factorized to obtain L matrix. Then, L matrix is multiplied by L_e^{-1} as in equation (5.4) to get L'. The candidate part of L' and R is found to be:

$$K_c = \begin{array}{c} \theta_1 \\ I_{12} \\ I_{13} \\ \theta_2 \\ I_{21} \\ I_{24} \\ \theta_3 \\ I_{31} \\ I_{34} \\ I_{35} \\ \theta_4 \\ I_{42} \\ I_{43} \\ I_{45} \\ \theta_5 \\ I_{53} \\ I_{54} \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -0.5 & -0.5 & 0.5 & -1.0 \\ 0 & 0 & 1.0 & 0 \\ 0.5 & 0.5 & -0.5 & 1.0 \\ 0.5 & 0.5 & -0.5 & 1.0 \\ 0.5 & -0.5 & 0.5 & -1.0 \\ 0 & 0 & -1.0 & 0 \\ 0 & 0 & -1.0 & 0 \\ 0 & -1.0 & 0 & -2.0 \\ 0 & -1.0 & 0 & -1.0 \\ 0 & 1.0 & -1.0 & 2.0 \\ -0.5 & 0.5 & -0.5 & 1.0 \\ 0 & 1.0 & 0 & 2.0 \\ 0 & 0 & 0 & 1.0 \\ 0 & 1.0 & -1.0 & 1.0 \\ 0 & 1.0 & 0 & 1.0 \\ 0 & 0 & 0 & -1.0 \end{bmatrix} \Rightarrow R = \begin{bmatrix} 0 & 0 \\ 0.5 & 0.5 \\ 0 & 1.0 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0 & 1.0 \\ 0 & 1.0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1.0 \\ 0.5 & 0.5 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1.0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.20}$$

B matrix shows which phasors are measured by PMUs:

$$B = \begin{array}{c} \\ PMU1 \\ PMU2 \\ PMU3 \\ PMU4 \\ PMU5 \\ PMU6 \\ PMU7 \\ PMU8 \\ PMU9 \\ PMU10 \\ PMU11 \\ PMU12 \end{array} \begin{bmatrix} \theta_1 & I_{12} & I_{13} & \theta_2 & I_{21} & I_{24} & \theta_3 & I_{31} & I_{34} & I_{35} & \theta_4 & I_{42} & I_{43} & I_{45} & \theta_5 & I_{53} & I_{54} \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{5.21}$$

Calculating A for this example gives:

$$\begin{aligned} A &= R^T \cdot B^T \\ A &= \begin{bmatrix} 0.5 & 0 & 1.0 & 1.0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 1.0 & 1.0 & 1.0 & 2.0 & 1.0 & 1.0 & 1.5 & 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix} \end{aligned} \tag{5.22}$$

The constraint matrix (5.22) shows that there are 4 PMUs that can handle both of the critical measurements. So, both critical measurements can be eliminated by installing any one of these 4 PMUs. Those 4 PMUs measure the phasors shown in Table 5.2.

Table 5.2: 4 optimal candidate PMUs

	PMU1	PMU3	PMU4	PMU8
Voltage Phasor	θ_1	θ_2	θ_2	θ_4
Current Phasor	I_{12}	I_{21}	I_{24}	I_{42}

5.4 Simulation Results

Measurement placement algorithm described in the previous section is tested on larger systems. Two solutions are obtained for each simulation. First critical measurements are eliminated by using only conventional measurements (power injections). Then, the same system is solved by using PMUs. The measurement placement algorithms (PMU and conventional) developed are programmed using Matlab. The codes for these programs are attached in the appendix. All simulations are carried out with the following assumptions:

1. All buses that do not already have any injections are listed as remaining measurements. Likewise, all branches and buses that do not already have PMUs installed are included in the measurement Jacobian.

2. The installation and maintenance cost for all candidate measurements are assumed to be the same. However, any other cost data can be easily fed into the optimization function.

3. System observability is taken for granted. The algorithm checks for observability and then terminates if the system is unobservable to start with.

4. The decoupled linear $P - \theta$ model is used assuming that real and reactive power measurements come in pairs. Therefore, only real power injections and flows are considered.

5.4.1 IEEE 14 Bus System

There are 7 injections and 9 flows in the IEEE 14 bus system in Figure 5.2. The measurement configuration is checked for critical measurements, and 6 injections and 2 flows are reported as critical, which are listed in Table 5.3.

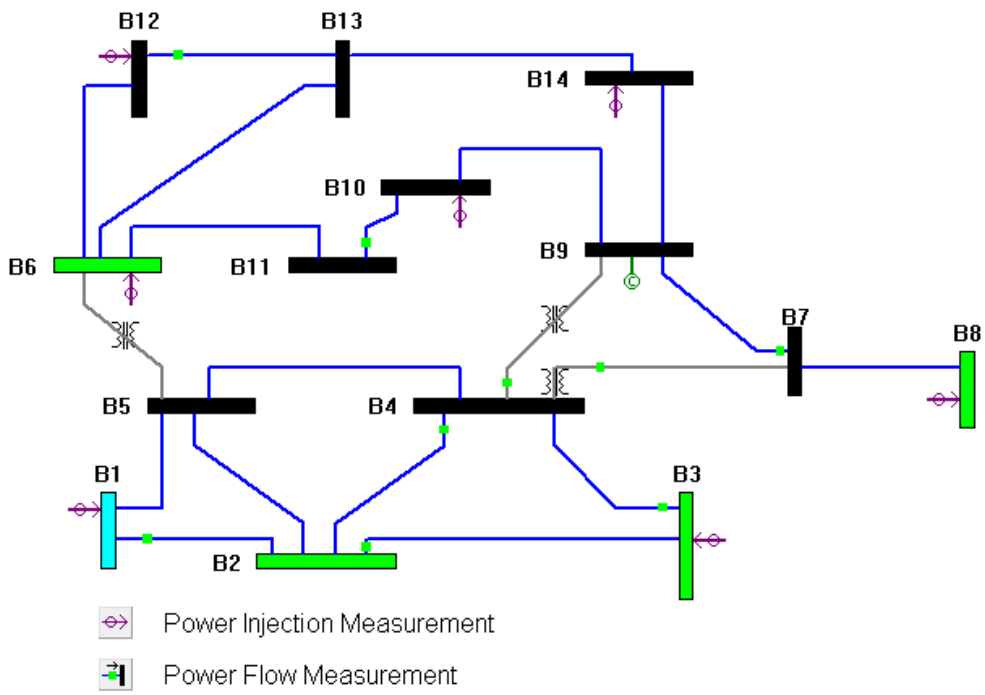

Figure 5.2: IEEE 14 bus system analyzed for critical measurements

Injection Solution

The optimal injection placement finds 2 candidate injections to eliminate all of the 8 critical measurements. These candidates are injections at buses 7 and 9.

PMU Solution

The result of optimal PMU placement shows that 2 PMUs are sufficient to transform all critical measurements. These PMUs need to be installed at buses 8 and 14. They measure voltage phasor of buses 8 and 14, and line phasors from buses 8 to 7 and 14 to 9. The candidate PMUs are shown in Table 5.4.

Table 5.3: Critical measurements in the IEEE 14 bus system

Critical Measurement	Measurement Type	Measurement Location
1	Injection	Bus 1
2	Injection	Bus 6
3	Injection	Bus 8
4	Injection	Bus 10
5	Injection	Bus 12
6	Injection	Bus 14
7	Flow	Bus 10 to Bus 11
8	Flow	Bus 12 to Bus 13

Table 5.4: Optimal placement of candidate PMUs for 14 bus system

	PMU1	PMU2
Voltage Phasor	θ_8	θ_{14}
Current Phasor	I_{8-7}	I_{14-9}

5.4.2 IEEE 30 Bus System

Figure 5.3 shows the measurement configuration of IEEE 30 bus system. There is a total of 31 measurements, 18 injections and 13 flows. It turns out that 26 of these measurements are critical, which are listed in Table 5.5.

Injection Solution

26 critical measurements in IEEE 30 bus system can be eliminated by installing 3 additional injections at buses 2, 12, and 25.

PMU Solution

The PMU solution gives 5 PMUs. With the addition of 5 PMUs no critical measurements will exist in the system. The list of candidate PMUs are given in Table 5.6.

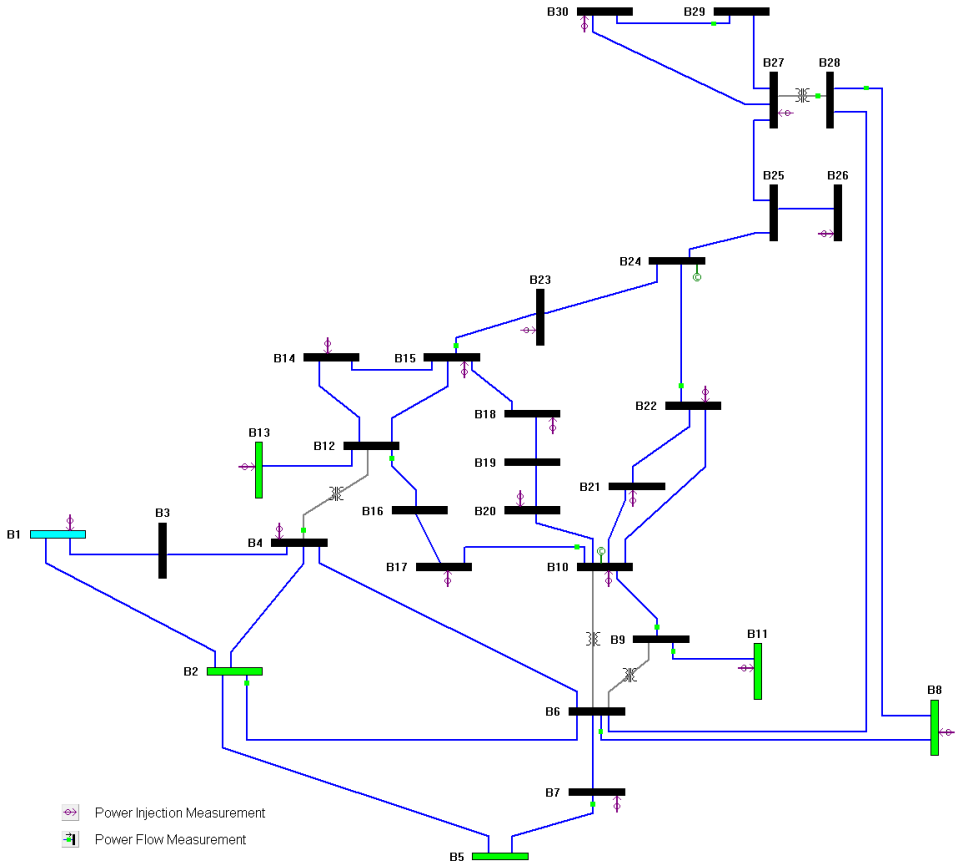

Figure 5.3: IEEE 30 bus system analyzed for critical measurements

5.5 Conclusions

In this chapter, critical measurements are shown to be the weak spots in a system. It is discussed that loss of a critical measurement will make system unobservable. Furthermore, it is not possible to detect the errors in critical measurements. Therefore, the existence of critical measurements in a system threatens the security and reliability of that system. In solution of this problem, identification of critical measurements

is explained first. Once critical measurements are identified, their elimination by placing additional meters are discussed. Optimization of the elimination process is accomplished by minimizing an installation and maintenance cost function. In the examples, since the cost of all the meters are assumed to be the same, the optimum solution gives the minimum number of measurements.

Placement of conventional measurements and PMUs are handled separately due to a slight difference in the optimization process. The simulations of IEEE 14 and 30 bus systems demonstrate that the optimal placement algorithm given in this chapter succeeds in bringing the system into a critical measurement free condition, thus improves system security and reliability.

Table 5.5: Critical measurements in the IEEE 30 bus system

Critical Measurement	Measurement Type	Measurement Location
1	Injection	Bus 1
2	Injection	Bus 4
3	Injection	Bus 7
4	Injection	Bus 10
5	Injection	Bus 13
6	Injection	Bus 14
7	Injection	Bus 15
8	Injection	Bus 17
9	Injection	Bus 18
10	Injection	Bus 20
11	Injection	Bus 21
12	Injection	Bus 22
13	Injection	Bus 23
14	Injection	Bus 26
15	Injection	Bus 27
16	Injection	Bus 30
17	Flow	Bus 4 to Bus 12
18	Flow	Bus 6 to Bus 2
19	Flow	Bus 7 to Bus 5
20	Flow	Bus 10 to Bus 9
21	Flow	Bus 12 to Bus 16
22	Flow	Bus 15 to Bus 23
23	Flow	Bus 17 to Bus 10
24	Flow	Bus 24 to Bus 22
25	Flow	Bus 27 to Bus 28
26	Flow	Bus 30 to Bus 29

Table 5.6: Optimal placement of candidate PMUs for 30 bus system

	PMU1	PMU2	PMU3	PMU4	PMU5
Voltage Phasor	θ_1	θ_5	θ_{13}	θ_{24}	θ_{26}
Current Phasor	I_{1-2}	I_{5-2}	I_{13-12}	I_{24-25}	I_{26-25}

Chapter 6

Conclusions

Power system operators use state estimation as a tool to evaluate the system conditions. In this thesis, an explanation of state estimation function is given followed by methods to improve state estimation results. These improvements are in the areas of observability analysis and bad data detection.

The first task in observability analysis is to find unobservable branches in the system. Identification of unobservable branches enables grouping the buses into observable islands. The second objective of the observability analysis is to find an optimal number of additional measurements to merge observable islands. An iterative process to gradually merge observable islands is developed. Conventional measurements and PMUs are considered in measurement placement algorithm. The measurement equations and Jacobian matrix are very sparse in power systems, and there is no matrix inversion in the placement algorithm. Therefore, the process of placing measurements is computationally fast and efficient. The simulations of IEEE 14 and 30 bus systems show that the algorithm finds minimum number of measurements, either conventional ones or PMUs, making the system observable.

An observable system does not necessarily imply a bad data free system. In order to detect errors in measurement data and improve state estimation results, redundancy in measurements is required. Since the detection of errors in the critical measurement data is not possible, the system becomes vulnerable to bad data. Furthermore, if any critical measurement is lost, then the system loses its observability. In this thesis, identification of critical measurements and their elimination are studied. If critical measurements are found, then they are eliminated by placing a minimum number of extra measurements, which are either conventional measurements or PMUs. Power injections and PMUs are considered as additional measurements to eliminate critical ones. The algorithm developed is tested on IEEE 14 and 30 bus systems. The set of additional meters found in simulations are able to eliminate critical measurements in the system.

The tools to improve state estimation described in this thesis can be used to gradually improve the state estimation performance when there is a limit on the number of meters to be added to the system. However, the ideal solution would be installing all the meters suggested by the algorithms, so that system operators can have a better control on their system.

Bibliography

[1] A.R. Bergen and V. Vittal, *Power Systems Analysis*, 2nd ed. New Jersey: Prentice Hall, 2000.

[2] T.E. Dy Liacco, "Real-Time Computer Control of Power Systems", *Proceedings of the IEEE*, vol. 62, no. 7, pp. 884-891, July 1974.

[3] A. Monticelli and F.F. Wu, "Network Observability: Theory", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-104, no. 5, pp. 1042-1048, May 1985.

[4] F.C. Schweppe and J. Wildes, "Power System Static-State Estimation, Part I: Exact Model", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 120-125, Jan. 1970.

[5] F.C. Schweppe, "Power System Static-State Estimation, Part III: Implementation", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 130-135, Jan. 1970.

[6] A. Abur and A.G. Expósito, *Power System State Estimation: Theory and Implementation*. New York: Marcel Dekker, 2004.

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

[8] F.C. Schweppe and D.B. Rom, "Power System Static-State Estimation, Part II: Approximate Model", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 125-130, Jan. 1970.

[9] G.R. Krumpholz, K.A. Clements, and P.W. Davis, "Power System Observability: A Practical Algorithm Using Network Topology", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-99, no. 4, pp. 1534-1542, July 1980.

[10] K.A. Clements, G.R. Krumpholz, and P.W. Davis, "Power System State Estimation Residual Analysis: An Algorithm Using Network Topology", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-100, no. 4, pp. 1779-1787, April 1981.

[11] K.A. Clements, G.R. Krumpholz, and P.W. Davis, "State Estimator Measurement System Reliability Evaluation - An Efficient Algorithm Based on Topological Observability Theory", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-101, no. 4, pp. 997-1004, April 1982.

[12] K.A. Clements, G.R. Krumpholz, and P.W. Davis, "Power System State Estimation with Measurement Deficiency: An Algorithm that Determines the Maximal Observable Subnetwork", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-101, no. 9, pp. 3044-3052, Sep. 1982.

[13] K.A. Clements, G.R. Krumpholz, and P.W. Davis, "Power System State Estimation with Measurement Deficiency: An Observability/Measurement Placement

Algorithm", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-102, no. 7, pp. 2012-2020, July 1983.

[14] R. Nucera and M. Gilles, "Observability Analysis: A New Topological Algorithm", *IEEE Trans. on Power Systems*, vol. 6, no. 2, pp. 466-473, May 1991.

[15] B. Stott and O. Alsac, "Fast Decoupled Load Flow", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-93, no. 3, pp. 859-867, May 1974.

[16] A. Garcia, A. Monticelli, and P. Abreu, "Fast Decoupled State Estimation and Bad Data Processing", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-98, no. 5, pp. 1645-1652, Sep. 1979.

[17] J.J. Allemong, L. Radu, and A.M. Sasson, "A Fast and Reliable State Estimation Algorithm for AEP's New Control Center", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-101, no. 3, pp. 993-944, April 1982.

[18] A. Monticelli and A. Garcia, "Fast Decoupled State Estimators", *IEEE Trans. on Power Apparatus and Systems*, vol. 5, no. 2, pp. 556-564, May 1990.

[19] A. Monticelli and F.F. Wu, "Network Observability: Identification of Observable Islands and Measurement Placement", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-104, no. 5, pp. 1035-1041, May 1985.

[20] B. Gou and A. Abur, "A Direct Numerical Method for Observability Analysis", *IEEE Trans. on Power Systems*, vol. 15, no. 2, pp. 625-630, May 2000.

[21] B. Gou and A. Abur, "An Improved Measurement Placement Algorithm for Network Observability", *IEEE Trans. on Power Systems*, vol. 16, no. 4, pp. 819-824, Nov. 2001.

[22] F.H. Magnago and A. Abur, "A Unified Approach to Robust Meter Placement Against Loss of Measurements and Branch Outages", *IEEE Trans. on Power Systems*, vol. 15, no. 3, pp. 945-949, Aug. 2000.

[23] A.G. Phadke, J.S. Thorp, and K.J. Karimi, "State Estimation with Phasor Measurements", *IEEE Trans. on Power Systems*, vol. 1, no. 1, pp. 233-241, Feb. 1986.

[24] T.L. Baldwin, L. Mili, M.B. Boisen, and R. Adapa, "Power System Observability with Minimal Phasor Measurement Placement", *IEEE Trans. on Power Systems*, vol. 8, no. 2, pp. 707-715, May 1993.

[25] R. Zivanovic and C. Cairns, "Implementation of PMU Technology in State Estimation: An Overview", in *Proc. IEEE* 4th *AFRICON*, vol. 2, pp. 1006-1011, Sep. 1996.

[26] H.M. Merrill and F.C. Schweppe, "Bad Data Suppression in POwer System Static Estimation", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-90, no. 6, pp. 2718-2725, Nov. 1971.

[27] A. Monticelli and A. Garcia, "Reliable Bad Data Processing for Real-Time State Estimation", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-102, no. 3, pp. 1126-1139, May 1983.

[28] L. Mili, Th. Van Cutsem, and M. Ribbens Pavella, "Hypothesis Testing Identification: A New Method for Bad Data Analysis in Power System State Estimation", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-103, no. 11, pp. 3239-3252, Nov. 1984.

[29] L. Mili and Th. Van Cutsem, "Implementation of HTI Method in Power System State Estimation", *IEEE Trans. on Power Systems*, vol. 3, no. 3, pp. 887-893, Aug. 1988.

[30] E. Handschin, F.C. Schweppe, J. Kohlas, and A. Fiechter, "Bad Data Analysis for Power System State Estimation", *IEEE Trans. on Apparatus and Systems*, vol. PAS-94, no. 2, pp. 329-337, March 1975.

[31] Xiang Nian-de, Wang Shi-ying, and Yu Er-keng, "A New Approach for Detection and Identification of Multiple Bad Data in Power System State Estimation", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-101, no. 2, pp. 454-462, Feb. 1982.

[32] M.B Do Coutto Filho, J.C Stacchini de Souza, F.M.F. de Oliveira, and M.T. Schilling, "Identifying Critical Measurements & Sets for Power System State Estimation, in *Power Tech Proceedings, 2001 IEEE Porto*, vol.3 , pp.6, Sep. 2001.

[33] A. Simoes-Costa, T.S. Piazza, and A. Mandel, "Qualitative Methods to Solve Qualitative Problems in Power System State Estimation", *IEEE Trans. on Power Systems*, vol. 5, no. 3, pp. 941-949, Aug. 1990.

[34] A.G. Exposito and A. Abur, "Generalized Observability Analysis and Measurement Classification", *IEEE Trans. on Power Systems*, vol. 13, no. 3, pp. 1090-1096, Aug. 1998.

[35] J. Chen and A. Abur, "Placement of PMUs to Enable Bad Data Detection in State Estimation", *IEEE Trans. on Power Systems*, vol. 21, no. 4, pp. 1608-1615, Nov. 2006.

[36] B. Xu and A. Abur, "Observability Analysis and Measurement Placement for Systems with PMUs", in *Proc. IEEE Power Eng. Soc. Power Systems Conf. Expo.*, New York, Oct. 10-13, 2004.

Appendix A

Measurement Placement Algorithm for Observability

The Matlab code for the measurement placement algorithm in Chapter 4 is given in this section. The candidates chosen by this algorithm can be power injections or voltage phasors (PMUs). The algorithm given in this section is the power injection version of the placement algorithm. The internal functions used in this algorithm are given in the last appendix section.

```
clear all;

% This script checks observability. If the system turns out to be
% unobservable, the algorithm places power injections iteratively until
% observability is restored.

% input files
measFile='..\..\..\PET_full\PET\mergeIslPaper\30bus\measure.dat';
```

```
pfinputFile='..\..\..\PET_full\PET\mergeIslPaper\30bus\pfinput.dat';

% NoBus = 14;   %number of bus
NoBus = 30;
% NoBus = 118;
% NoBus = 6;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[Y NumBran Branches ext_busPET int_busPET] = readAndBuildY(pfinputFile,NoBus);
L = buildL2(Y, NoBus);

[Pinj Pflow] = readMeas(measFile,NoBus);
Pinj = int_busPET(Pinj);
Pflow(:,1) = int_busPET(Pflow(:,1));
Pflow(:,2) = int_busPET(Pflow(:,2));

check0 = length(Pinj);
check = length(Pinj)-1;
cntDummy = 0;
while check < length(Pinj)
    check = length(Pinj);

    [H Hall candH rHall cHall origMeas candMeas]=buildH2(Y,L,Pinj,Pflow,NoBus);
    [rH cH] = size(H);
```

```
[Laa U P] = lu(H);
[ar bc] = size(Laa);
[cr dc] = size(U);
if bc < cH
    Laa1 = zeros(ar,cH);
    Laa1(1:ar,1:bc) = Laa;
    Laa = Laa1;
end
if cr < cH % if the number of measurements is less than NoBus - 1.
    U1 = zeros(cH,dc);
    U1(1:cr,1:dc) = U;
    U = U1;
end
cnt = 0;
for it = 1:cH
    if abs(U(it,it)) < 1e-6
        cnt = cnt+1;
        zeroPivot(cnt,1) = it;
    end
end

for it1 = 1:rH
    shift(it1,1)=it1;
end
order = P*shift;
```

```
H1 = P*H;
[er fc] = size(H1);
if er < cH % if the number of measurements is less than NoBus - 1.
    H2 = zeros(cH,fc);
    H2(1:er,1:fc) = H1;
    H1 = H2;
end

if cnt > 0
    for it=1:length(zeroPivot)
        H1(zeroPivot(it),zeroPivot(it))=H1(zeroPivot(it),zeroPivot(it))+1;
    end
end
Hnosh = [H1
        candH];

[Lall Uall Pall] = lu(Hnosh,0);

cnt1 = 0;
for it2 = 1:cH
    if abs(Uall(it2,it2)) < 1e-6
        cnt1 = cnt1+1;
        zeroPivot1(cnt1,1) = it2;
    end
end
```

```
[gr hc] = size(Hnosh);


if er >= cH
    for it3 = 1:rHall
        shift1(it3,1)=it3;
    end
    order1 = Pall*shift1;
elseif er < cH % if the number of measurements is less than NoBus - 1.
    for it3 = 1:gr
        shift1(it3,1)=it3;
    end
    order1 = Pall*shift1;
end



if cnt > 0
    if er >= cH
        list = Lall(rH+1:end,zeroPivot);
    elseif er < cH % if the number of measurements is less than NoBus - 1.
        list = Lall(cH+1:end,zeroPivot);
    end
    c = find(abs(list)>1e8);
    e = find(abs(Lall)>1e8);
```

```
if ~isempty(c)
    disp('Too high values in Lall, numerical errors');
    [rc cc] = size(list);
    d = ceil(c/rc);
    cnt4=0;
end

lencandMeas = length(candMeas);
lenzeroPivot = length(zeroPivot);
dummy = ones(1,lencandMeas);
A = sparse(lenzeroPivot,lencandMeas);
b = ones(lenzeroPivot,1);

if lenzeroPivot == 1
    cntDummy = cntDummy + 1;
    A = [A;dummy]; % this is required to fix matlab bintprog error
    b = [b;1]; % this is required to fix matlab bintprog error
end
C = ones(lencandMeas,1);
for it = 1:lenzeroPivot
    f1 = find(abs(list(:,it))>1e-4);
    A(it,f1) = 1;
end

[X Cval] = bintprog(C,-A,-b);
```

```
        list1 = full(list);
        cM = candMeas(find(X~=0));
        Pinj = [Pinj;cM];
    end
    zeroPivot = [];
    zeroPivot1 = [];
    shift1 = [];
end
candidateMeasurements = ext_busPET(Pinj(check0+1:check,1))
```

Appendix B

Measurement Placement Algorithm for Critical Measurement Elimination

The Matlab code for the measurement placement algorithm in Chapter 5 is given in this section. Note that this algorithm has two modes. In the first one, candidate measurements are chosen from only power injections. In the second mode, candidates are PMUs. The second version of the algorithm, which places PMUs, is displayed in this section. The internal functions used in this algorithm are given in the last appendix section.

This algorithm identifies critical measurements and then finds candidate measurements to eliminate those critical ones.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Form H, find critical measurements, and choose an optimal set of
% candidates to eliminate critical measurements.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

% These are the input files to the algorithm
measFile='..\..\..\PET_full\PET\thesis\MeasPlaceBadDataChap\5bus\measure.dat';
pfinputFile='..\..\..\PET_full\PET\thesis\MeasPlaceBadDataChap\5bus\pfinput.dat';

% Number of buses
NoBus = 5;
% NoBus = 14;
% NoBus = 30;
% NoBus = 118;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[Y NumBran Crit_int_bus Crit_ext_bus] = readAndBuildY(pfinputFile,NoBus);

L = buildL2(Y, NoBus);

[Pinj Pflow] = readMeas(measFile,NoBus);
Pinj = Crit_int_bus(Pinj);
Pflow(:,1) = Crit_int_bus(Pflow(:,1));
Pflow(:,2) = Crit_int_bus(Pflow(:,2));

PMUs = textread('pmu14.dat');
PMUs = Crit_int_bus(PMUs);
```

```
[H Hall candH rHall cHall origPmu candPmu] = buildHPMU(Y,L,Pinj,Pflow,NoBus,PMUs);
[rH cH] = size(H);

[LAA Ub P] = lu(H); % triangular factorization

% check for observability
unobserve=0;
if rH < NoBus-1
    unobserve=1;
else
    for i=1:NoBus-1
        if abs(Ub(i,i)) < 1e-6
           unobserve = 1;
           sprintf('zero at diagonal %d',i)
           break;
        end
   end
end

cnt = 0;
for i=1:NoBus-1
    if abs(Ub(i,i)) < 1e-6
        cnt = cnt+1; zeroDiag(cnt) = i;
    end
```

```
end

if unobserve == 1
    sprintf(' the system is unobservable')
end

for it8 = 1:rH
    order(it8) = it8;
end

row_shifting = P*order'; % the new order of rows after permuation matrix
Lb = LAA(1:NoBus-1,:);
Kb = LAA/(Lb);

% This part indentifies the critical measurements
LR = Kb(NoBus:rH,:);
cnt1 = 0;
for it2 = 1:cH
    a = find(abs(LR(:,it2)) > 1e-6);
    if isempty(a)
        cnt1 = cnt1 + 1;
        c(cnt1,1) = it2;
        critMeasLR(cnt1,1) = row_shifting(it2);
    end
end
```

```
if cnt1 > 0
    critMeasLR = sortrows(critMeasLR); % these are the rows of H,
                                       % corresponding to measurements
else
    disp('No critical measurements');
    return;
end

H1 = P*H;
Hnosh = [H1;
        candH];
[Lall Uall Pall] = lu(Hnosh,0); % factorization without row pivoting
Lcand = Lall(rH+1:end,:);
Kcand = Lcand/Lb;
Kcand1 = Kcand(:,c); % these are the columns corresponding to crit. meas.

[Re Be pmuBus] = compactPmu(c,Kcand1,NoBus,L); % preparation of selection
F = Re'*Be';

cnt3 = 0;
[rc cc] = size(candH);

b = ones(cnt1,1);
NoPmu = size(Be,1);
```

```
candPmu = 1:NoPmu;
candPmu = candPmu';
C = ones(NoPmu,1);

[X Cval] = bintprog(C,-F,-b);
cM = candPmu(find(X~=0));


% This part prints chosen PMUs
[chosenPmu] = findPmu(cM,L,pmuBus);
chosenPmu(:,1) = Crit_ext_bus(chosenPmu(:,1));
chosenPmu(:,2) = Crit_ext_bus(chosenPmu(:,2));
chosenPmu

% This part prints critical measurements
[r c] = size(Pinj);
[r1 c1] = size(Pflow);
a = find(critMeasLR <= r);
critPinj = Crit_ext_bus(Pinj(critMeasLR(a)));
critPinj = sortrows(critPinj);

b = find(critMeasLR > r);
[r2 c2] = size(b);
critPflow = zeros(r2,2);
critPflow(:,1) = Crit_ext_bus(Pflow(critMeasLR(b)-r,1));
```

```
critPflow(:,2) = Crit_ext_bus(Pflow(critMeasLR(b)-r,2));
critPflow = sortrows(critPflow, [1 2]);
```

Appendix C

Functions Used in the Measurement Placement Algorithms

The functions used in the placement algorithm are given in this section. Some of these functions are common in both placement algorithms.

C.1 Read Network Parameters and Calculate Y matrix

```
function [y NoBran int_bus ext_bus] = readAndBuildY(pfinputData,nobus)
% read network data and creates Y matrix, assuming j1.0 for all branches

fnet=fopen(pfinputData,'r');
line=fgetl(fnet); % skip first 2 lines of input
```

```
line=fgetl(fnet);
iter2 = 0;
while 1
    line=fgetl(fnet);
    iter2=iter2+1;
    if line(1:4)=='BRAN'
        break;
    end
    buses(iter2,1) = str2num(line(1:4));
end
buses(end,:) = []; % this is the entry corresponding to -999

for it=1:nobus
    ext_bus(it,1) = buses(it,1); % assigns external bus numbers
    int_bus(ext_bus(it),1) = it; % internal bus numbers
end

iter=0;
iter1 = 0;
while 1
    line=fgetl(fnet);
    if line(1:4)=='-999'
        break;
    end
    iter=iter+1;
```

```
    cir(iter,1) = str2num(line(17)); %circuit number
        iter1 = iter1+1;
        FromBus(iter1,1)=str2num(line(1:4));
        ToBus(iter1,1)=str2num(line(6:9));
end
NoBran=iter1;

FromBus = int_bus(FromBus);
ToBus = int_bus(ToBus);

% create Y matrix with Xi = j1.0 and no shunt cap.
ys = 1;
[rowBr col1] = size(FromBus);
for it5 = 1:rowBr
    Yi(ys) = FromBus(it5,1); Yj(ys) = ToBus(it5,1); Yv(ys) = -1; ys = ys + 1;
    Yi(ys) = ToBus(it5,1); Yj(ys) = FromBus(it5,1); Yv(ys) = -1; ys = ys + 1;
    Yi(ys) = FromBus(it5,1); Yj(ys) = FromBus(it5,1); Yv(ys) = 1; ys = ys + 1;
    Yi(ys) = ToBus(it5,1); Yj(ys) = ToBus(it5,1); Yv(ys) = 1; ys = ys + 1;
end
y = sparse(Yi,Yj,Yv,nobus,nobus);
disp('created Y matrix');
ST = fclose(fnet);

function L = buildL2(Y, NoBus)

L=zeros(NoBus,1); % Define L matrix. first column -> bus number,
```

```
                  % successive columns are neighbor buses
for p=1:NoBus
    L(p,1)=p; w=2;
    nz = find(Y(p,:)~=0);
    [r iter] = size(nz);
    for q=1:iter
        ne = nz(q);
        if p~=ne %if there's any branch between two lines
            L(p,w)=ne; w=w+1;
        end
    end
end
disp('created L matrix');
```

C.2 Read Measurement Data

```
function [Pinj Pflow] = readMeas(measData,NoBus)

% read measurement
fmeasure=fopen(measData,'r');

% read voltage measurement
line=fgetl(fmeasure);
  if length(line)==4
      num_vol=str2num(line(1:4));
```

```
  end

for i=1:num_vol
    line=fgetl(fmeasure);
    VoltMag(i,1)=str2num(line(1:3));
    VoltMagValue(i,1)=str2num(line(8:15));
end

% read real inject measurement
line=fgetl(fmeasure);
  if length(line)==4
      num_real_inj=str2num(line(1:4));
  end

for i=1:num_real_inj
    line=fgetl(fmeasure);
    Pinj(i,1)=str2num(line(1:3));
    PinjValue(i,1)=str2num(line(8:15));
end

% read reactive inject measurement
line=fgetl(fmeasure);
  if length(line)==4
      num_reac_inj=str2num(line(1:4));
  end
```

```
for i=1:num_reac_inj
    line=fgetl(fmeasure);
    QInj(i,1)=str2num(line(1:3));
    QinjValue(i,1)=str2num(line(8:15));
end

% read real power flow measurement
line=fgetl(fmeasure);
  if length(line)==4
      num_real=str2num(line(1:4));
  end

for i=1:num_real
    line=fgetl(fmeasure);
    Pflow(i,1)=str2num(line(1:4));
    Pflow(i,2)=str2num(line(5:8));
    PflowValue(i,1)=str2num(line(13:20));
end

% read reactive power measurement
line=fgetl(fmeasure);
  if length(line)==4
      num_reac=str2num(line(1:4));
  end
```

```
for i=1:num_reac
    line=fgetl(fmeasure);
    Qflow(i,1)=str2num(line(1:4));
    Qflow(i,2)=str2num(line(5:8));
    QflowValue(i,1)=str2num(line(13:20));
end

ST = fclose(fmeasure);
```

C.3 Build Measurement Jacobian

```
function [H Hall candH rHall cHall origMeas candMeas]
= buildH2(Y,L,pinj,pflow,nobus)

% given Y L Pinj and Pflow this function calculates H (original meas.) and
% Hall (orig meas + cand meas)

[rPinj cPinj] = size(pinj);
[rPflow cPflow] = size(pflow);
[origMeas candMeas] = candidatemeas1(pinj,nobus,rPinj);

hs = 0;
for it7 = 1:rPinj
    bus = pinj(it7,1);
```

```
    for it8 = 2:nnz(L(bus,:))
         hs = hs + 1; Hi(hs) = it7; Hj(hs) = L(bus,it8); Hv(hs) = -1;
         hs = hs + 1; Hi(hs) = it7; Hj(hs) = bus; Hv(hs) = 1;
    end
end

cnt3 = 0;
for it9 = it7+1:it7+rPflow
    cnt3 = cnt3+1;
    FB1 = pflow(cnt3,1); TB1 = pflow(cnt3,2);
    hs = hs + 1; Hi(hs) = it9; Hj(hs) = FB1; Hv(hs) = 1;
    hs = hs + 1; Hi(hs) = it9; Hj(hs) = TB1; Hv(hs) = -1;
end
Hrow = it9;
H = sparse(Hi, Hj, Hv, Hrow, nobus);
clear Hi Hj Hv;
H(:,1) = []; % take out one column of H to form a full rank square matrix

[rCand NoCandPinj] = size(candMeas);
cs = 0;
for it8 = 1:rCand
    bus1 = candMeas(it8);
    for it9 = 2:nnz(L(bus1,:))
        cs = cs + 1;
        candHi(cs) = it8; candHj(cs) = L(bus1,it9); candHv(cs) = -1;
```

```
        cs = cs + 1;
        candHi(cs) = it8; candHj(cs) = bus1; candHv(cs) = 1;
    end
end

candH = sparse(candHi,candHj,candHv,rCand,nobus);
candH(:,1) = []; % get rid of the first col.

Hall = [H
        candH];

[rHall cHall] = size(Hall);
disp('build H matrix');

function [H Hall candHpmu rHall cHall origPMU candPMU]
= buildHPMU(Y,L,pinj,pflow,nobus,pmus)

% given Y L Pinj and Pflow this function calculates H (original meas.) and
% Hall (orig meas + cand meas)

[rPinj cPinj] = size(pinj);
[rPflow cPflow] = size(pflow);
[rPmu cPmu] = size(pmus);
[origPMU candPMU] = candidatepmus(pmus,nobus,rPmu);

hs = 0;
```

```
for it7 = 1:rPinj
    bus = pinj(it7,1);
    for it8 = 2:nnz(L(bus,:))
         hs = hs + 1; Hi(hs) = it7; Hj(hs) = L(bus,it8); Hv(hs) = -1;
         hs = hs + 1; Hi(hs) = it7; Hj(hs) = bus; Hv(hs) = 1;
    end
end

cnt3 = 0;
for it9 = it7+1:it7+rPflow
    cnt3 = cnt3+1;
    FB1 = pflow(cnt3,1); TB1 = pflow(cnt3,2);
    hs = hs + 1; Hi(hs) = it9; Hj(hs) = FB1; Hv(hs) = 1;
    hs = hs + 1; Hi(hs) = it9; Hj(hs) = TB1; Hv(hs) = -1;
end

if rPmu > 0
    cnt4 = 0;
    for it10 = it9+1:it9+rPmu
        cnt4 = cnt4+1;
        hs = hs + 1;
        Hi(hs) = it10; Hj(hs) = pmus(cnt4); Hv(hs) = 1;
    end
    Hrow = it10;
else
```

```
    Hrow = it9;
end

H = sparse(Hi, Hj, Hv, Hrow, nobus);
clear Hi Hj Hv;
H(:,1) = []; % take out one column of H to form a full rank square matrix

[rCandPmu NoCandPmu] = size(candPMU);
cs = 0;
row = 0;
for it8 = 1:rCandPmu
    cs = cs + 1;
    row = row + 1;
    candHpmui(cs) = row; candHpmuj(cs) = candPMU(it8); candHpmuv(cs) = 1;
    NZ = nnz(L(candPMU(it8),:)); % neighbor buses
    for it11 = 2:NZ
        row = row + 1;
        cs = cs+1;
        candHpmui(cs) = row; candHpmuj(cs) = candPMU(it8); candHpmuv(cs) = 1;
        cs = cs+1;
        candHpmui(cs)=row; candHpmuj(cs)=L(candPMU(it8),it11); candHpmuv(cs)=-1;
    end
end

candHpmu = sparse(candHpmui,candHpmuj,candHpmuv,row,nobus);
```

```
candHpmu(:,1) = [];

Hall = [H
        candHpmu];

[rHall cHall] = size(Hall);
disp('build H matrix');
```

C.4 Matrices Required for PMU Optimization

```
function [R B PMUbus] = compactPmu(critCol,kcand1,nobus,N)

% these are the R and B matrices for PMU optimization

rcc = size(critCol,1);
rch = size(kcand1,1);
colB = nnz(N(:,2:end));

B = sparse(colB,rch);
R = sparse(rch,rcc);

col = 1;
row = 0;
cnt = 0;
for it = 1:nobus
```

```
    NZ = nnz(N(it,:));
    for it1 = 2:NZ
        row = row+1; cnt = cnt + 1;
        B(row,col) = 1;
        B(row,col+cnt) = 1;
        PMUbus(row,1) = it;
    end
    col = col+NZ;
    NZ = [];
    cnt = 0;
end

makeOnes = find(abs(kcand1) > 1e-6);
R(makeOnes) = abs(kcand1(makeOnes));
```

C.5 Display Chosen PMUs

```
function [ChosenPMU] = findPmu(cm,N,PMUbus)

cMr = size(cm,1);

for it = 1:cMr
    a = PMUbus(cm(it,1));
    b = find(PMUbus==a);
    c = cm(it,1) - b(1,1);
```

```
    ChosenPMU(it,1) = a;
    ChosenPMU(it,2) = N(a,c+2);
end
```