Optimal Placement of Phasor Measurement Units using the Advanced Matrix Manipulation Algorithm

Prepared by:
Abdul-Aziz Fish

Department of Electrical Engineering
University of Cape Town

Prepared for:
Dr S. Chowdhury &
A/Prof S.P. Chowdhury

Department of Electrical Engineering
University of Cape Town

February 2013

Submitted to the Department of Electrical Engineering at the University of Cape Town for fulfilment of the academic requirements for a Master of Science degree in Electrical Engineering

Key Words: Phasor measurement Unit, Advanced Matrix manipulation, Power system, Optimal placement, full observability

Declaration

1. *I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.*
2. *I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this thesis report from the work(s) of other people has been attributed, and has been cited and referenced.*
3. *This thesis report is my own work.*
4. *I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.*

Name: **Abdul-Aziz Fish**

Signature: **Date:** 11 February 2013

Acknowledgements

I firstly wish to thank God the Almighty, who has given me everything I have, including all my ideas and guided me to the people mentioned below who have helped me in completing my thesis.

I would like to thank the following people for assisting me in whichever way I needed, for the completion of my thesis:

- Dr S. Chowdhury and A/Prof S.P. Chowdhury, my supervisors who gave me advice whenever I needed and who also organised outside assistance for thorough understanding and application of the topic
- My parents and sisters who supported me through my progress and always reminded me not to give up
- John Fadiran, who inspired the pursuit of the phasing algorithms
- Rustum Emjedi, who helped me to clarify a few critical aspects as well as sourcing me with journals to be used for the Literature review
- Mohamed Sideeq Mansura, my friend who helped me with Matlab syntax
- Armien Edwards, who gave me guidance about large networks
- All the academics whose articles I referenced throughout the thesis
- All my family, friends, neighbours and colleagues who wished me well for the completion of my project.

Terms of reference

The thesis titled "Optimal Placement of Phasor Measurement Units using the Advanced Matrix Manipulation Algorithm" intends to develop a Matrix Manipulation Algorithm for optimal placement of Phasor Measurement Units (PMUs) for the purpose of State Estimation of electrical power systems. This algorithm is intended to optimally place PMUs for ensuring full observability for the system. The following areas are expected to be covered in this thesis:

1. Review of SCADA systems
2. Review of PMU architecture and installation
3. Review of how PMU systems are integrated into SCADA systems
4. Review of State Estimation techniques
5. Review of improvement in State Estimation using PMUs
6. Analysis of different optimal PMU placement algorithms
7. Development of the Advanced Matrix Manipulation algorithm for optimal PMU placement with features such as Phasing of PMU Placement, Bad Data Detection Aid and Islanding Detection functionalities
8. Application of the proposed algorithm to IEEE power system test networks and typical South African Networks
9. Comparison of the proposed algorithm with the Integer Linear Programming (ILP) algorithm with respect to their performance for all the systems mentioned in item 9
10. Discussion of results
11. Conclusion and discussion of future research

Abstract

This thesis investigates the problem of the Optimal Placement scheme of Phasor Measurement Units in electrical power systems for State Estimation to facilitate improved monitoring and control of the system parameters.

Research

The research work done for this thesis begins with review of Supervisory Control and Data Acquisition systems (SCADA). SCADA-based systems are currently employed for condition monitoring and control of industrial and utility electrical power systems. For utility power networks, the main problem with voltage and current phasor data captured by SCADA systems is that they are not synchronised with respect to each other in a present-time or Real-time framework. This implies that both magnitude and phase angle of the measured phasors tend to get affected by slow data flow provided by SCADA to the points of utilization and also by differences in time instants of data capture. These factors inhibit the efficiency and quality of the power system monitoring and control.
"Phasor Measurement Unit" (PMU) is a relatively new technology that, when employed in power networks, offers real-time synchronised measurements of the voltages at buses and currents along the lines that connect them. This is accomplished by using a GPS based monitoring system which facilitates time synchronisation of measurements and unlike SCADA, makes the measured data available in Real-Time format. SCADA is not able to provide Real-time data due to the low speeds at which RTUs (Remote Terminal Units) provide data. Availability of time-stamped phasor measurements makes PMUs preferable for power system monitoring and control applications such as State Estimation, Instability Prediction Analysis, Real-time Monitoring of the system conditions, Islanding Detection, System Restoration and Bad Data Detection.

PMUs intend to increase the efficiency of power system operations by their data presentation compared to the current SCADA system data management. The collected system data is used in a process called State Estimation which intends to provide a representation of the present State of the system parameters of the entire network, aslo refered to as the State Vector. If the data is old and doubtful, it will cause the generalised State to be false and unreliable. This is dangerous since the state of the network is used to maintain the smooth operation of the electric power system. Therefore it is very important to have access to Real-time data, as provided by PMUs.

Since PMUs are expensive, their procurement and installation needs to be planned both in terms of economy and utility. Usually utilities like to see that the power network becomes fully observable with minimum number of PMUs placed at strategic buses. Where full observability refers to all the buses in the networked are actively monitored. Thus the problem of optimal placement of PMUs is formulated as an optimization problem where the number of PMUs is minimized subject to complete system observability. Several optimization algorithms are reported to have been used to address optimal PMU placement and they all aim at placing minimum number of PMUs at strategic buses within the network. A few such algorithms reviewed are Binary Particle Swarm algorithm, Immunity Genetic Algorithm, Binary Search Algorithm and the Integer Linear Programming (ILP) Algorithm. Each algorithm has its strengths and disadvantages which are related to the logic behind its optimisation approach.

After researching these algorithms, careful consideration was given to what a good PMU placement method needs. A new algorithm was developed in this thesis which intends to minimise the amount of PMUs required for full system observability and either to provide similar or better results than the researched algorithms. This algorithm is called the Advanced Matrix Manipulation (AMM) Algorithm.
The AMM algorithm uses a simple Boolean Algebra approach to converge to the final solution. It uses the AND function collectively on all of the connectivity matrix rows. As with all binary ANDing formulations, a minimum binary vector results which compensates for all the binary inputs. The idea behind it is quite simple and is used in many other applications today.

Methodology

The proposed AMM Algorithm is developed in Matlab. The algorithm is developed first as a program setup for input of a connectivity Matrix and output of optimal bus locations for PMU placement. Then additional functionalities are added to the program to address issues such as Phasing of PMU installation, Bad data detection and Islanding detection.
Phasing the installation process occurs in the situation when there are monetary issues. "Phasing the PMU placement" feature of the algorithm advises what the best locations will be for specified number of PMUs that could currently be afforded. This feature offers the program user a range of input flexibilities in terms of the predefined number and location preferences of PMUs.

The Bad data detection aid feature is included to ensure a certain level of measurement redundancy (multiple measurements of the same bus) when calculating the number of PMUs for complete observability of the network. The level of measurement redundancy can be chosen before the calculation takes place. The intention is to improve State Estimation Bad Data Detection methods, by providing the required measurement redundancy of the data.

Islanding detection functionality helps to strategically place the PMUs to ensure system observability even if intentional or unintentional islanding occurs in the power network. The PMUs are placed to ensure observability of the network even if the Islands do not form.

Results

The AMM algorithm is tested on IEEE 14, 30, 57 and a typical 1009 bus South African utility network and compared with well-reported Integer Linear Programming (ILP) algorithm in terms of the locations and amount of required PMUs calculated.

There is an option of applying the AMM algorithm without its aforesaid additional features. The AMM algorithm without its additional features is referred to as the standard AMM algorithm in the. Comparison of results for the Standard AMM and ILP algorithms when including zero-injection bus compensation are provided in Table 1 below:

Table 1: Standard AMM and ILP results

	IEEE 14 Bus System	IEEE 30 Bus System	IEEE 57 Bus System	1009 Bus SA Network
Algorithm	Minimum No. of PMUs	Minimum No. of PMUs	Minimum No. of PMUs	Minimum No. of PMUs
AMM	3	6	15	187
ILP	3	7	12	226

The Bad data detection algorithm was applied to the IEEE 14, 30 and 57 bus systems. For the redundancy level of 2 PMUs observing each bus the following results were obtained:

1. IEEE 14 bus requires a minimum of 9 PMUs
2. IEEE 30 bus requires a minimum of 16 PMUs
3. IEEE 57 bus requires a minimum of 29 PMUs
4. 1009 bus requires a minimum of 508 PMUs

AMM includes two separate phasing installation algorithms:

1. For inside input – where there are no critical buses in the system and the client does not necessarily wish for any buses to specifically include a PMU
2. For outside input – where there are critical buses and the client wishes to place PMUs at specific user-defined buses

One example of application of Inside Input algorithm is presented for the IEEE 30 bus system where in the first phase it places 2 PMUs at buses 2 and 10. In the second phase, the algorithm determines that 4 more PMUs are necessary and these are to be placed at buses 1, 12, 15, 20.

Another example of Outside Input algorithm is presented again for the IEEE 30 bus system where in the first phase 4 PMUs are placed at 4 client-specified buses 3, 12, 17 and 25. In the second phase, the algorithm calculates that 4 more PMUs are necessary and are to be placed at buses 2, 10, 15 and 18.

Recommendations

The results show that the proposed AMM Algorithm is capable of handling both small and large power networks. This algorithm is recommended especially for large power networks where there are zero-injection buses connected to other zero-injection buses, since the AMM has a special zero-injection bus compensator which reduces the required amount of PMUs further (in this situation) than other algorithms.

When phasing of PMU installations is a client requirement, it is recommended that either of the phasing algorithms be used, specific of course to what the situation might be (inside or outside input).

It is seen that for Bad data detection methods employed in State Estimation methods, the proposed algorithm places the PMUs at buses to provide the desired level of measurement redundancy.

When Islanding detection is required for either intentional or unintentional islanding, the AMM algorithm assists with observability issues by placing the PMUs strategically to ensure full observability of the network, even in the event of islanding behaviour.

For detection of unintentional islanding, it is recommended to use the standard AMM algorithm on the network that is addressed. When intentional Islanding is the focus of the network designer, three different ways in which AMM can help, are presented:

1. A standard AMM calculation for optimal PMU placements is followed by the Islanding planner using these PMU positions to complete the Islanding design. Where the islanding design refers to the deliberate segmentation of the network into islands for system restoration.
2. Again the standard AMM is applied to get optimal positions for the PMUs placements to render all the islands fully observable. Possibilities of smaller Islands will be analysed. If it is possible, then AMM allows the Islanding planner to have a say where some PMUs are necessary and the rest of the PMUs required for full observability will be determined by the calculation.
3. The situation where the planner is given complete flexibility (only likely when capital is not an issue because the amount and costs of the PMUs to be used is not the priority) to design the different islands of the power network without considering how the buses will be monitored. After the islands are apparent, the PMUs will have to be placed according to the topology of the islands using AMM. This needs to ensure that each island be fully observable on its own.

The student has also presented in the thesis an ILP-AMM hybrid PMU placement algorithm to show that combining these two techniques produce even better results than the standard AMM algorithm for large systems with zero-injection buses connected to other zero-injection buses. A table showing the results of ILP-AMM algorithm applied to the same systems as the standard AMM algorithm is shown in Table 2 below:

Table 2: ILP-AMM results

	IEEE 14 Bus System	IEEE 30 Bus System	IEEE 57 Bus System	1009 Bus SA Network
Algorithm	Minimum No. of PMUs	Minimum No. of PMUs	Minimum No. of PMUs	Minimum No. of PMUs
AMM	3	6	15	187
ILP	3	7	12	226
ILP-AMM	3	6	16	174

It is recommended that ILP-AMM algorithm be applied alongside the AMM algorithm for PMU placement for full observability.

It is therefore finally recommended that when PMUs are to be optimally placed for full observability of the power network, various algorithms might be used on the same network. The reasons are that different systems have different topologies and the different algorithms address certain topologies better than others. An example of this is the AMM and ILP-AMM Hybrid algorithms producing much better results than a normal ILP algorithm for a large system with zero-injection buses connected to other zero-injection buses. If the client is not sure as to which specific algorithm is best for the network at hand, then the safest route is to apply as many algorithms as possible.

Table of Contents

List of Figures

List of Tables

Glossary

PMU	Phasor Measurement Unit
TVE	Total Vector Error
N_{PMU}	Number of PMUs
N_{sol}	Number of possible solutions (Binary search)
BPSO	Binary Particle Swarm Optimisation
ILP	Integer Linear Programming
IGA	Immunity Genetic Algorithm
BOI	Bus observability Index
SORI	System Observability Redundancy Index
SCADA	Supervisory Control and Data Acquisition
AMM	Advanced Matrix manipulation algorithm
Island	Segment of a power network
IED	Intelligent Electronic Device
DG	Distributed Generation
EMS	Energy Management system
WLS	Weighted Least Squares
WAMS	Wide Area Monitoring System
GPS	Global Positioning System
PDC	Phasor data processor
CT	Current Transformer

VT	Voltage Transformer
PT	Potential Transformer
IEEE	Institute of Electrical and Electronic Engineers
KCL	Kirchhoff's Current Law
MTU	Main Terminal Unit
RTU	Remote Terminal Unit
SE	State Estimation

1. Introduction

1.1 Background to the study

In Power systems, it is generally understood that there is no effective mass energy storage system for electrical energy generated at power stations. Therefore power is usually generated based on the load requirements at the specific times. This suggests that it is imperative to have access to loading information. The Wide Area Monitoring System's (WAMS) purpose is to attain this information and make it available to wherever it is required for normal operation of the power system. The WAMS oversees the entire grid network where its operations are often modulated to monitor various segments of the network separately. SCADA systems currently govern WAMS operations.

The product of WAMS is the State Vector. The State Vector is achieved through a process called State Estimation which makes use of voltage and current data of the various system buses. The measurements of each bus of the system are represented as separate vectors and are processed by the main SCADA computer to form the state vector. These measurements are conventionally made by SCADA data measurement systems. SCADA makes use of RTUs for its measurement applications. The problem is that SCADA RTU measurements are not synchronised with respect to each other, nor are they available in real-time. When a fault occurs within the system, the measurements will not reflect whether the fault had actually occurred until moments later. This inhibits the system response mechanisms' efficiency due to slow availability of information [**1**]. Slow reactions to fault conditions cause damage to electrical infrastructure. Huge Blackouts may also occur due to inability to contain the fault in time.

Lots of attention was then given to WAMS operations, in order to improve the reliability of the data that is used for State Estimation. It was necessary to develop measurement systems that could provide data at a much faster rate and less prone to measurement errors, compared to that provided by the conventional measurement devices. The product of this research was the development of the Phasor Measurement Unit. Phasor Measurement Units are able to measure the voltages and currents of the buses in the branch lines. They provide real-time, synchronised measurements of the buses of a network due to a high sample refresh rate and GPS clock system. Their measurements are in phasor form and are easily processed. When PMUs are included in the power system, they make the State Estimation model more reliable, since the measurements are now synchronised with respect to one another and are in real-time. So instead of the conventional Static State Estimation, which uses sampled SCADA data allowing the State vector to be updated every few minutes, Dynamic State Estimation can be applied, which uses PMU real-time data to provide a real-time State Vector. The efficiency of the power system operation is improved because it can recognise fault conditions immediately as they occur and thus respond.

The replacement of RTUs with PMUs in power networks will not happen very quickly. This is mainly due to monetary issues. However it is clear that PMUs need to be incorporated into power networks as soon as possible, because of the critical advantages they present. For this specific reason, the study of optimal placement of PMUs was introduced. The amount of PMUs used in the network needs to be minimised to a level where they can monitor the system state conditions of the entire network, while using as few of the devices as possible. Various researchers have applied different optimization algorithms to determine the optimal locations for the PMUs.The toplogy of the network significantly influences the algorithm solution due to consideration of which buses are connected to each other and how many buses there actually are. The reason being that PMUs can measure the voltages of the buses they are placed at as well as the currents flowing in the lines connected to the placement bus. If the line impedances are known, the voltage data of the connected buses can be determined by Kirchhoff's Current law (KCL). The actual optimization logic is unique to each method. Several Algorithms are referred to in this thesis and it is shown that due to their differences, they present different advantages for different situations.

1.2 Objectives of this study

PMU placement for State Estimation is the main focus of this thesis. The main objective is to assess various placement algorithms used for Optimal PMU placement in terms of which algorithms provide what solutions for different network topologies. Once this assessment is complete, the student is to present an algorithm that can generate PMU placement schedules for any given power system. The algorithm must produce placement solutions that are cost effective and either similar to or better than placements of other algorithms. This is developed as *The Advanced Matrix Manipulation Algorithm*.

Due to monetary issues not all the PMUs will be installed at once. Therefore it is a requirement of the presented algorithm to schedule PMU placements for various stages of installation. In each of these stages, a placement schedule must be produced while taking into consideration how much funding is available for PMUs in each stage. A total number of stages must also be predefined by the algorithm.
System engineers often require certain buses to be monitored with greater importance compared to others. It is therefore an additional requirement that the algorithm will provide schedules to make such buses observable before others. Due to the critical buses possibly not being on the list of optimal bus locations for PMUs, it is likely that the minimum amount of PMUs required will increase. It is therefore a requirement that the algorithm must reassess the network after every bus is added that is not on the optimal locations list, in order to determine the optimal bus locations for the network to make the rest of the buses observable. This will prevent the case where PMUs are added to the optimal list to specifically make the critical buses observable, instead of optimising the list by taking into consideration the PMUs installed to monitor the critical buses, which is desired.

Another topic the new algorithm will address is aiding Bad Data detection processes. Bad Data is identified by State Estimation methods. It is a requirement of this process to be provided with redundant measurement data. The redundant measurements can be provided by PMUs if they are located at certain bus locations. These locations depend on the topology of the network and will be different for every system. The presented algorithm will be adapted to incorporate such a function. This aims at providing PMU placements to meet a desired level of measurement redundancy.

Islanding of the network sometimes occurs unintentionally due to factors such as distributed generation. This causes lines to be livened, sometimes unexpectedly. It is a requirement of the algorithm to locate PMUs to be able to detect the formations of these livened islands within the network.

Islanding also occurs when the system was designed to operate in that way, where distributed generation is specifically localised to certain areas of the network to intentionally form islands of livened buses once blackouts occur. In such a situation, the system data must still be accurately monitored for State Estimation. The PMUs have to be located so that each island is observed independantly of all others. The algorithm must cater for this function.

1.3 Assumptions and Limitations

The following assumptions were made during the process of this thesis:

1. All PMUs cost the sameThis refers to PMUs that connect to different amounts of lines, for e.g., a 4 line PMU and an 8 line PMU. It is probable that these PMUs would be priced differently. If the price of PMUs was incorporated into the optimization algorithm, the algorithm would try to use the cheaper PMUs more often than the expensive PMUs. The main goal would be to attain a placement that would cost the least. To do this effectively prices of PMUs from all manufacturers would be necssary. This however was not part of the scope for this thesis project. It can be ammended on to the algorithm at a later stage for further work.
2. There are sufficient telecommunication facilities for PMU operations. The telecommunication infrastructure allows the PMU to send and receive data. If data cannot be sent or received via this facility, the PMU would be rendered useless. This thesis only covers the power infrastructure used in monitoring applications. It is therefore assumed that the telecommunication area is taken care of.

The programs ran on a PC which has an Intel Pentium ® 4 3.00 GHz and 960mB RAM. If a faster processor had been used or more RAM available, a quicker running time may have been achieved.

1.4 Plan of development

1.4.1 Literature review

This section is a review of articles, papers, reports and books about topics relating to the thesis. SCADA systems are discussed; what they comprise of and how they function. Then Phasor Measurement Units are addressed in a similar manner. The strengths and weaknesses of the two technologies will be compared. Then the process of incorporating PMUs into SCADA systems is dealt with. Once these issues are explained, the PMU placement process can be explained. Here basics of bus observability issues are explained. Five placement algorithms are mentioned discussed afterwards. They are Binary Search algorithm, Binary Particle Swarm algorithm, Integer Linear Programming, PMU placement for Complete and Incomplete Observability and lastly the Immunity Genetic Algorithm. The logic behind the different optimization methods will be explained and compared to each other. The results that they produce are tabulated and discussed.

1.4.2 Methodology

In this chapter the stages of the development of the Advanced Matrix Manipulation Algorithm are explained in detail. The logic behind each section of the algorithm is also explained. It is shown that each section of the algorithm is specifically designed to achieve the various objectives of the thesis. The algorithm is set to be tested on the IEEE 14, 30 and 57 bus systems, as well as a large 1009 bus system received from a utility case-file. An ILP counter algorithm will be tested on the same systems to compare results with that of the Advanced Matrix Manipulation.

The PMU phasing placements and Bad Data Detection Aid algorithms will also be tested on the above mentioned bus systems.

The Islanding detection solver is not a separately coded add-on algorithm by itself (such as the standard optimal placement solver, the phasing PMU installation solver and Bad Data detection aid solver). However it is accomplished through special use of the phasing algorithm. This will be discussed in detail. A single example for each method of Islanding detection which is presented will be done using the IEEE 30 Bus test system.

1.4.3 Results

All the results of the tests described in the methodology section will be displayed in this chapter. The general format of a result will comprise of the bus locations for PMU placement and sometimes the calculation time is included. The results are summarised in tabular form. The tables will include information of which bus system was simulated on and which buses are to receive a PMU placement. The results of the ILP counter calculation is included in the tables for quick comparison between the the algorithms. Should it be desired to see the full outcome of each test done in Matlab, then this can be found in the Results Appendix file on the CD (where the full solutions contain descriptions of the calculations at each stage. The solutions are in list format rather than tabulated).

1.4.4 Discussion of results

Here the results will be analysed and compared to each other. Observation reports will be made on the results for each algorithm tested for the different bus systems they were applied to. Most of the results follow the same general trend, which will be discussed. The results which deviate are specifically mentioned and an explanation is provided for these occurrences. It must therefore be noted that not all of the results will specifically be mentioned as there are far too many (compared to the results section where every single result is tabulated), which will become tedious to read.

1.4.5 Conclusion

This chapter summarises the objectives of the thesis and compares it to what has actually been achieved. Conclusions will be drawn on the results obtained as to whether they are acceptable or not, depending ofcourse on whether they meet the objectives described. This will determine whether the Advanced Matrix manipulation algorithm is acceptable to be used in practice, or if it requires further work to be done.

1.4.6 Recommendations

Here recommendations will be made regarding algorithm choice for various applications in PMU placement. These recommendations are subject to the research work completed in this thesis. It is up to power system engineers and planners what choices they will make when planning PMU schedules for their networks.

2. Literature Review

2.1 Introduction

The main focus of this thesis is the placement scheduling of Phasor Measurement Units (PMUs) in a power network. Before that is dealt with, an introduction and description is presented of the systems that are associated with PMU placement. To understand where PMUs fit in, one has to be made accustomed with the Energy Management System (EMS).

The EMS is the general term given to the monitoring, control and optimization of energy conversion at a power station level, transmission and further distribution of this energy to where it is required in the connected grid system. The purpose of the EMS is to govern the flow of energy across the grid and maintain stable operation of the grid at all times. Currently EMS systems are governed by conventional Supervisory Control and Data Acquisition (SCADA) systems. The review will start with the broader picture of SCADA systems and narrow down to PMU placement topic which is the main focus of this thesis.

2.2 SCADA Systems

2.2.1 Basic Architecture

Supervisory Control and Data Acquisition (SCADA) systems are used in power systems as the governing software interface for the Energy Management System, for monitoring and control of system variables in order to maintain suitable operating conditions. SCADA systems comprise a control centre (where all the data processing takes place), Remote terminal units (RTUs) and communication networks that transfer the data [**2**]. The control centre collects the data, stores it and sends control instructions for system operations. The control centre further comprises a control server, Main Terminal Unit, Communication Routers, Human-Machine Interface (HMI) and a data storage, or data historian system which are all connected in a LAN [**3**]. The HMI is a user interface that allows the operator to manage the system. The data historian stores the data received from the RTUs. The Main Terminal Unit acts as the SCADA server. Communication routers enable data transfer to the Main Terminal Unit from the RTUs. The Communication mediums used are generally either telephone or power cabling methods, radio-wave methods or satellite communication methods. A summary of the SCADA system's architecture is shown in Figure 1 below.

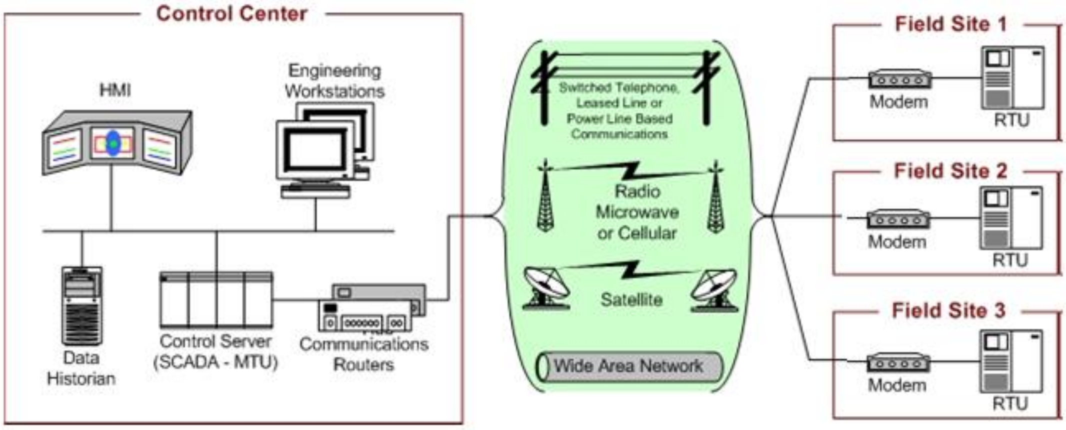

Figure 1 SCADA system architecture [3]

2.2.2 SCADA Functions

- Monitors and controls the status of circuit breakers, connectors and relays
- Monitors status of lines and buses
- Performs State Estimation of the power network
- Performs short-term load forecasting for the following few hours according to which day it is, what the weather conditions are etc.
- Assesses network faults conditions
- Performs network analysis of current and forecasted conditions for stability optimization
- Performs power generation control scheduling
- Supervises power flow throughout the network

2.2.3 Remote Terminal Units

RTUs are used for both measurement and control purposes. RTUs consist of Digital input and output channels which connect to the substation. They are 10 to 30V DC powered and connect to a PC via RS232 or TCP/IP protocol. The RTU uses an RS485 network to communicate with the Master station. They connect to meters which are connected to transducers such as current transformers and similar devices.

The data gathered includes real and reactive power flows in transmission and distribution lines, bus voltage magnitudes and circuit breaker statuses. Measurements are taken in a time window of a few seconds [**4**]. Access units may be available at the remote sites where the RTUs are located for operators to use.

The communication between the control centre and remote sites commonly use Modbus® or DNP3 communication protocols. The data is transported via cables, telephone lines, fibre optics, satellite or radio frequency methods, as mentioned above. RTUs act as the interface between the utility grid or power network and the control centre. A RTU and its modules are shown in Figure 2 below.

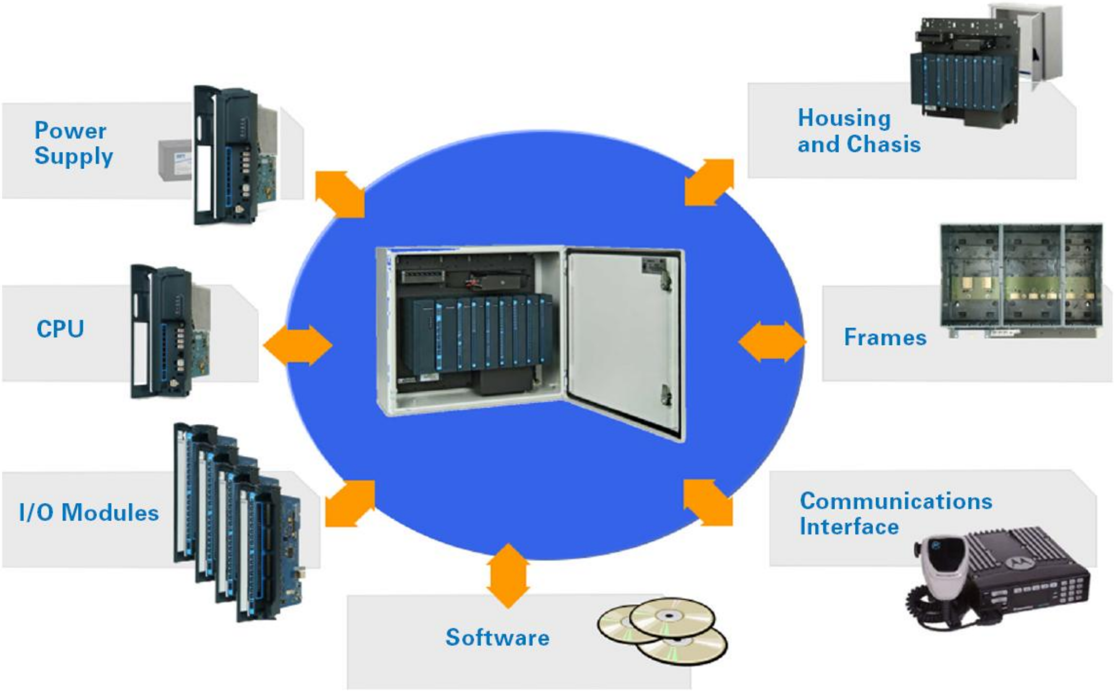

Figure 2 Basic RTU setup [5]

Advantages of RTUs

- RTUs have proven to be very reliable and robust in their operation
- They have the ability to process monitoring and control actions

Drawbacks of RTUs

- RTUs provide measurement data samples several seconds apart from each other
- The Main Terminal Unit is not able to time-align these measurement samples on a real-time basis
- RTU measurements cannot be compared to one another on a real-time basis
- The State Estimation using RTU measurements takes long due to complexity of the measurements

2.3 Phasor measurement units (PMUs): Architecture and Functionality

A PMU is a measurement processing device. Like the RTU, it is connected to buses at substations. The PMU makes use of measured currents and voltages from voltage and current transformers. This is received in an analogue form and is processed by an anti-aliasing filter which removes all components of the signal larger than half the Nyquist sampling rate [**6**]. The measurement signal is then put through an Analogue to Digital Converter (ADC). The digitised measurement is thereafter time-stamped by the GPS clock. The digital samples go through a microprocessor which converts the measurements into a phasor form of data. Now the measured sample is ready to be sent to the Phasor Data processor (PDC). The PDC can be compared to a mini-control centre, only there may be multiple PDCs in a system. In other words, singular centralised control is not necessary.

For systems of 50 to 60 Hz, 50 to 60 samples are provided per second respectively by the PMU. The data is in accordance with the IEEE C37.118 communication protocol. This new standard is used to enable communication of real time data. The data flow of the PMU is possible via Ethernet or serial standard form. Figure 3 shows a diagram of control blocks of a PMU.

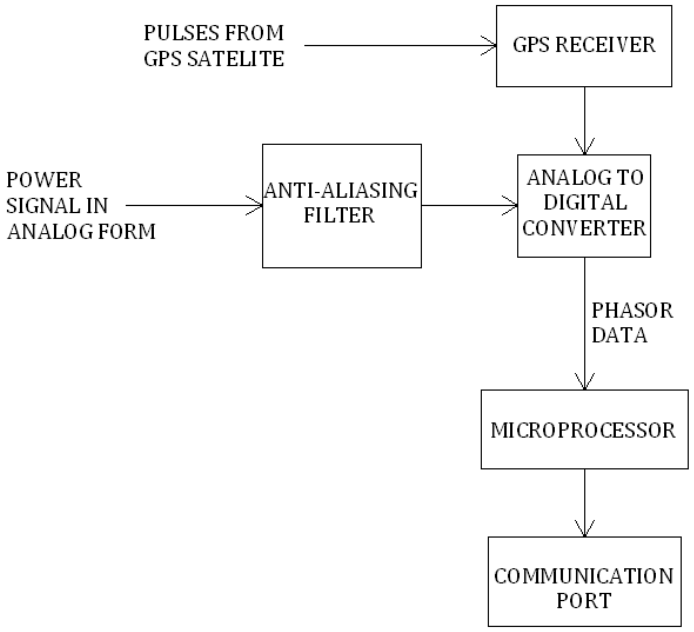

Figure 3 Basic PMU functionality [7]

The actual phasor is derived from the complex voltage and current sinusoidal waveforms of the measured buses. PMUs measure the voltage phasor of the resident bus and current phasors in the lines connecting the adjacent buses. The length of the phasor is taken from the maximum value of the sinusoidal graph and the angle, defined by IEEE C37.118 is zero degrees when the cosine maximum matches with a GPS pulse; and -90 degrees when the positive gradient zero of the graph matches with another GPS pulse. A reference bus is chosen within the network. All the other buses' readings are relative to the reference bus. As shown in Figure 4 below, the grey dotted sinusoidal is the reference signal, which allows the blue and red sinusoidals to be compared to each other.

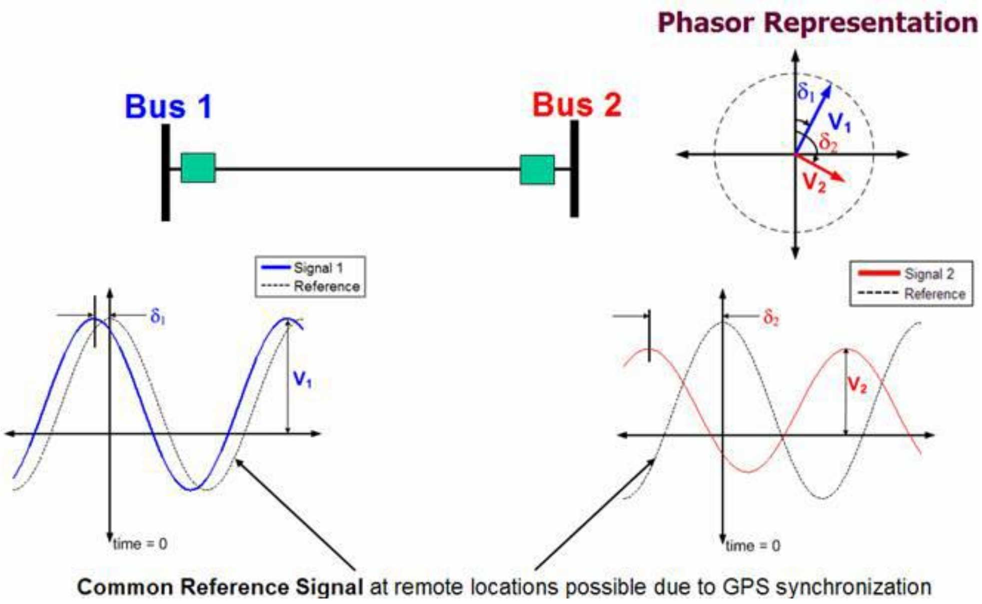

Figure 4: Phasor representation measured waveforms by a PMU [8]

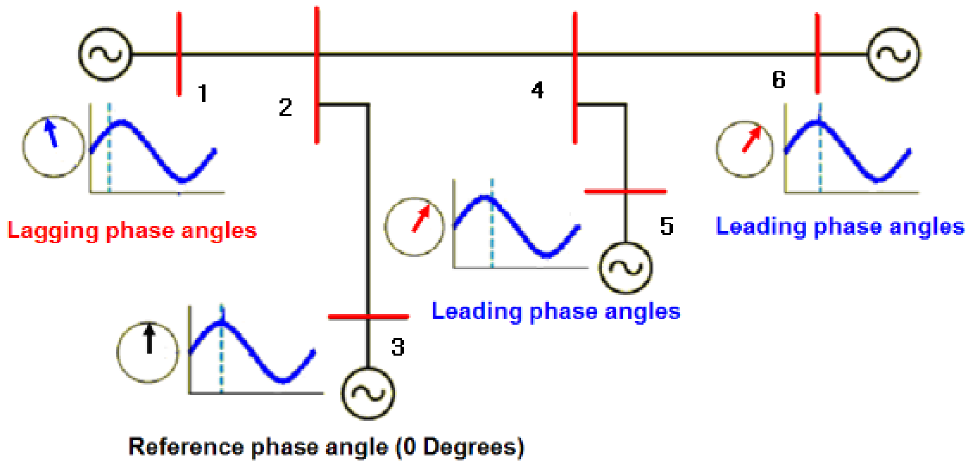

Figure 5 Bus network with phasor-sinusoidal relationship [8]

In Figure 5, measurements are made at buses 1, 3, 5 and 6 of the network shown. The measurement at bus 3 is chosen as the network reference bus. It is to be noted how the phase angles of the other measurements are either leading or lagging behind the reference bus measurement.

2.3.1 PMU installation process

A Synchrophasor is a synchronised phasor measurement of either voltage or current. Therefore the PMU must be connected to a current transformer (C/T) as well as a potential transformer or P/T (also known as a voltage transformer, V/T). Due to three phase reticulation and the presence of two measurement devices, six connections are required. The C/T's and V/T's are also known as instrumentation transformers as they are necessary to get the currents and voltages down to a level that is safely usable by a low current instrument such as the PMU. The C/T and V/T's outputs are further attenuated before entering the PMU as shown in Figure 6 below. The PMU also needs an antennae connection for the GPS system it uses as well as a connection to the communication medium. The PMU requires a power source and grounding. A competent person will have to test whether the system is working once completing the installation design provided by the engineer. [**8**]

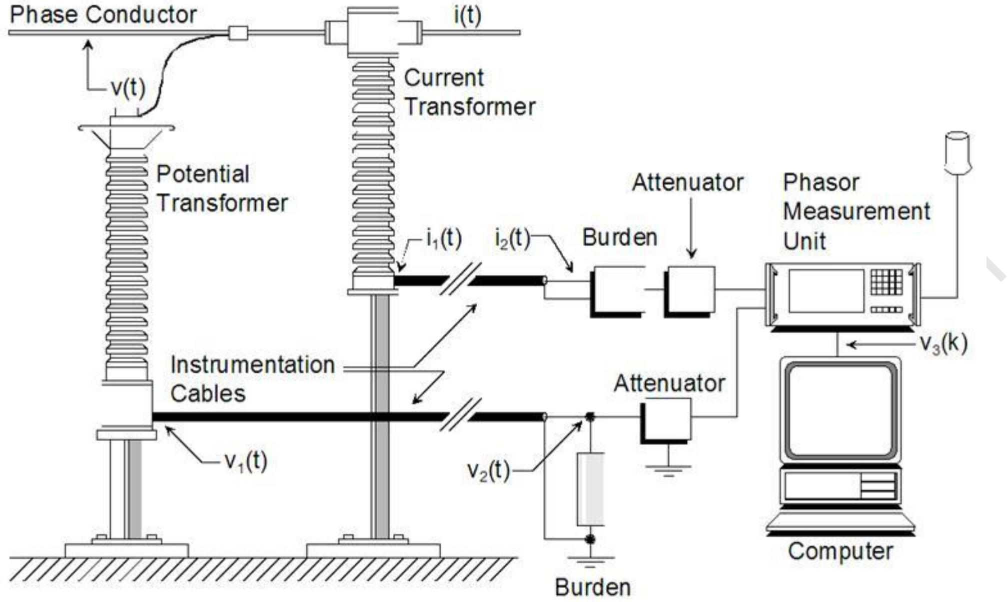

Figure 6 Connection from conductor to PMU [**8**]

2.3.2 Phasor Data processor

The PDC is a local PMU PC that gathers all the data from the PMUs linked to it. It time aligns the sampled measurements according to their time of origin. This process is critical since a misaligned phasor can cause a phase shift in its phasor representation [**9**]. The PDC stores this data as well as carry out a number of quality checks to determine the presence bad data. A PDC might either share its information with a super PDC (when only PMUs are used in the network), or to the existing main SCADA Control centre computer. A point noted is that SCADA systems mainly use Modbus or DNP3 protocols. However, PMU manufacturers have developed converters which transform IEEE C37.118 to Modbus® or DNP3 for use in the SCADA operations [**10**].

2.3.3 Synchrophasor communication systems

The communication medium for PMUs may be of the following methods:

- Power line communication
- Wireless networks
- Optical fibre networks
- Satellite Networks

These are the same as that used by RTUs. Therefore it is not necessary to install entirely new communication networks if segments of the existing network can be used. All of these methods have their respective drawbacks such as:

- Power lines have high bit error rates
- Wireless networks have limited coverage capacity and security issues
- Optical fibre installations can be costly
- Satellite communication has high usage costs and is hampered by the poor weather

The recommended option is optical fibre networks; due to its high bandwidth capacity and the speeds at which data can be transported at. The high cost is due to it being a wired network, so installation will not come cheap. The cost pros and cons are debatable, since once it is implemented, the network can be used by other entities as well [**11**]. It is therefore up to utility managers to decide whether they will allow such usage of their fibre optics network.

2.4 Applications of Phasor Measurement Units

PMUs facilitate various applications due to their time-synchronism and real-time characteristics. A few of these applications will be addressed in this section, namely: State Estimation, Bad Data Detection, Instability Prediction and Islanding Detection.

2.4.1 State Estimation

State Estimation via PMUs is described in this section. To understand how the State Estimation models are improved, State Estimation via RTU measurements will firstly be discussed. The Weighted Least Squares (WLS) method of State Estimation is discussed in this section.

2.4.1.1 State Estimation using SCADA measurements –Static State Estimation

State Estimation is the process of obtaining voltage profiles of all the buses in a power system and ultimately making up a state voltage vector. This is carried out by means of current and voltage transformer devices that measure the power-flow in buses. The accuracy of the State Estimation process directly affects the reliability and effectiveness of analysis and control of the power system. At present, mostly SCADA systems' RTUs are used to measure and manage power system profiles [**12**]. SCADA systems provide measurements of voltages, currents, active and reactive power-flows and injections [**13**].

State Estimation here is carried out by the Weighted Least squares method and its formulation is given below [**14**] as equation (2-1):

$$\left[\boldsymbol{G}\left(\boldsymbol{x}^{\boldsymbol{k}}\right)\right]\Delta\boldsymbol{x}^{\boldsymbol{k+1}} = \boldsymbol{H}^{\boldsymbol{T}}\left(\boldsymbol{x}^{\boldsymbol{k}}\right)\boldsymbol{R}^{-\boldsymbol{1}}\left[\boldsymbol{z} - \left(\boldsymbol{x}^{\boldsymbol{k}}\right)\right] \qquad \textbf{2-1}$$

where:
$G(x^k)$ is called the Gain Matrix
x^k is the State Vector
H^T is the measurement Jacobian matrix
R is the measurement error covariance matrix
$h(x^k)$ is the measurement function
and $z = h(x^k)$ + e where e is the error vector for each measurement

The Gain Matrix is made up using matrix H and R and is given by Equation (2-2):

$$\boldsymbol{G = H^T R^{-1} H} \qquad \textbf{2-2}$$

G is therefore a symmetric matrix and is formed by iterating through a single measurement at a time. k is the measurement iteration being dealt with.

The State vector x^k = $[x_1, x_2, x_3, \ldots\ x_n]$ represents the bus vectors as a single vector. This is updated for every measurement iteration of k.

The measurement function $h(x^k)$ is a representation of the measurements recorded and is in terms of the state vector. It is either represented in polar or rectangular coordinates.

The measurement Jacobian H is made up from the measurement function values and measurement difference values evaluated from the derivatives of the measurements with respect to the State variables. The matrix is divided into real and reactive power injection vectors, real and reactive power flow vectors and voltage and current magnitude vectors.

The covariance of the vector of measurement errors gives the value of R. It must therefore be noted that this method intends to minimize both the errors between measurements as well as the errors between estimations of these measurements. More information is available in [**15**].

The process of WLS State Estimation is carried out by firstly setting k = 0 and initialize the state vector x^k for a flat start. For each of the iterations of k, x^k needs to be calculated so that H in turn can be determined. Once this is completed and R is in place, the right hand side of the WLS equation is complete. The Gain matrix must be calculated from H and R. The Gain matrix must now go through Cholesky decomposition in order to obtain the form in equation (2-3):

$$\boldsymbol{G = L.L^T} \qquad \textbf{2-3}$$

Now we can see that:

$$\boldsymbol{L.L^T \Delta x^k = H^T(x^k) R^{-1} [z - (x^k)]} \qquad \textbf{2-4}$$

So make a substitution of $L^T \Delta x^k = u$ and then equation (2-5) shows that:

$$\boldsymbol{L.u = H^T(x^k)R^{-1}[z - (x^k)].} \qquad \textbf{2-5}$$

Solve for u. Now Δx^k can be determined from $L^T \Delta x^k = u$.
Δx^k is the change in the State vector; what we are looking for.
Δx^{k+1} is expanded as $x^{k+1} - x^k$. The Gauss Newton method [**14**] makes this SE process iterative. The iterations come from RTUs providing measurements every 5-10 seconds. Therefore each sample is used as an iteration which allows the SE model to provide results several minutes apart from one another.

Weaknesses of the WLS Estimation method

The stability of the State Estimation is determined by the condition number of the formulation and is given as [**12**] equation (2-6):

$$\boldsymbol{Condition(G) = ||G||.||G^{-1}||} \qquad \textbf{2-6}$$

The condition number is 1 for identity matrices and approaches infinity for matrices that are more and more singular. A system is ill-conditioned when small errors in the equation inputs translate into huge errors in the solution. The following causes of ill-conditioning in WLS Estimation are known:

- Long and short lines connected to the same bus
- Lots of injection measurements to deal with

Weaknesses of Static State Estimation:

State Estimation traditionally uses the data provided by RTUs. Because of its provision of state vectors only every few minutes, no wide area real time control can be accomplished in the face of fast transients. Below are a few factors owing to this:

- Long execution time due to complexity of measurements
- Complexity of SCADA communication requirements (data must go to centralised computer to be processed)
- Bad data is complex to identify and separate from true data

2.4.1.2 State Estimation using PMU data –Dynamic State Estimation

Dynamic State Estimation implies that the State model is continuously providing an estimated State vector. This means that the Dynamic State Estimation model does not use iterations because it effectively uses real-time data provided by the PMUs. Dynamic SE uses instantaneous time-stamped data which is synchronised with other measurements.

The measurement vector z is given in equation (2-7) below [**16**]:

$$\boldsymbol{z = [H]x + e} \qquad \textbf{2-7}$$

Where H is the new Jacobian matrix of the measurements
e is a vector of the errors to be minimized

z can be expanded to look like equation (2-8):

$$\begin{bmatrix} V_1 \\ V_2 \\ . \\ V_b \\ I_1 \\ I_2 \\ . \\ I_l \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 & 0 & . & 0 \\ 0 & 1 & . & 0 \\ . & . & . & 0 \\ 0 & 0 & . & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 & . & 0 \\ 0 & 0 & . & 0 \\ . & . & . & 0 \\ 0 & . & . & 0 \end{bmatrix} \\ \begin{bmatrix} (k_1)_1 & (k_1)_2 & . & 0 \\ 0 & (k_1)_2 & . & (k_1)_l \\ . & . & . & . \\ (k_1)_1 & 0 & . & (k_1)_l \end{bmatrix} & \begin{bmatrix} (k_2)_1 & (k_2)_2 & . & 0 \\ 0 & (k_2)_2 & . & (k_2)_l \\ . & . & . & . \\ (k_2)_1 & 0 & . & (k_2)_l \end{bmatrix} \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \\ . \\ E_b \\ F_1 \\ F_2 \\ . \\ F_l \end{bmatrix} + e$$

2-8

V and I are the voltage and current phasors, respectively
l and b are the max number of lines and buses, respectively
k_1 and k_2 are obtained from the current equation (2-9) for current in a branch :

$$\boldsymbol{I_{ab} = k_1 V_a + k_2 V_b} \qquad \textbf{2-9}$$

where *a* and *b* are the start and end of the line.

By using the new linear form of *H*, Δx, the new State vector is related to the measurements by equation (2-10):

$$\boldsymbol{G\Delta x = H^T R^{-1} z} \qquad \textbf{2-10}$$

Where G is the gain matrix made up from *H* and *R* as in Static SE above.
Since the above formula shows an independence of iterative behaviour, this form of State Estimation is able to provide a dynamic State Vector, since it calculates the state vector in a single step. The Dynamic SE converges faster than the Static SE because:

- Measurements are in voltage and current phasor form, instead of having to use non-linear related power measurements which have to be be converted back to voltage and current form to be used in the state estimation calculation. This makes calculations easier and faster
- PMU data can be processed locally by PDCs

2.4.1.3 *Hybrid Static and Dynamic State Estimation*

PMU Systems are not able to completely replace SCADA's RTUs as yet, since Synchrophasor systems themselves are quite costly. Therefore it is proposed by [**12**], [**16**] and [**13**] that PMUs be introduced in power systems with existing SCADA systems. A hybrid State Estimation model was developed to use the results of classical Static SE with Dynamic PMU measurements.
The Hybrid technique used is a Two Pass procedure [**17**] where a Linear SE is carried out by making use of PMU measurement data and the current results of Static SE, to provide State Vector solutions. Since this will be making use of the linear Estimator, the same formulation is used as above in equation (2-11):

$$\boldsymbol{G\Delta x = H^T R^{-1} z} \qquad \textbf{2-11}$$

The measurement vector z = Hx + e can be expanded as follows [**12**] in equation (2-12):

$$\begin{bmatrix} \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{\theta} \end{bmatrix}_{\boldsymbol{PMU\_V}} \\ \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{\theta} \end{bmatrix}_{\boldsymbol{PMU\_I}} \\ \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{\theta} \end{bmatrix}_{\boldsymbol{SE}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H_{11}} & \boldsymbol{H_{12}} \\ \boldsymbol{H_{21}} & \boldsymbol{H_{22}} \\ \boldsymbol{H_{31}} & \boldsymbol{H_{32}} \\ \boldsymbol{H_{41}} & \boldsymbol{H_{42}} \\ \boldsymbol{H_{51}} & \boldsymbol{H_{52}} \\ \boldsymbol{H_{61}} & \boldsymbol{H_{62}} \end{bmatrix} \begin{bmatrix} \boldsymbol{V_R} \\ \boldsymbol{V_I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{e_1} \\ \boldsymbol{e_2} \\ \boldsymbol{e_3} \\ \boldsymbol{e_4} \\ \boldsymbol{e_5} \\ \boldsymbol{e_6} \end{bmatrix} \qquad \textbf{2-12}$$

where $\begin{bmatrix} U \\ \theta \end{bmatrix}_{SE}$ is the Static SE result in polar form
$\begin{bmatrix} U \\ \theta \end{bmatrix}_{PMU\_V}$ and $\begin{bmatrix} U \\ \theta \end{bmatrix}_{PMU\_I}$ are PMU voltage and current phasors respectively.
The elements of the H matrix are in turn matrices themselves which make up the entire Jacobian H matrix.

Hybrid SE therefore presents a conversion from static to dynamic SE by means of PMU data. If gradual phasing of PMUs continues until only PMUs are observing the entire power system, then pure Dynamic SE can be used.

The more PMUs are included in the system, the smaller the condition number becomes. This means that the SE becomes numerically more stable. This is shown in more detail in [**12**].

<u>Advantages of Hybrid SE</u>

- Improved State Estimation convergence time (from Static SE)
- Improved precision (from Static SE results)
- More Reliable estimation of the state vector since the inclusion of real-time data

2.4.2 Power System restoration

Power system restoration is the procedure of restoring all the functional components of the power system such as transmission lines, generators and loads after a partial or a total collapse of the power system grid. System restoration is aimed at achieving restoration in minimum time and with minimum damage to equipment. Reference [**18**] presents an algorithm where PMUs are used to assist with system restoration by restoring energy to sections of the power system (referred to as "islands") separately, in order to speed up the entire process of system restoration. This is achieved by making each island observable with PMUs and using the advantage that PMUs give synchronised, real-time measurements for fast restoration to take place.

2.4.3 Islanding phenomenon

Islanding refers to the situation when parts of a power network continue to be energised through distributed generators (that might or might belonging to the utility) despite the main utility supply being cut off. Islanding is therefore very useful for consumers who require an uninterrupted supply of energy, since the DGs compensate for the loss of utility supply. However, unintentional islanding is undesirable since it causes lines to be unexpectedly live when work has to be done on them. This is very dangerous to working personnel. Another case where islanding is addressed is in restoration of the network after a fault has occurred. According to [**19**], restoring grid conditions to each island instead of firstly restoring energy to the entire network proves for a faster system restoration duration. The more islands there are, the faster the restoration, because more attention can be given to the specific elements of that island. The voltage levels and frequencies of the buses in the island must be correct before they can reconnect to the Grid. This is a case of intentional islanding.

In both cases of intentional and unintentional islanding, the islanding phenomenon needs to be detected. PMUs are able to detect live buses and hence can be employed for islanding detection. The preference of PMUs over RTUs is due to the real-time measurements made by the PMUs. RTUs also have to send their data to the main SCADA PC for processing. If the central region where the Main PC is located is not powered due to the grid collapse, then the RTU measurement will not be processed by the main PC. PMU measurements can however be locally processed by PDCs. If a PDC is located within an island, then the PMU measurements can be processed and information on which buses are livened will be available.

2.4.4 Bad data detection

Bad data detection is the process of detecting false measurements that are made and discarding these measurements in especially critical situations where the redundancy of the measurements is low. For the purpose of State Estimation, the data being used must be accurate for the State Estimator to provide a reliable, true State Vector. However, at times data that deviates far from the current mean is picked up. There are State Estimation methods which deal with detection and removal of bad data

[**13**], [**20**] but that is not dealt with in this thesis. For these State Estimation methods it is assumed that a certain level of measurement redundancy exists to aid the process. The level of measurement redundancy refers to how many times a single bus is observed. The more times a bus is observed, the higher the level of redundancy. This helps with bad data processing, because if one PMU measures a bus's data incorrectly another PMU observing the same bus might measure the bus's data correctly. The more PMUs observing a single bus, the better the chance of determining whether the data is corrupted or if it is valid.

2.4.5 Instability Prediction

Instability predictions are calculation-based methods to determine how the power system would behave in the near future. The conventional ways of accomplishing this is by the direct integration of the system dynamic equations. The calculations involved are very extensive and had to be restricted to offline stability studies [**21**]. When PMUs were introduced, the real-time analysis advantage allowed instability prediction to go to a new level since the measurements made are real-time measurements which allowed prediction models to be easily adapted for prediction of the system for a few seconds ahead with a level of confidence [**21**]. Rephrased, the Instability Prediction is made more reliable than before due to the condition number of the power system with PMUs is better than for the system without PMUs.

2.5 Problems with PMUs

Price

The main motivations behind the research of optimal placement of PMUs are limited communication lines and more importantly the cost of PMUs. *SEL Smart Grid solutions* provides a total cost for a PMU system per substation to be $51000 U.S. dollars [**22**]. This is very expensive when it is considered that PMUs need to be installed over an entire power network. Therefore the fewer PMUs are required, the better. Due to monetary issues, the installation of PMUs is likely to proceed in different phases of installation, where after the final phase the power network is completely observable by PMUs. These phases would vary in time depending on the size of the network and how much capital is made available.

Uncertainties

PMU measurements are not free from errors. The measurement errors occur due to Analogue to Digital Converters, transformers and the connecting cables [**23**]. For this we consider the single PMU shown in Figure 7:

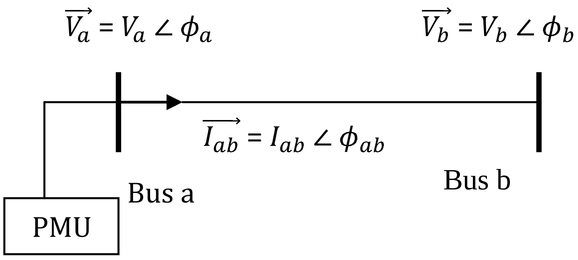

Figure 7: Basic PMU setup measuring two buses from [24]

The uncertainties in the voltage and the angle respectively [**23**] given as equation (2-13) and (2-14):

$$\boldsymbol{u(V_b)} = \sqrt{\sum_{k=1}^{4}[\partial V_b/\partial p(k)]^2\,[u(p(k))]^2} \qquad \textbf{2-13}$$

$$\boldsymbol{u(\phi_b)} = \sqrt{\sum_{k=1}^{4}[\partial \phi_b/\partial p(k)]^2\,[u(p(k))]^2} \qquad \textbf{2-14}$$

where *p(k)* is a function comprising $[V_a, \phi_a, I_{ab}, \phi_{ab}]$ and u(p) is defined as the standard uncertainty. The manufacturer-specified uncertainty is Δp and is given in equation (2-15)

$$\Delta p = \sqrt{3} * u(p) \qquad \textbf{2-15}$$

In the state estimation section above, the term e refers to these uncertainties.
The total vector error is an error factor of the PMU measurements compared to the expected values of the phasors and is shown below in equation (2-16):

$$TVE = \frac{|X_{measured} - X_{expected}|}{|X_{expected}|} \qquad \textbf{2-16}$$

2.6 Summarised Advantages and Disadvantages of PMUs

Advantagesof PMUs

- Measurements are effectively real-time measurements due to their high sampling rate
- Measurements are synchronised with respect to each other
- Measurements within the same system can be compared to one another on a real-time basis
- Relays with PMU functionality have been produced and therefore serve as monitoring and control devices

Disadvantages of PMUs

- PMUs are very expensive to implement on a large scale
- PMUs are still subject to component measurement errors

2.7 Phasor Measurement unit placement procedures

As mentioned before, the number of PMUs will have to be minimised due to cost and communication line availability. Many algorithms have been developed to tackle this problem and some of them will be discussed in due course. A good PMU placement algorithm is defined in [**24**] to compensate for the following issues:

- Outage of a PMU
- Acknowledging zero-injection buses
- Phasing the placements of PMUs

For this we consider Figure 8 showing the diagram of a simple 6-bus system:

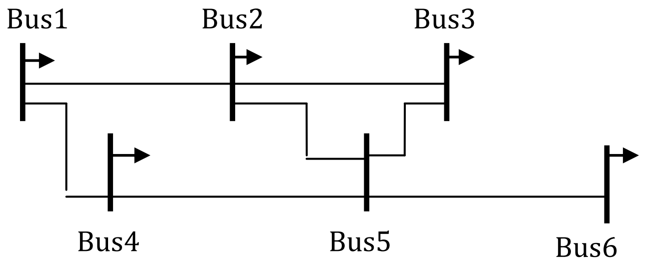

Figure 8: A simple 6-bus system

where Bus 5 is a Zero-injection bus and Bus 6 is Radial bus.

2.7.1 Outage of a PMU

Like all equipment, PMUs are prone to malfunction due to some cause or another. The same can be said for communication lines [**25**]. A good PMU placement algorithm should have methods of placing the PMUs such that when PMU or Communication outages occur, the level of observability of the buses measured by that PMU are kept constant by means of redundant measurements of surrounding PMUs. An example to explain this concept is taken from the Figure 8 above. If a PMU was located at Bus 4 and another at Bus 2, then Bus 1 would be rendered observable by the PMUs at bus 4 and bus 2. If the PMU at Bus 4 were to fail, then Bus 1 would remain observable due to the PMU at Bus 2.

2.7.2 Acknowledgement of Zero-injection buses

A zero-injection bus is a bus that has no load or generation sources attached to it. Therefore it can be modelled as a line with its own parameters. A PMU is able to observe the voltage phasors of the bus it is located at, as well as that of the buses adjacently linked to that bus. This is due to the ability of current measurement in the lines adjacent to the PMU bus. Since the line parameters are known, the voltage of the buses at the end of the lines can be calculated. In Figure 8, above, if a PMU was placed at Bus1, then Bus1, Bus2 and Bus 4 would be rendered observable by that PMU.

When a zero-injection bus is considered, no load is connected to this bus and so the sum of the currents entering the bus is the same as that leaving the bus. When a PMU is located at a bus directly linked to a zero-injection bus (and since it is able to observe the current through the lines), all the unobserved buses directly linked to the zero injection buses become observable [**24**]. In Figure 8 if a PMU is placed at Bu1 and Bus 2, then all the buses become observable. Thus the zero-injection bus presents a fundamental advantage in allowing for the requirement of less PMUs to make a system observable (if used correctly), than for a system without any zero-injection buses. If a PMU is placed at the zero-injection bus, then its advantage would be nullified. This is due to the PMU now treating the zero-injection bus like an ordinary bus, since it will only be able to measure the voltage of the housing bus and the current through the adjacent lines. In Figure 8, if a PMU was placed at bus 5, then buses 2, 3, 4, 5 and 6 would be observed but not bus 1.

2.7.3 Phasing the placements of PMUs

Phasing of PMUs refers to installing PMUs in stages and not all at once. In [**26**] it is generalized that an average of 20-30% of the buses in a system would require a PMU for full observability of the network. Phasing is due to the high cost of installation. When phasing takes place, it may occur that more PMUs will be installed than the optimal lowest amount that is needed. This is due to the desired amount of PMUs that are installed in that phase, are placed at locations with the highest level of observability. This logic is repeated for the next phase where the PMUs will be installed at the next highest observability locations. This continues until the entire system is observable. In [**24**] a method is introduced for phasing of PMUs such that the total number of PMUs at the end of phasing is not more than the optimal amount of PMUs required for full observability. This is achieved by installing the PMUs with the highest level of observability in the initial stages chosen from the locations of the optimal solution and phased by installing the next set of PMUs at the locations with the following highest observability level and so on. This method sounds more pleasing than the first method, but one must realise that it may be a requirement from the utilities that certain buses receive measurement preference over others. These preferred buses do not necessarily fall within the calculated list of optimal locations and therefore might cause the amount of PMUs to be more than initially calculated. However, the choice is up to the utility which methods of installation they prefer.

2.7.4 Radial buses

A Radial bus is a bus at the end of a branch of buses. It is linked to only one bus. In Figure 8 Bus 6 is a Radial bus. For the case of redundancy of observability, Radial buses can have a maximum of two PMUs observing them. In Figure 8 if bus 5 was not a zero-injection bus, then the most bus 6 could be observed would be twice, by a PMU at bus 6 and a PMU at bus 5. However in our example, when considering the presence of the zero-injection bus 5, the maximum times bus six can be observed is five times; by a PMU at bus 6, 5, 2, 3 and 4. PMUs at buses 2, 3 and 4 observe bus 6 through bus 5.

2.7.5 Connectivity Matrix

The connectivity matrix is a binary square matrix representing the connections of all the buses to one another. The number of rows and columns is determined by the number of buses in the power system. Each row represents the connections of each bus and each column has a binary 1 as an indication of the bus being directly linked to the incident bus. The connectivity matrix is spawn from the connectivity equations. Below is the list of connections of each bus for the example of the 6 bus system in Figure 8.

Bus1: $x_1 + x_2 + x_4$
Bus2: $x_1 + x_2 + x_3 + x_5$
Bus3: $x_2 + x_3 + x_5$
Bus4: $x_1 + x_4 + x_5$
Bu5: $x_2 + x_3 + x_4 + x_5 + x_6$
Bus6: $x_5 + x_6$

The connectivity matrix of the above system is given as

$$A = \begin{matrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{matrix}$$

2.7.6 Remarks

The items highlighted in this section are the main issues which the placement algorithms use to optimise the locations of the final PMU placements. Different algorithms will focus more on particular issues than others and therefore their results will differ. The reasons may be that the algorithm style is more suited to naturally deal with one aspect than others.

2.8 Placement Algorithms

This section contains a presentation of various PMU placement algorithms. All of these algorithms aim to minimise the amount of PMUs required to provide full observability of the network. These algorithms are mathematical optimisation methods that are also used for other applications outside of power system analysis. These algorithms have therefore been tailored to deal with PMU placements. Due to the nature of optimisation algorithms, there will always be several constraints present amongst which the algorithm must optimise to reach the best solution for optimal locations for the PMUs. The constraints are phrased differently for each algorithm, although they all adhere to the general constraint of minimising PMUs in whichever way that is allowed. The algorithms will take into consideration the issues highlighted in section 9.5 for placement procedures.

2.8.1 Binary search PMU placement algorithm

This algorithm considers all the possible combinations of location solutions of PMUs and narrows down to the optimal solution by undertaking a binary search formulation. The first step is to generate all the possible combinations of solutions. From [25] the total number of candidate buses for a PMU placement is given as P; and the (initial) number of PMUs N_{PMU} is given in equation (2-17)

$$\boldsymbol{N_{PMU}} = [(N + s/2)/3] \qquad \textbf{2-17}$$

Where N is the total number of buses in the system and s is the number of unknown power injections. Then the number of combinations N_{sol} is given in equation (2-18):

$$N_{sol} = \binom{P}{N_{PMU}} = \frac{P!}{N_{PMU}!(P-N_{PMU})!} \qquad \textbf{2-18}$$

A flow chart of the search method is given in Figure 9 below:

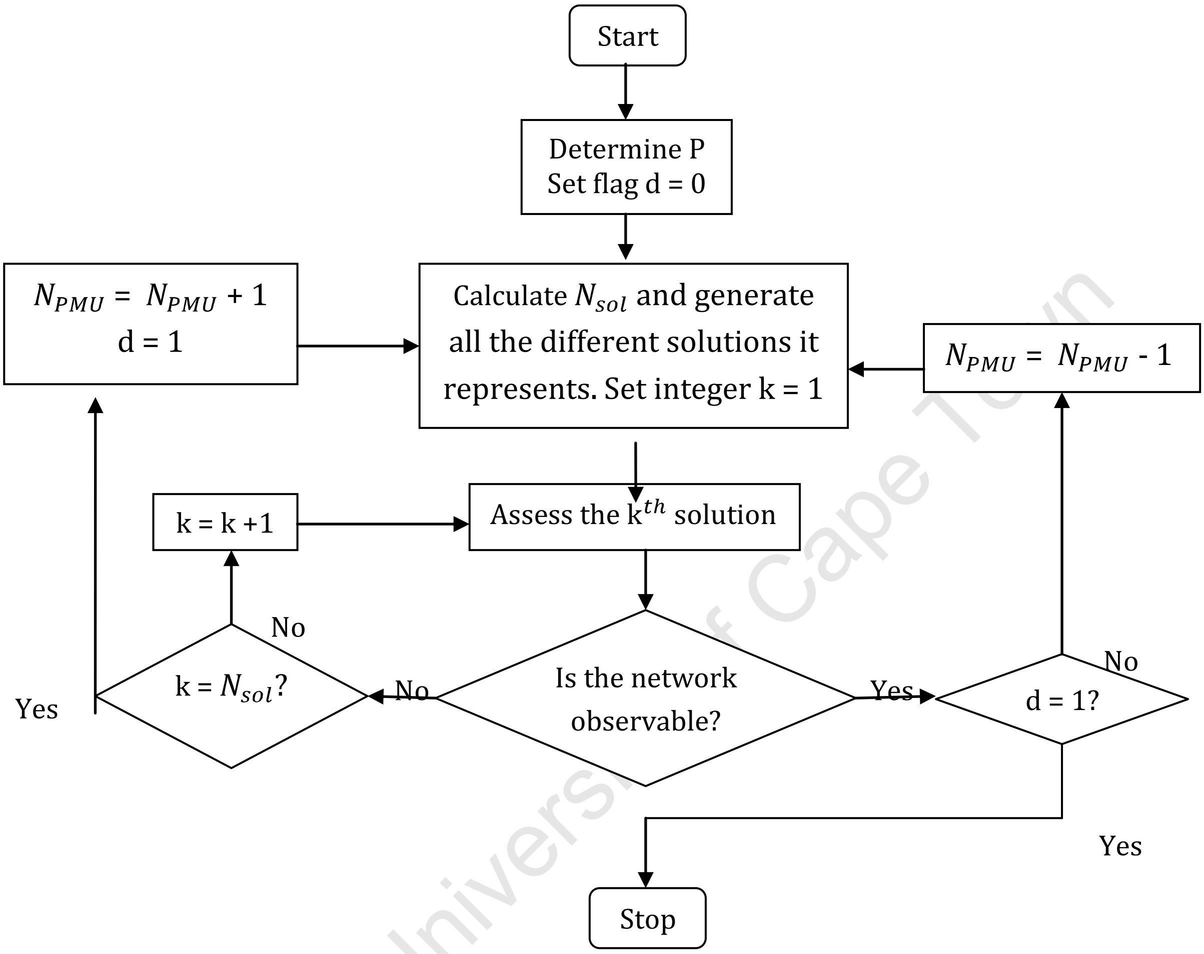

Figure 9: Flow diagram of the Binary search algorithm [28]

The method begins with determining the amount of candidate PMU locations P and initializing a flag d=0, which will be used later. Next the number of combinations can be computed and all the combinations of solutions can be generated. Set an integer k = 1; this represents the iterations of the formulation. Examine the kth solution and check whether it makes the system observable. If not, then increment k if it is not the last iteration and check the next solution. If k represents the last solution N_{sol}, then increment N_{PMU} and do a new calculation for total number of combinations and carry out the formulation again; also set flag d = 1. On the other hand, if the solution checked did make the system observable, then there is a check to see whether the flag is set or not, before that specific solution can be declared the optimal solution. If d is not equal to 1 then decrement N_{PMU} and calculate a new N_{sol} and do the formulation again. The purpose of checking whether the flag is set, (and decrementing N_{PMU} if it is not), is to check whether N_{PMU} is truly the least amount of PMUs required for full system observability. If the flag is not set, the method will check to see if a solution can be found with one PMU less and will repeat this loop until the solution renders the system unobservable and N_{PMU} has to be incremented. Then the previous result would be confirmed as the solution that uses the least amount of PMUs since the flag would be set.

Compensation for single line outages

Bus line outages are prone to occur and the PMU placement problem considers this event. The way Binary search algorithm achieves this is by following the same method as above, but after a solution is deemed observable, there is another check to see whether the system is observable when a line outage occurs. It checks for each line outage separately and if the solution makes the system fully observable for all of the line outage instances, then it is considered a possible solution. The method will then carry on as above to decrement N_{PMU} to pursue the least possible amount of PMUs for full observability.

Solving redundancy issues

When there happens to be more than one solution with all of these having the same least number of PMUs, the one with the highest redundancy level must be chosen as the optimal solution. To do this, define an integer variable r_i = [r_1 r_2 . . . r_q] where q is the number of candidate solutions. Now do a check and sum the amount of times each bus of solution 1 to q is made observable by a PMU and add each redundancy to each r value respectively. The solution with the largest r value is the optimal solution.

If it is desired that there be a minimum observability redundancy level for each bus, then $r_{min}$is the minimum amount of PMUs observing each bus and now r_i must now be compared to r_{min} of the whole solution with all $r_i \geq r_{min}$.

2.8.2 Binary Particle Swarm optimisation (BPSO)

This method to solve the PMU placement problem is based on the concept of a swarm of particles changing their positions and directions with time in order to collectively narrow down on a set of candidate solutions. The optimal solution is reached by individual particle experience combined with the experience of each particle's neighbours. This means that each individual particle adjusts its position due to its own experience combined with the knowledge of past experience, as well as the experience of the neighbours of that particle. Together they converge to the location of the optimal solution. If a better position exists from a past event or from the knowledge of one of the particle's neighbours, then the particle changes its direction; if the particle's position is better than that of the swarm then this will become the newest preferred position for the entire swarm to use [**27**]. The solution is found when no better positions are found by the swarm.

The BPSO method regulates and updates the particle position and velocity according to equations (2-19) and (2-20):

$$\boldsymbol{v_i^{t+1}} = \boldsymbol{v_i^t} + \mathbf{c1} \oplus \boldsymbol{d_{1,i}^t} + \mathbf{c2} \oplus \boldsymbol{d_{2,i}^t} \qquad \textbf{2-19}$$

$$\boldsymbol{x_i^{t+1}} = \boldsymbol{x_i^t} + \boldsymbol{v_i^{t+1}} \qquad \textbf{2-20}$$

Where $d_{1,i}^t$ and $d_{2,i}^t$ are defined in equations (2-21) and (2-22) respectively:

$$\boldsymbol{d_{1,i}^t} = \boldsymbol{pbest_i} \oplus \boldsymbol{x_i^t} \qquad \textbf{2-21}$$

$$\boldsymbol{d_{2,i}^t} = \mathbf{gbest} \oplus \boldsymbol{x_i^t} \qquad \textbf{2-22}$$

v_{id}^{t+1} is the next velocity
x_i^{t+1} is the next position
c1 and c2 are two random binary vectors
pbest is the best position of the individual particle so far
gbest is the best position found by the swarm so far
$\oplus$ is the logical XOR function

v_i^t is limited to v_{max}, which sets the search domain for the particles. v_{max} must be chosen carefully since too big a v_{max} produces the chance that the optimal solution is never reached and if $v_{max}$is too small, it can cause that the best solution not be found due to it not being present in that domain. $v_{max}$is chosen as a fraction of the number of buses in the system. If v_i^t reaches v_{max}, then to get a new v_i^t, it is reduced to below $v_{max}$by randomly decreasing the number of ones.

When starting up the formulation, previous methods chose an initial starting point. This starting point would help to converge to the optimal solution quickly if chosen correctly. It was chosen that the first PMU is placed at the bus with most buses linked directly to it; then the next PMU selected as the next bus with the next most buses etc. But this method proved to increase the total amount of PMUs required since there remained unobservable buses on the outskirts of the system. This showed that an optimisation formulation was necessary to keep the PMUs required to a minimum despite the phasing stage.

When the search space is considered, certain buses have to be removed from the group of candidate PMU locations because of their nature. One such bus is a zero-injection bus; if a PMU is located there, the advantage presented by a zero-injection bus (that it allows the PMU to observe buses one bus beyond the zero-injection bus) would be lost. Another is if the PMU is placed at a Radial bus. It would be wasting a PMU since at that location, the PMU can only observe a maximum of 2 buses (if the second last bus in that branch is not a zero-injection bus); a better location must be found that will make the radial bus observable without a PMU at that bus.

The fitness function which determines the fitness of all the individual particles is defined below in equation (2-23):

$$\mathbf{f = c1^*.N_{PMU} + c2^*N_{unobs}} \qquad \mathbf{2\text{-}23}$$

Where N_{PMU} and N_{unobs} is the number of PMUs presently in the system and the number of unobserved buses, respectively.

Figure 10 shows a flow chart of the formulation steps that must be followed:

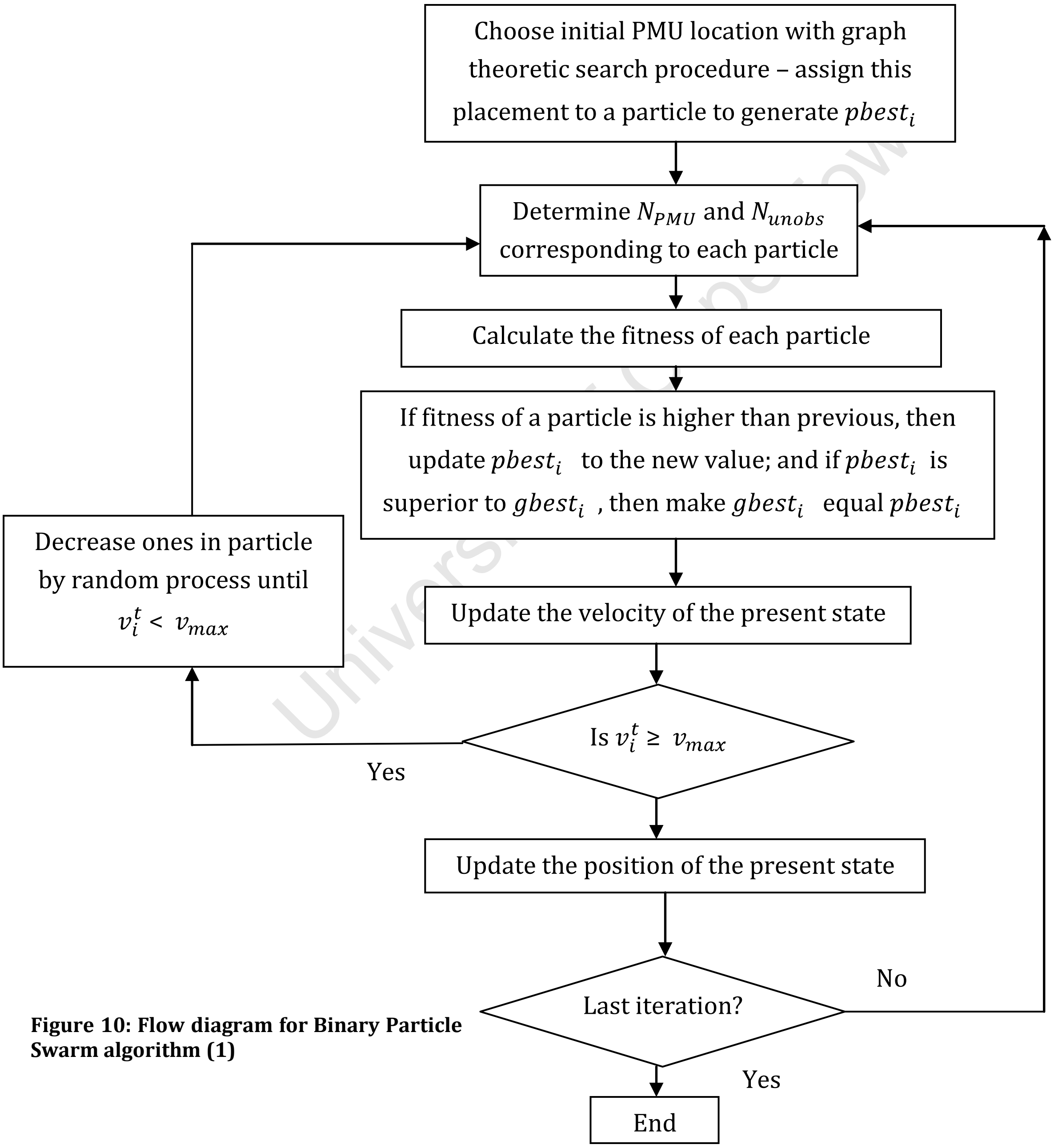

Figure 10: Flow diagram for Binary Particle Swarm algorithm (1)

2.8.3 Integer Linear programming (ILP)

The task of optimal PMU placement can be solved by using a linear programming approach. For this type of linear programming problem, an integer based formulation must be used to achieve the correct results. This is how linear programming has been specifically tailored for PMU placement. As in all linear programming problems, there is an objective function to be optimised which is subject to certain constraints; it is no different in the PMU optimisation case. The objective function is given as [**28**] equation (2-24):

Minimise: $\boldsymbol{x_1 + x_2 + x_3 + x_4 + \ldots + x_{N_{bus}}}$ **2-24**

Where N_{bus} is the total number of buses in the system.

Define a vector x_i = $[x_1\ x_2\ x_3 \ldots x_{N_{bus}}]^T$ which represents the vector of the buses. The constraints are thus given as equation (2-25):

$\mathbf{A}\boldsymbol{x} \geq \mathbf{e}$ **2-25**

Where e is a vector: $[1\ 1\ 1 \ldots 1]^T$ which generates the constraint that each bus must be observed by at least one PMU.

A Matlab optimisation toolbox such as Tomlab® can be used to solve this formulation by using the *cplex* function. The output is the amount and locations for optimal PMU placement.

The zero-injection bus ILP approach

Normally, to obtain a solution when including zero-injection buses, non-linear based methods are used, but [**24**] presents a linear approach to solving the problem. Their method works with identifying the zero-injection buses and modifying their constraint equations and that of the buses directly linked to them. If a system has zero-injection buses at buses 1, 2 and 4, the following process is carried out:

The method states that one of the buses in this inequality can remain unobservable. Now define a vector u: $[u_1\ u_2\ u_3 \ldots u_{N_{bus}}]^T$ where u depicts the confirmation of observability of the bus, by its binary nature of being equal to 1 when the bus is observable and equal to zero when it is not. Then the right hand side of the inequality above must change from 1 to u_1 and the same for bus 2 and 4, replacing the 1's with u_2 and u_4 respectively. The new inequality constraint is added and is given by equation (2-26):

$\boldsymbol{u_1 + u_2 + u_4 \geq 2}$ **2-26** where the number “2” represents the advantage of one of the three buses being unobservable. This means that only 2 of the three buses need to be observed, since the third bus is observed by a PMU which observes either of the other buses.

In reality, the zero-injection bus is allowing observability by the PMU one step beyond the zero-injection bus; this is how it is modelled for linearity. The new equality is then added to the list of constraints governing the objective function.

Redundancy criterion with ILP

When it occurs that more than one solution with the same minimum number of PMUs provides full observability, redundancy modelling must take place. This will allow for the true optimal solution to be chosen. In [**24**] a *Bus Observability Index* (BOI) is defined as the total amount of PMUs that can observe the bus in question. SORI or *System Observability Redundancy Index* is the sum of all the BOI’s of all the buses. The solution with the highest SORI is branded the optimal solution.

Sometimes it is desirable to have at least 2 PMUs observing a single bus or 2 PMUs for all the buses. This is dealt with in ILP by changing the constraint equalities; as can be seen with the example in Figure 8 bus 1:

Bus1: $\boldsymbol{x_1 + x_2 + x_4 \geq 2}$

This means that the optimal solution is that which allows at least two PMUs to observe the state of that bus. The purpose of this redundancy is in the case that one of the PMUs fail, then the bus will not be rendered unobservable due to another PMU monitoring the same bus.

2.8.4 PMU placement for complete and incomplete observability

This algorithm uses a spanning tree method to obtain a solution for full observability. The concept of depth of observability is introduced where the level of unobservability is explained as the number of directly linked unobservable buses connected to an observed bus which does not have a PMU situated at that bus.

The spanning tree method works by starting off at the root node (this can be chosen) and branches off to the terminal nodes. Terminal nodes are the nodes at the end of the branch. There are two categories: parent terminal node, which is at the end of the main branch; and the spanning tree terminal node, which is an intermediate branch from the path. After reaching the terminal branch, it will backtrack to the splitting point of the branch and explore the other directions in the same manner. Eventually it will backtrack to the root node after all terminal nodes have been reached. The first PMU is placed at the first bus after the root node for observability of the root node and of that bus. The next PMU is placed according to the following rule [**29**] in equation (2-27)

$$\mathbf{d_p = u + 3} \qquad \mathbf{2\text{-}27}$$

Where d_p is the number of buses away from the current PMU placement the next one will be; u is the desired level of unobservability.

When the spanning tree reaches the main terminal bus, it will not place the PMU at that bus even if d_p is less than u + 3. Instead it places the PMU one bus away, which still makes the terminal bus observable. This is to avoid the issue of placing a PMU at the radial bus/ terminal bus because of the issue mentioned. Then it backtracks to the previous split and searches this area until the entire network is exhausted and the root node is reached. Thereafter a new root node is selected and the process is repeated a few times with a different root node each time in order to ascertain the minimum number of PMUs for system observability.

The procedure is shown below as a flow diagram in Figure 11:

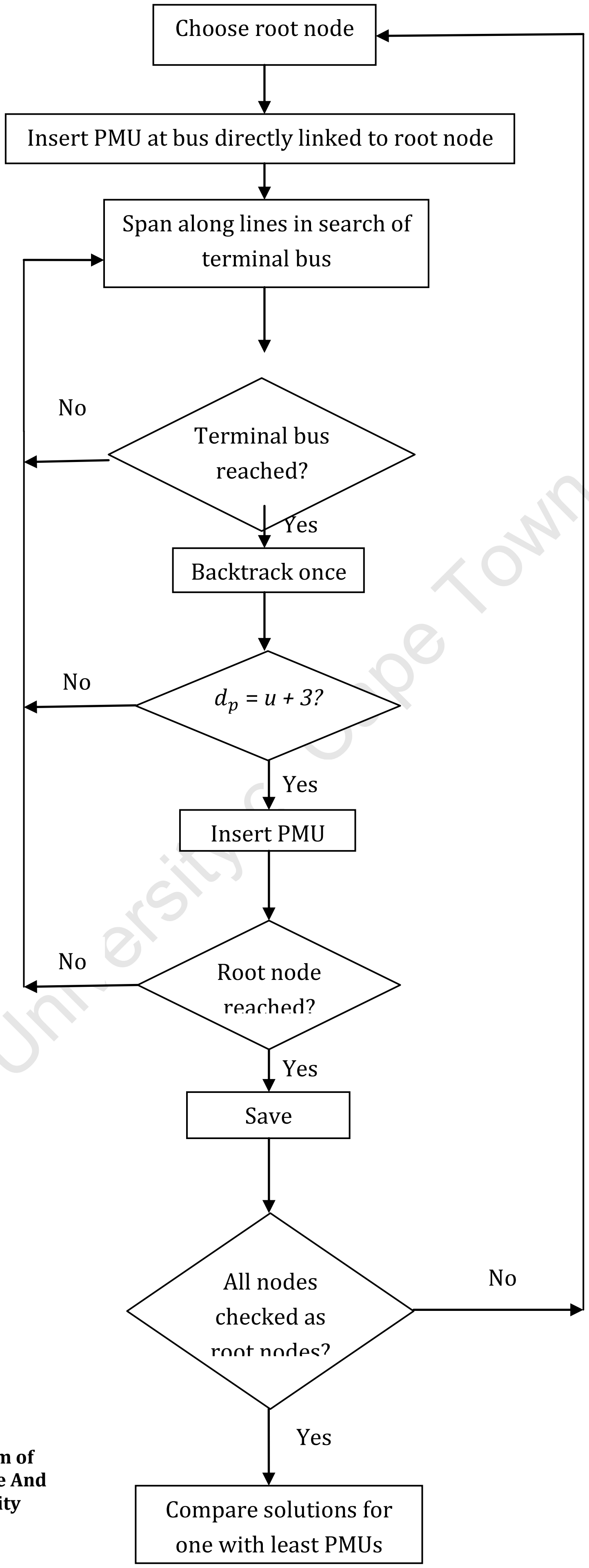

Figure 11: Flow diagram of Placement for complete And Incomplete Observability Algorithm [30]

2.8.5 The Immunity Genetic Algorithm for Optimal PMU placement

IGA is a uniquely adapted version of the Genetic Algorithm for Optimal PMU placement.

The genetic algorithm (GA) is based on the concept of the evolution of species in the changing environments they are exposed to. The characteristics are stored on the chromosomes and are updated when breeding as well as mutation of the species takes place.

The immunity algorithm (IA) is based on the idea of how an immune system that protects the body against viruses, diseases and bacterial bodies. This provides for a good optimiser by procedure of choosing a vaccine and immunity selection; where vaccination implies amending some genes to raise the fitness of the population. Immune selection is to prevent degenerative aspects in the evolution of the population by selecting the fittest individuals in the population [**30**].

IGA is made up of combined characteristics of the GA and of the IA for combined advantaged optimisation abilities.

IGA is governed by certain rules pertaining to the candidate PMU locations:

Rule 1

When a PMU is situated at a bus, it can observe the voltage and current phasors of that bus as well as the buses directly linked to that bus.

Rule 2

If there is at least one PMU directly linked to a zero-injection bus, all the buses directly linked to the zero-injection bus become observable by use of Kirchhoff's Current Law (KCL) making use of the current through the zero-injection bus.

Other rules fall under these two rules where they are applied according to the circumstance. The vaccines that are applied serve as limiters for the placement algorithm. They are mentioned below:

Vaccine1:

A PMU must not be placed at a radial bus. Placement at a radial bus is seen as a waste since it will only render the bus in question and the one bus directly linked to it observable.

Vaccine2:

The bus directly linked to a radial bus must be made a candidate PMU location if it is not a zero-injection bus.

Vaccine3:

PMUs should not be placed at zero-injection buses unless it provides the optimal solution if placed at that specific bus.

The formulation of the method is as follows [**31**]:

Step 1

Attain the connectivity matrix

This is the A matrix that shows which buses are linked in the system.

Step2

Generate the initial population

The initial population is made of a matrix with size: (N_{pop} x N). Where N_{pop} is the number of populations and N is the number of buses in the system.

Step3

Calculate the fitness of each individual of the population.

The fitness is given by the inverse of the following objective function [**30**] in equation (2-28):

$$\mathbf{C\ (x_i) = w1{*}N_{PMU} + w2{*}N_{Unobs}} \qquad \mathbf{2\text{-}28}$$

Where N_{PMU} is the number of PMUs in the system and N_{Unobs} is the number of unobservable buses in the system with w1 and w2 are tunable constants. The objective function must be minimised because it is desired that there be a minimal number of PMUs in the system with no unobservable buses.

Step4

Select the fittest individuals

Step5

Apply the crossover and mutation processes [**32**]. Crossover is the process of reproduction between two parents where two offspring are produced. Relating this to the PMU placement problem, each individual parent is represented as a possible placement solution. The offspring are new possible solutions determined by the genes of the parent. Mutation occurs randomly in nature. It is modelled with a mutation probability factor defined as: $\frac{1}{N}$. A random number between 0 and 1 is generated and compared to the mutation probability factor. If it is greater, then the individual's genes are inverted; if it not, then nothing happens.

Step6

Injection of vaccinations

The vaccination conditions mentioned above are applied to the system

Step7

Perform immune selection

Step 8

Close off

This is the final stage where this generation is now completed and the next generation will be examined.

Figure 12 shows a flowchart of the process:

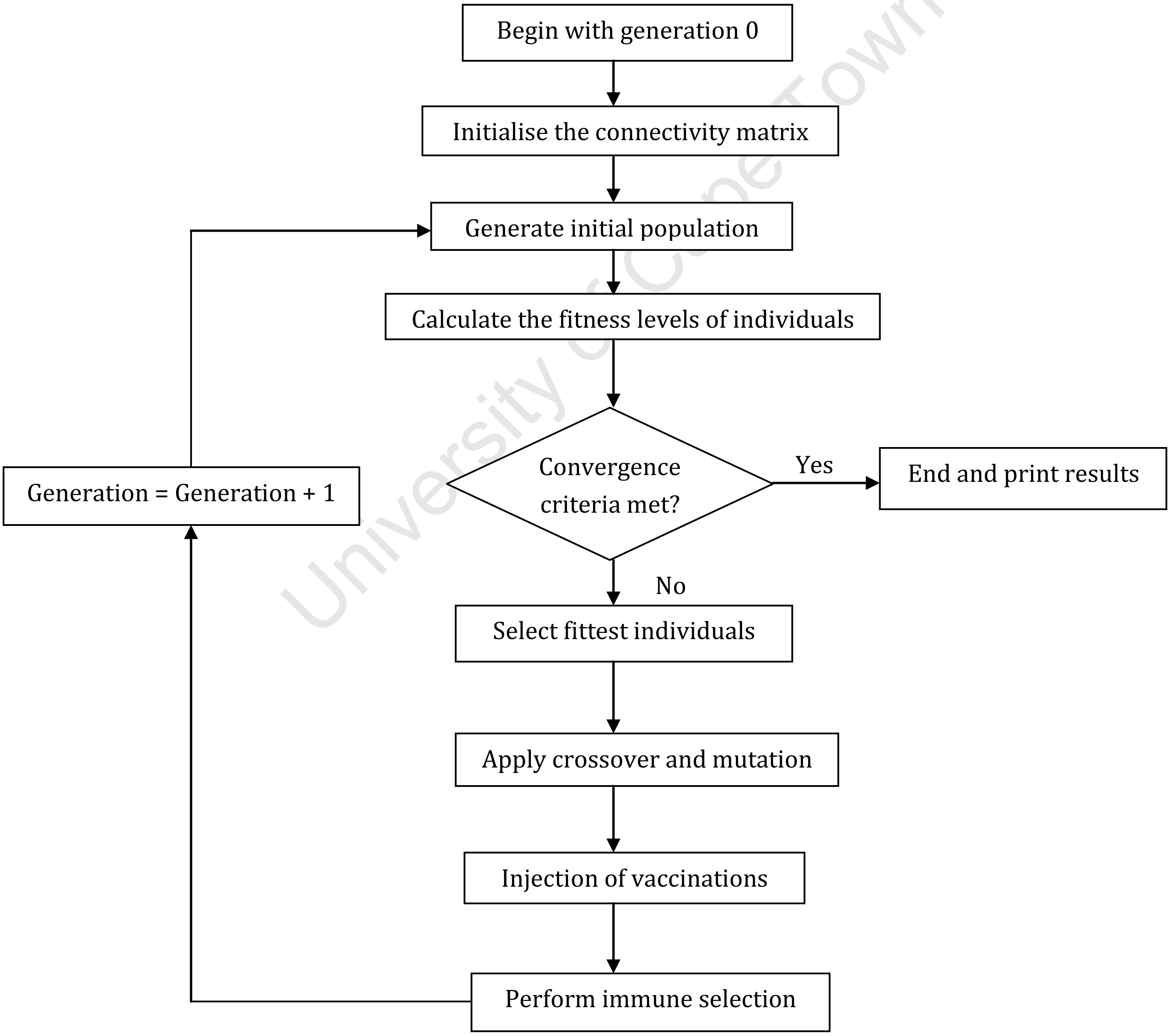

Figure 12: Flow diagram of IGA [31]

2.9 IEEE 14, 30 and 57 bus systems tested by the above reviewed algorithms

Table 3: Various algorithms applied to the IEEE 14 bus system

IEEE 14 bus system				
Algorithm	Zero-injection-buses	Zero-injection-bus positions	PMUs required	Positions of PMUs
Particle Swarm Optimisation	1	7	3	2,6,9
Immunity Genetic Algorithm	1	7	3	2,6,9
Binary Search Algorithm	1	7	3	2,6,9
Spanning Tree	1	7	3	2,6,9

Table 4: Various algorithms applied to the IEEE 30 bus system

IEEE 30 bus system				
Algorithm	Zero-injection-buses	Zero-injection-bus positions	PMUs required	Positions of PMUs
Particle Swarm Optimisation	6	6,9,22,25,27,28	7	2,3,10,12,18,24,27
Immunity Genetic Algorithm	6	6,9,22,25,27,28	7	1,5,10,12,1,24,30
Binary Search Algorithm	6	6,9,22,25,27,28	7	1,2,10,12,15,19,27
Spanning Tree	6	6,9,22,25,27,28	7	3,5,10,12,19,24,27

Table 5: Various algorithms applied to the IEEE 57 bus system

IEEE 57 bus system				
Algorithm	Zero-injection-buses	Zero-injection-bus positions	PMUs required	Positions of PMUs
Particle Swarm Optimisation	15	4,7,11,21,22,24,26,34,36,37,39,40,45,46,48	11	1,5,13,19,25,29,32,38,41,51,54
Immunity Genetic Algorithm	15	4,7,11,21,22,24,26,34,36,37,39,40,45,46,48	11	1,6,13,19,25,29,32,38,51,54,56
Binary Search Algorithm	15	4,7,11,21,22,24,26,34,36,37,39,40,45,46,48	11	1,6,13,19,25,29,32,38,51,54,56
Spanning Tree	15	4,7,11,21,22,24,26,34,36,37,39,40,45,46,48	11	1,5,13,19,25,29,32,38,41,51,54

It is observed in Tables 3, 4 and 5 above, that all of these algorithms produce exactly the same results for these three IEEE test bus systems. The bus locations may differ slightly, but are mostly the same with each algorithm. This therefore suggests that there is a generalised optimal set of locations for the PMUs for each system and that these algorithms are able to realise that solution.

It must be noted however, that these test systems are quite small. For larger systems with possibly more zero-injection buses and more radial buses etc., the algorithms solutions might differ more due to their optimisation logic focusing more on separate issues.

2.10 Summary of Algorithms

The algorithms discussed all have their own respective strengths and weaknesses. The Binary Particle Swarm method is good for small and larger systems where it gives the least number of required PMUs as shown in [**27**]. The ILP models some aspects with non-linear equations, although in [**24**] it is linearized, which simplified the placement problem and is now easily computable and gives results very quickly. The IGA as well as Binary swarm optimisation use statistical methods with discrete variables since they do not use differentials of the cost function [**27**]. The IGA is an improvement of the GA algorithm due to application vaccinations for issues pertaining to degeneration which speeds up the converging time. Binary search is an exhaustive method where it assures that the proper solution is found, but will take a while to converge for larger systems. Incomplete observability is good for dealing with systems where phasing of PMUs will take place. Despite their differences, they all have similar results for the optimal placement problem. Therefore when an algorithm is to be chosen, the size, nature and method of installation of PMUs must be considered.

2.11 Future Prospects for PMUs

PMUs are a rather recent development and their advantages and uses though evident, may not have reached their full potential as yet. Many institutions are therefore researching ways how PMUs can help improve the ways of managing power systems [**21**]. Bad Data Detection in State Estimation is hugely benefited by use of PMUs over RTUs. If power systems should have PMUs observing the entire networks, the network operations should be more efficient due to the benefits PMUs present. With the development of Synchronous vector processors (SVPs), State Estimation itself might be replaced by combinations of direct state vector measurements. Future Islanding Design for system restoration will be greatly influenced if PMUs are strategically placed to suite their requirements.

When one considers PMUs as sensing interfaces, then it suggests that there is a much larger scope for this technology to be used in other applications. SCADA technologies are used in various applications in industry. If PMUs can be developed to suite other important wide area monitoring systems such as for fresh water distribution, then these systems can possibly be made more efficient due to the GPS clock system which timestamps the data in a real-time format and synchronises them with respect to each other.

Hopefully PMUs can be made cheaper so that more PMUs can be installed in power systems for higher redundancy observability levels which would make analysis easier. If PMUs become cheap, there will be no more need for optimisation algorithms. For the time-being though, the placement strategies are in place and are still being studied to compensate for full system observability with minimum amount of installed PMUs as well as many other applications that may benefit the efficiency of power system operations.

3. Methodology

The Advanced Matrix Manipulation (AMM) algorithm deals with the optimisation of PMU placements for full observability of the power network using a graph theory and Boolean algebra approach and not a conventional constrained optimization approach. Full observability of any power network is a requirement for State Estimation. The AMM consists of the following algorithms for enhanced flexibility in performance:

- Standard Advanced Matrix Manipulation - For optimally placing PMUs for full observability
- Phasing of PMU Installations - For calculations of required number of PMU installation in phases due to phasing of procurement of PMUs by the utility
- Bad Data Detection Aid - Placing PMUs with desired measurement redundancy
- Islanding Detection - Placing PMUs for detection of Network Islands that may form in the event of faults or other network contingencies

All the algorithms of AMM are explained in detail in the thesis with the help of a 6-bus example system and flowcharts. The AMM algorithm is tested and validated by applying the same on IEEE 14, 30, 57 test systems and a typical South African 1009 bus utility system.

Figures 13, 14 and 15 represent the IEEE test systems, respectively. The 1009 bus system information was received from the local utility in CDF format. It is their wish that only certain aspects of the network be discussed while other aspects remain confidential. Therefore this system will not be presented in a figure as the IEEE systems are. The connectivity Matrices for these networks used in this thesis were derived from the network data and images of the IEEE systems. This is possible because only the basic topology of the network is required. The 1009 bus system data was derived from the information in the CDF file that was given to the student.

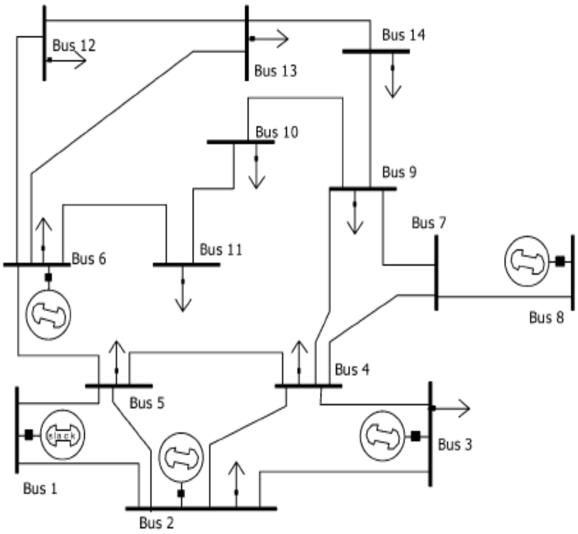

Figure 13: IEEE 14 bus configuration [28**]**

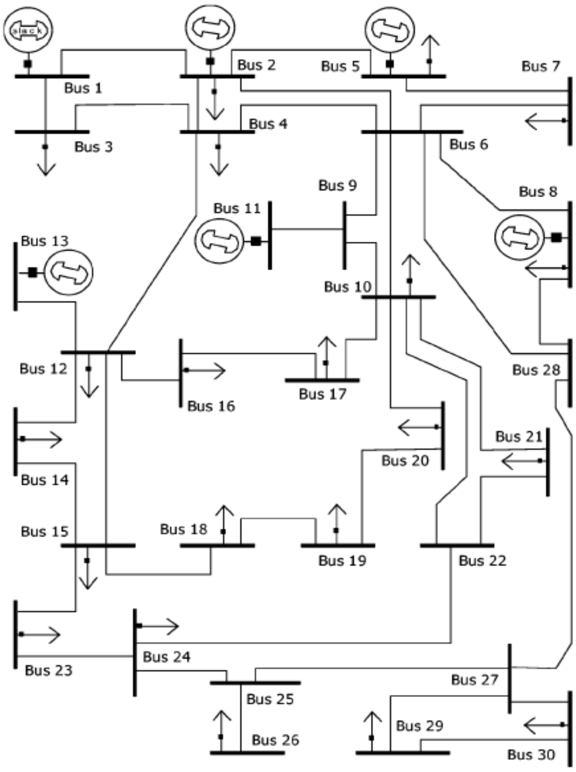

Figure 14: IEEE 30 bus configuration: [28**]**

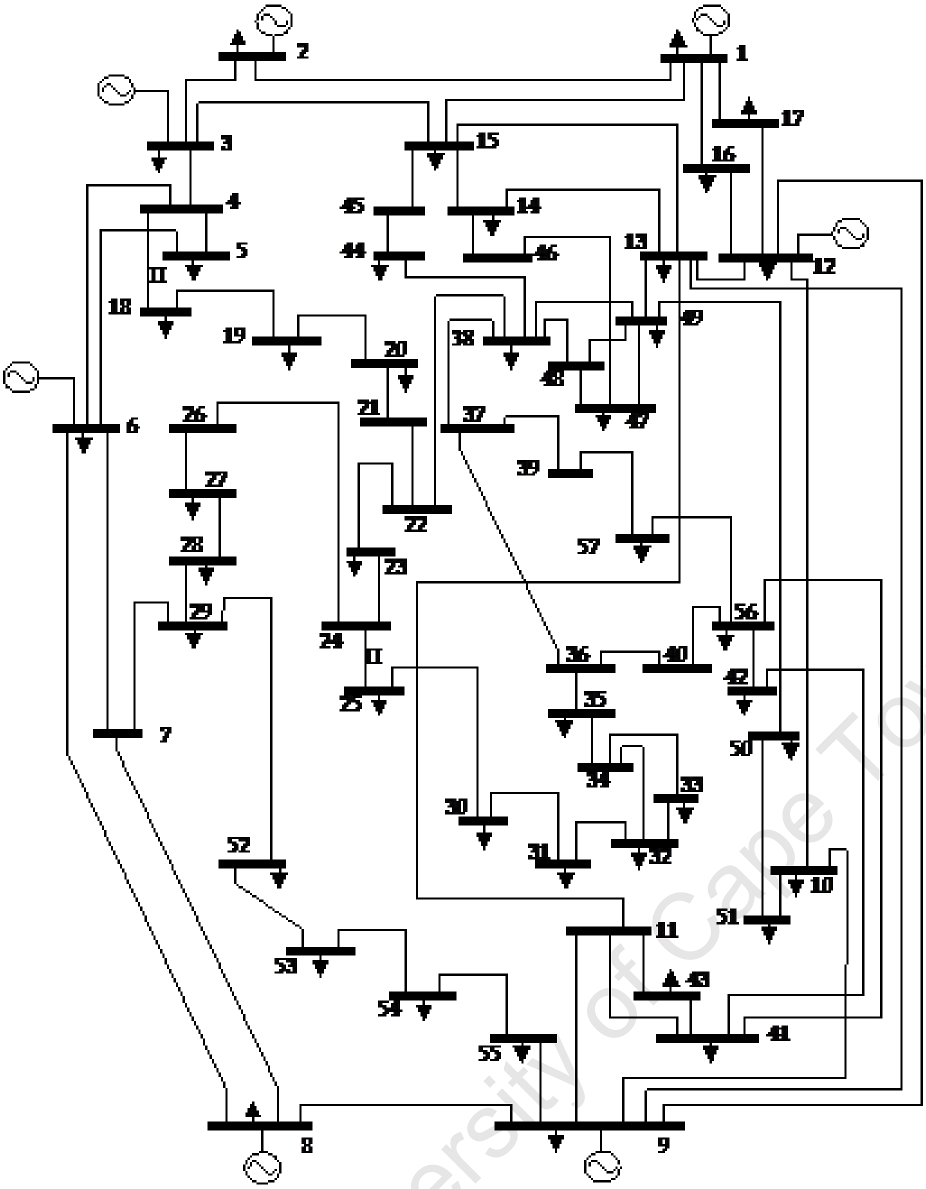

Figure 15 IEEE 57 Bus Configuration: [33]

The Advanced Matrix Manipulation Algorithm

3.1 The standard Advanced Matrix manipulation Algorithm

The algorithm uses a Boolean-Algebra-based method to converge to the optimal amount of PMUs necessary for full observability of the power network. It must be understood that this method is not a pure optimization type method as is with ILP, IGA, BPSO etc. reviewed in Chapter 2. Those methods use constraints on the connectivity matrix to converge to a solution. The Matrix Manipulation takes all the rows of the connectivity matrix and now AND's them together. This will leave a single row vector made up of binary elements representing the minimum binary ones for each row to have shared at least once. For the PMU placement problem presented in this thesis, this translates to finding the minimum amount of PMUs so that each row (representing each bus) is observed at least once by a PMU; where a PMU's presence is represented by a binary one.

This is demonstrated by using the connectivity matrix of the 6 bus system (from Figure 8) given below. The formation of the connectivity matrix from the bus configuration is explicitly explained in chapter 2.

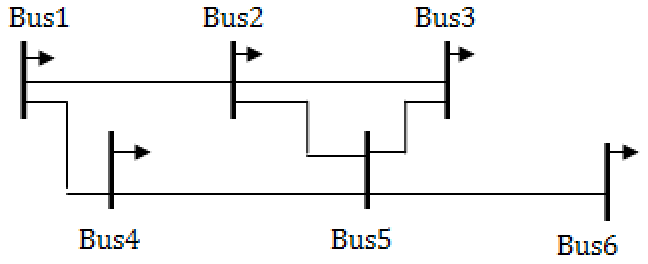

Figure 8: A simple 6-bus system

$$\begin{matrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{matrix}$$

When the logical AND operator is applied to row 1 and 2, the result would be:

$$\begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \end{matrix}$$

As can be noted, the AND operator removes all the dissimilar binary ones between the two rows. This result means that a PMU at bus 1 or 2 would render rows 1 and 2 observable. When row 1, 2 and 3 are ANDed together, the result is:

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 & 0 \end{matrix}$$

This shows that a PMU at bus 2 is necessary to observe row 1, 2 and 3. This calculation is done on all the rows to obtain a final minimum solution. The final solution will comprise locations of PMUs that make all the buses observable. The Matlab AND operator was used to for programming this algorithm.

Once all the rows are ANDed together, the result looks like:

0 1 0 1 1 0

This means that a total of 3 PMUs are required to observe all of the 6 buses and they are to be placed at buses 2, 4 and 5.

The algorithm is coded to enable the rows to be ANDed together with a memory function. This means that after the first step: (row1 AND row2), it will proceed to row3 by: ((row1 AND row2) AND row3) and then row 4 as: (((row1 AND row2) AND row3) AND row4). By doing this the algorithm continously checks back to see whether there existed a better binary combination in previous rows. The solution after each row is then the best combination of binary elements to give the least amount of binary ones to make all the rows computed so far, observable. Therefore the final solution is the best solution fitting all the rows.

This algorithm also addresses the advantage of further minimising PMUs through the presence of zero-injection buses. If a PMU is meant to be placed at a zero-injection bus, the algorithm will look if that specific bus is observed through an adjacent bus's PMU or through an adjacent zero-injection bus that is adjacently observed by a PMU. This also applies to normal loaded buses; to be observed through an observed zero-injection bus. Zero-injection buses are explained in Chapter 2 in more detail.

For the 6 Bus example system, bus 5 is a zero-injection bus. The algorithm is designed to determine if bus 5 was one of the buses in the minimum slection. It so happens that bus 5 is one of the optimal bus locations for a PMU. The algorithm checks to see if another PMU bus from the optimal selection observes bus 5, because if it does, then the PMU at bus 5 can be excluded. Bus 2 and 4 are seen to both observe bus 5, so the PMU at bus 5 is removed by setting the element of the row matrix corresponding to bus 5 to zero. The final solution after zero-injection bus consideration is:

0 1 0 1 0 0

The algorithm checks through all the known zero-injection buses in the system. After testing if a zero-injection bus can be removed from the solution, the algorithm goes through thorough checks to see if the rows are still fully observed by the remaining PMUs. If they are not, the PMU at the zero-injection bus cannot be removed.

As mentioned, the algorithm is programmed in Matlab and is setup in different modules; each dealing with different issues. There are 4 different modules and they will be explained below:

Setup of algorithm:

First module:

This is where all the user input is received. The user has to type the amount and the locations of the zero-injection buses. (For Larger systems where there are much more zero-injection buses such as the 1009 bus system, the list of zero injection buses is stored in the program since it is very tedious to insert them each time).

Second module:

This is the main algorithm module. The connectivity matrix is firstly defined. It is then put through the Boolean Algebra ANDing formulation as explained in section 3.1.1 (above). The final result of this module is a row vector of binary elements representing bus positions for PMU placements.

Third module:

This module's only purpose is to deal with the zero-injection buses. As mentioned above, zero-injection bus minimisation is used to observe buses connected to zero-injection buses. This is truly only an advantage if the specific zero-injection bus is observed by a PMU that is not placed at that specific zero-injection bus. This means that it is observed by a PMU adjacently placed, or through an adjacent zero-injection bus that is in turn observed by a PMU. Provision must be made for a maximum possibility of all the buses being zero-injection buses (this will probably never happen). This concept is possible since the network line parameters are all known and Kirchhoff's Current Law can be applied to determine the voltages of the buses beyond the zero-injection buses. If this function is properly utilized, the total number of necessary PMUs can be highly reduced. This reduction is separate to the optimisation which already took place in the Second Module (above) and will only really be significant in large power networks where there are zero-injection buses connected to one another.

This is best understood through Figure 16 given below.

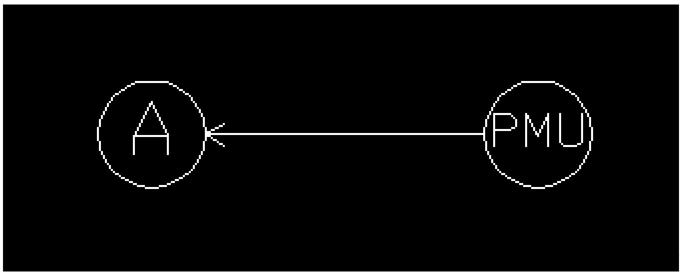

Figure 16: Normal Observation of a bus by a PMU

Figure 16 shows how a bus is normally observed by a PMU. The PMU can either be located one bus away or directly on bus A to make bus A observable.

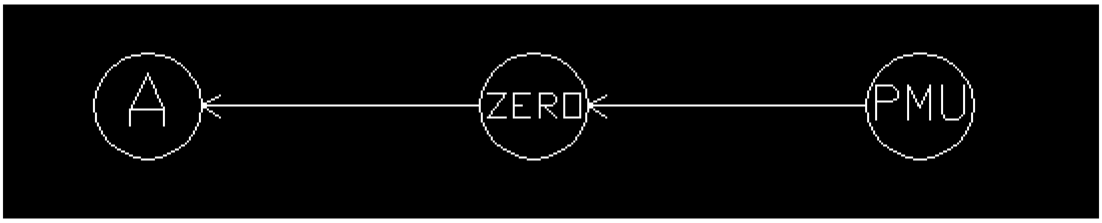

Figure 17: Bus A observed through by a PMU through a zero-injection bus

Figure 17 shows how optimization algorithms incorporate the Zero-Injection bus advantage. This is possible due to the line parameters being known.

Figure 18: Bus A observed through multiple zero-injection buses by the AMM algorithm

Figure 18 shows how the AMM algorithm allows multiple levels of zero-injection buses to be between the PMU bus and bus A. Again this is possible due to the line parameters being known from available power network data. Circumstances where there will be multiple levels of zero-injection buses exist in larger power networks such as the 1009 bus system used in this thesis.

Fourth module:

Here the final solution vector is displayed to the user.

Due to Matlab not having logical operators (required in the Second Module) to execute exactly what was required, a customized AND operation had to be programmed. In this process it was possible to approach the vectors from the right side or the left side of the ANDing vector. The two approaches produce different results; one with less PMUs than the other. This is due to the ANDing function addressing one row element at a time. Since it approaches from the left or from the right of the row, by chance it will occur that either approach will reach a specific element first, where this specific element was the best element to obtain a binary one or zero.
It cannot be predicted which approach will produce less PMUs and therefore both approaches must always be applied. Therefore the Advanced Matrix Manipulation Algorithm has two versions:

1. Right sided approach
2. Left sided approach

The AMM algorithm is tested and validated with the left and right sided approach using the IEEE 14, 30 and 57 test bus systems and a typical 1009 bus South African network. The results are presented in Chapter 4 of this thesis.

3.2 Phasing PMU installations with the Advanced Matrix Manipulation Algorithm

Due to the lack of immediate resources such as capital and communication networks, all the PMUs needed for full observability of the power network will not be able to be installed at once. Instead they will be deployed in different phases, as the resources become available. A phasing algorithm for PMU placements in each phase was worked on and the final algorithm for phasing placements has two variations, namely: “Phasing Installations for Outside Input” and “Phasing Installations for Inside Input”. A description of each algorithm is given below.

3.2.1 Phasing Installations for Inside Input

This algorithm is to be used when the network being dealt with has no associated critical buses relative to the other buses. In other words, all buses should be observed with the same level of importance and there exists no bus which specifically requires a PMU at that bus. The calculation engine is the same as for the standard AMM algorithm. This algorithm is a phase-based application which means that it calculates the buses required for the phase or stage of installation at hand.

The modifications include allowing the user to enter how many PMUs should be installed during the present phase. It also allows the user enter how many and where exactly PMUs were installed in all previous phases. The algorithm then uses this information for calculation of the best locations for the PMUs to be placed in the present phase.

The main idea of this algorithm and major difference to “Phasing Installations for Outside Input” is that since none of the buses are relatively critical, the only positions for PMU placements considered are the ones determined by the standard Advanced Matrix Manipulation calculation. An example the IEEE 14 bus calculation shows that it needs 3 PMUs at buses 2, 6 and 9 for full observability. “Inside Input” refers to a choice only from these 3 buses for placements during the phases e.g.: phase 1 install 1 PMU, then it will choose from 2,6 or 9 and phase 2 install 1 PMU, it will choose from the two positions left after the first phase’s choice.

The choice for the best position is based on the observability redundancy index. During every phase, it will choose the number of PMU positions from those that are left, which observe the most buses should a PMU be placed there. To do this, a “Redundancy meter” module is used to measure the redundancy of observability of the positions from the normal algorithm to determine which positions to use.

Once the positions have been calculated, they are finally printed for the user to see. For a calculation of the PMU placement positions of the next phase, the program must be re-run and information on past PMU placement numbers and positions must be entered.
A flow diagram of the steps followed by the algorithm is found below:

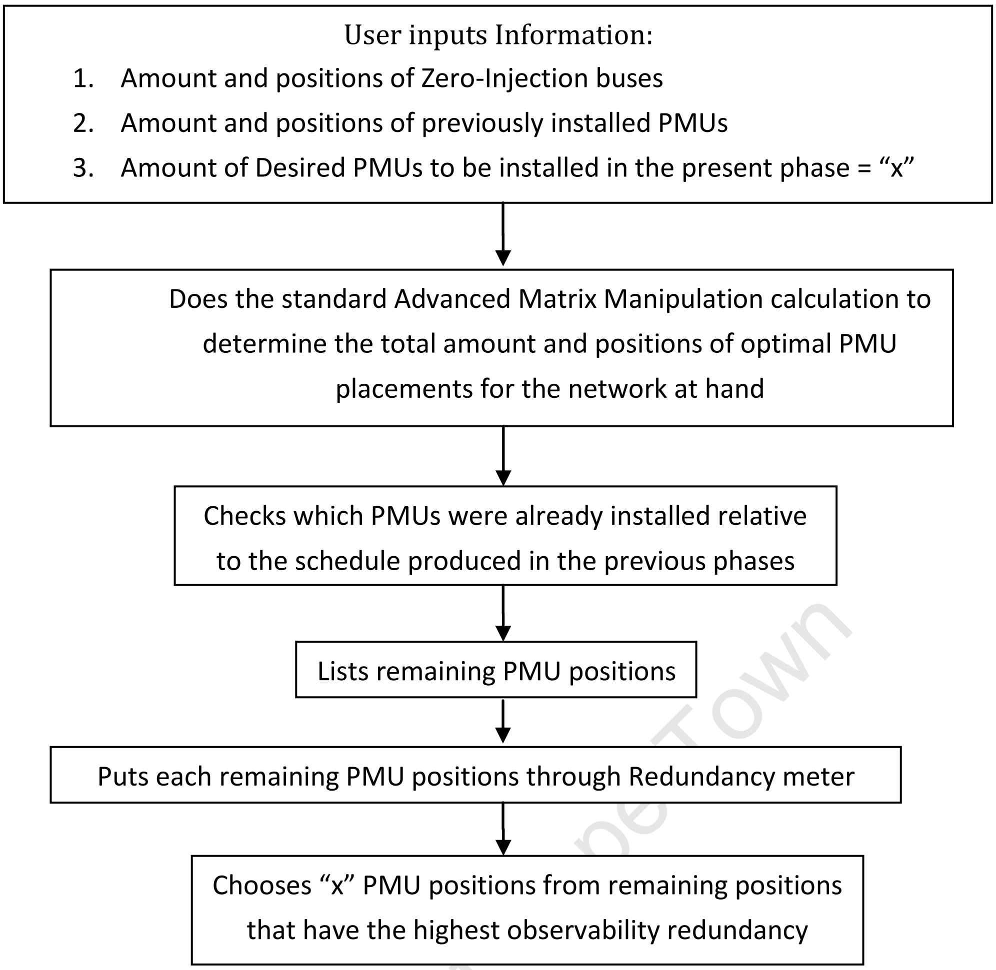

Figure 19 Flow Diagram of Advanced Matrix Manipulation for Inside Input

3.2.2 Phasing Installations for Outside Input

This Algorithm must be used for any of the following two reasons:

1. There exists critical buses with respect to the other buses
2. The phasing program has begun on this specific power network, where the Matrix Manipulation algorithm was not necessarily used

For application 1, it is required that certain buses receive PMUs despite these specific buses not necessarily being part of the group of buses that would generate the fewest possible amount of PMUs calculated by the standard Advanced Matrix Manipulation algorithm.

In application 2, another method might have been used to calculate the minimum number and positions of PMU placements which might not be the same as the standard Advanced Matrix Manipulation algorithm’s placement solution. So when the next phase of installation is determined via Matrix Manipulation for Outside Input, it will fully consider the previous positions.

Both of these applications require the handling of the possibility of placing PMUs at positions other than the optimal positions determined by the standard Advanced Matrix Manipulation. Therefore this version of the placement algorithm is different to “Phasing Installations for Inside Input” whereby the positions of the PMUs for the present phase is not calculated, but is inserted by the user. The advantage lies within how the algorithm is used. It uses the inputs from the user since the user is defining unique PMU positions. This algorithm then calculates how many and what the positions are for the remaining PMUs required to make the network fully observable, which can be installed in the following phase/s.

This version uses the same inputs from the user as placement for Inside Input such as how many zero-injection buses there are and where to find them, how many PMUs were previously placed and where to find them, but now also includes how many PMUs to be installed during this phase and where to find them. This version will then take this data and assess it thoroughly to determine positions for placements for the following phase. An important point is that the user defined buses need not be the actual entire present phase list of PMU locations. The algorithm calculates which PMU positions are required in addition to this input. Therefore the PMUs can be chosen from the additional list in conjunction with the user input for the present phase.

One might be troubled with the thought about the situation when only one critical bus is required in the first phase which allows for 4 PMUs and might be thinking “if bus A is critical, what positions do I use for the other 3 PMUs B, C and D?” This is however not a problem. To get those other positions, first it is suggested to run the program normally in “Phasing Installations for Inside Input” for an allowance of 3 PMUs. We will now have a list of 3 PMU locations which we can use in the “Phasing Installations for Outside Input” algorithm. Note that if one of those 3 locations includes the critical bus, then redo the “Phasing Installations for Inside Input” step for 4 PMUs. Now we will definitely have space for the critical bus as well as buses B, C and D.

The output of this algorithm will provide all the necessary PMU positions remaining to provide full observability for the Power network. These positions can be used in the next phase evaluation.

A flow diagram for the algorithm steps is shown below:

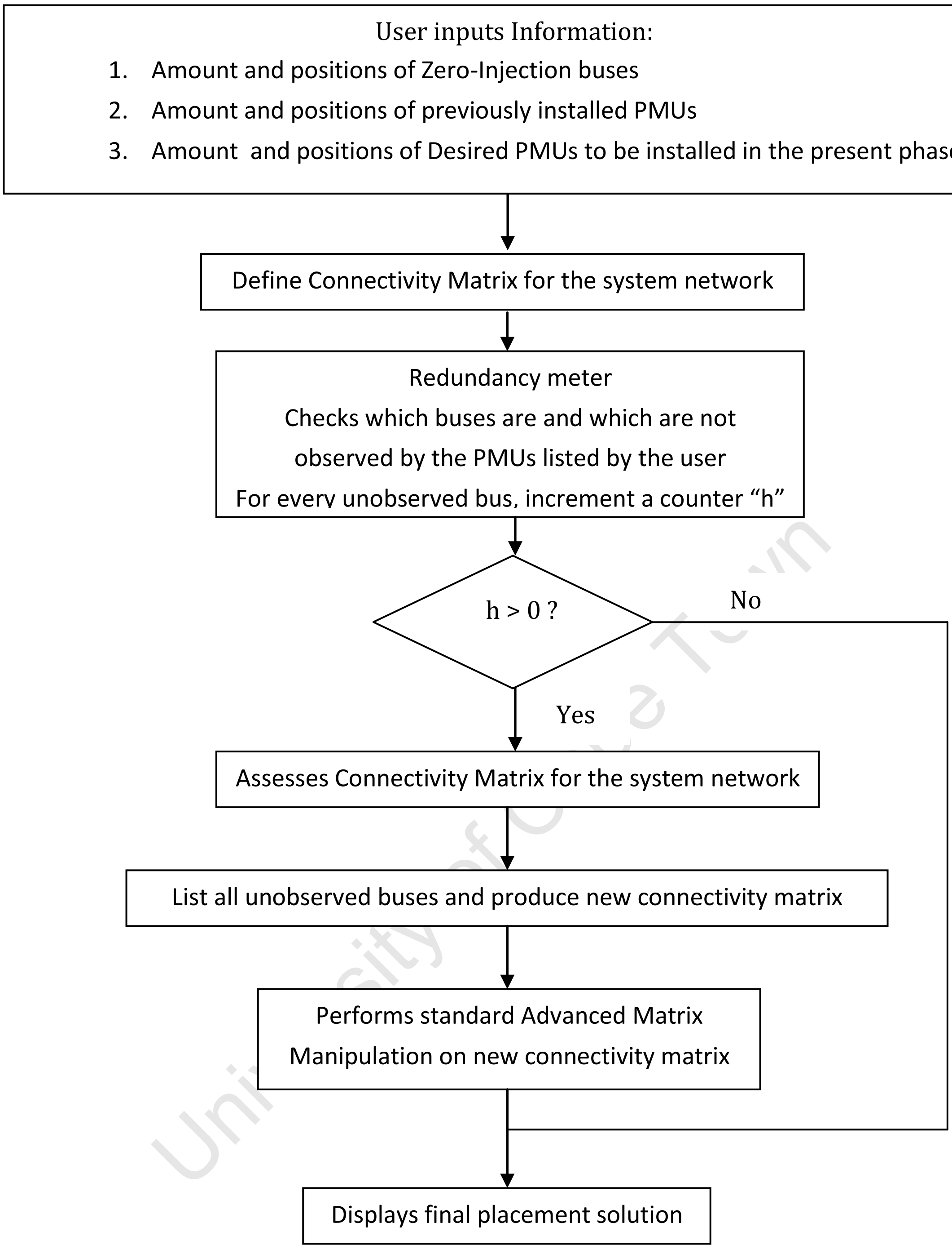

Figure 20 Flow diagram of Advanced Matrix Manipulation for Outside Input

This Algorithm will be applied to the IEEE 14, 30 and 57 bus systems as well as to the 1009 bus South African network case file. The algorithms abilities will be displayed with different inputs buses and different amounts of phases. The solutions will be displayed in the results section.

3.3 Bad Data detection Aid

This placement process is intended to make the buses of the power network observed with a level of redundancy set by the user. This will render the amount of necessary PMUs more than for the standard Advanced Matrix manipulation's minimum amount required for full observability (which caters for at least one PMU observing each bus).

It must be noted that Terminal buses, which are connected to only one other bus, may have a maximum of two PMUs observing it. This case is to be realised when the level of observability redundancy calls for more than two PMUs observing a single bus (this is however not likely). If a bus is observed by only one PMU, this bus is considered a critical bus [**34**]. A critical bus is one whose measurement cannot be reconfirmed as there is only one measuring system available to this bus. Therefore if at least two PMUs were to be measuring each bus, there would be no remaining critical buses in the system.

The algorithm first calculates the minimum number of PMUs for full observability as done by the standard Advanced Matrix manipulation. From this result, a redundancy meter is applied to determine the level of observability of each bus due to the minimum PMU placement. This only includes buses observing zero-injection buses to a depth of 1 (not PMUs at buses observing zero-injection buses, which in turn observe other zero-Injection buses) due to complications that arise. Once this is complete, the buses with insufficient level of observability are recorded. These are then used to determine the minimum extra PMUs required for the required redundancy of observability levels. The final solution is made up of the minimum PMU positions plus the new positions.

The nature of Bad data detection is due to problems with the measurement devices providing incorrect data. This algorithm compensates for measurement device outages and not necessarily branch outages. This is explained through an example (refer to figure 21):

If bus A is set to be observed by two PMUs (none of which are to be installed on Bus A), and bus A is connected to one zero-injection bus, it can be allowed that one or both of these PMUs are not directly connected to bus A and observe bus A through the same zero-injection bus. This means that Bus A can be observed by choosing 2 positions from Buses B, D and F. If branch outages were to be the focus, then the two PMUs observing bus A cannot be allowed to observe bus A through the same zero-injection bus, as there will be no measurement of bus A provided if the zero-injection bus was to be inoperative. Bus A will have to be observed by either Bus B, D or F in conjunction with a PMU at Bus E.

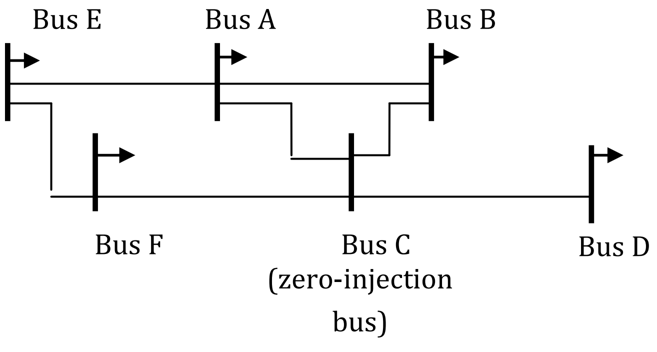

Figure 21 System to demonstrate branch outages

Branch outages are not addressed in this thesis because they are not problems caused by Bad Data itself. If there is a situation where it is required that critical buses need to be made immune to branch outages, this algorithm can be applied in conjunction with the *Phasing placements for Outside inputs* algorithm where it is allowed for the user to add PMUs wherever they are necessary. If one really wants to make all the buses completely safe from branch outages, then all buses should have a PMU. This could become a long term goal for when ample communication infrastructure and more cost effective PMU technology becomes available.

This algorithm is applied to the IEEE 14, 30 and 57 bus test systems as well as the 1009 bus utility case-file. The solutions are presented in the Results section.

3.4 Islanding detection with PMUs

3.4.1 Unintentional Island detection

Unintentional detection requires the placements of PMUs to be at least according to an optimal placement for the minimum amount of PMUs (for 1 PMU observing each bus) providing full observability of the power network. If a bus is energised during a power outage, the PMU observing that bus should pick it up and this can trigger an alert signal for an unintentional islanding condition to have occurred, due to knowledge provided by other PMUs (or the lack of knowledge) that the grid section had indeed collapsed. This will be able to give the location of the energised buses and thus of the island.

It is recommended that the Standard AMM algorithm – for optimal placement for full observability be used. This should provide positions of the PMUs necessaryfor detection of unintentional islands.

To be a degree more confident in the information read by the PMUs, it may be better to use the *Bad Data Detection Aid* algorithmas well. This will be able to reconfirm the information that the Grid section had actually collapsed due to redundant measurements. The fact remains that all the buses should remain observed whether or not the grid collapses; even for the case of unintentional Islanding.

3.4.2 Intentional Island detection

For the fastest restoration period after a system collapse, a maximum amount of islands must be used; this means that islands should be as small as possible [**18**]. According to [**18**] there are certain criteria for each island to meet for power restoration purposes:

1. Each island must have at least one DG
2. The buses on either side of a single transformer must be in the same island; this makes each island independent of another
3. The loading in that island must be compensated for by the generation sources, otherwise islands must overlap to neighbouring islands with DG sources capable to provide at least the lack of energy provided by existing DG sources
4. All the buses in the island must be observed by measurement devices within that island

It must be noted that the *Islanding-for-restoration* procedure is not centred about the placement of PMUs. This is an area of research by itself and for the purposes of this research, PMU placements; provision will be made for a solution for that specific purpose.

There are three recommendations for placing PMUs for *Islanding-for-restoration*:

1. Initially a normal placement for minimum PMUs providing full network observability using *Standard AMM* algorithm should be approached. The Islanding method must be applied taking into consideration the specific positions of the PMUs. The positions of the PMUs will considerably affect the location of the islands and determine if these islands are valid according to the criteria mentioned.

2. Another approach is to start off with the *Standard AMM* algorithm and get the optimal PMU locations. Then choose any size islands that use all the buses of the network. Now check if the buses are observed within their respective islands. If not all the buses are observed within each island, apply Outside Input AMM algorithm by using the existing PMU in the respective island as the user input and to change the connectivity matrix to just include the island's buses. The result would be the locations of the remaining PMUs required for that Island.

3. This approach allows the islands to be established officially by disregarding the criteria area number 4. The PMU placement is left for last and will be according to where the islands are. The PMUs will be placed to make each island completely observable, independent of PMUs of any other island. One could think of this as working backwards, however it is not hard to determine the positions for the PMUs. The *Standard AMM* algorithm can be used; where the difference now is that the connectivity matrix will be different for each island. The matrix will comprise the buses within that specific island and the algorithm will determine the optimal locations for PMUs to make that island fully observable. This approach gives maximum flexibility to the islanding algorithms but may increase the amount of PMUs relative to the above two approaches because it is probable that in this instance there would be more islands to deal with, since one of the major constraints was removed.

3.5 Conclusion

The Standard AMM algorithm was coded in attempt to achieve similar or better results than than commonly used algorithms mentioned in Chapter 2. The other algorithms, namely *Phasing for Inside Input, Phasing for Outside Input and Islanding detection*, were designed and coded due to the question "What will your algorithm do in this situation?" being addressed to the student. Debugging of this code was a major task, which caused the design concepts of the algorithms to be changed many times from the initial design.

4. Results

This chapter presents the results obtained from the simulation of each algorithm described in Chapter 3 for all the test systems namely IEEE 14 bus, 30 bus and 57 bus and a 1009 bus typical South African utility system. The order of presentation of these results in this chapter is given below.

Order of Results

The results are presented in the following order of algorithms:

1) Standard Advanced Matrix Manipulation results
2) Advanced Matrix Manipulation for Inside Input results
3) Advanced Matrix Manipulation for Outside Input results
4) Advanced Matrix Manipulation for Bad Data Aid results
5) Advanced Matrix Manipulation for Islanding Detection results

All of the results are presented in tabular form. In this tabular form, there will be an ILP comparison result for section 1. Sections 1 and 5 include separate sections for considerations of and for no consideration of zero-injection buses. All versions of the Advanced Matrix Manipulation have Left sided and Right sided versions, which produce different results to each other.

The full result from each algorithm simulation was too lengthy to put in the thesis write-up, but can be viewed in Appendix file on the CD, in the same order the Results chapter follows.

List of Zero-Injection buses for the test systems

Before the results are shown, one must be made aware of the zero-injection nature of the IEEE test bus systems and the utility case-file system. The Standard Advanced Matrix Manipulation and Bad Data Detection algorithms are applied to these systems with and without consideration for the zero-injection buses. When they are considered, their advantage of lowering the amount of PMUs required (mentioned above in Zero-injection bus section) is used. When they are not considered, the zero-injection bus is treated like a normal bus that has loading, generation or both.

Below is a list of each set of zero-injection buses for each respective system.

IEEE 14 bus test System:

Number of Zero-Injection buses:	1
Bus numbers:	7

IEEE 30 bus test System:

Number of Zero-Injection buses:	6

Bus numbers: 6, 9, 22, 25, 27, 28

IEEE 57 bus test System:

Number of Zero-Injection buses: 15
Bus numbers: 4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48

1009 bus System from a typical South African utility's case-file:

Information of this system was received from a utility provider in an IEEE CDF format file. The purpose of using a large network is to put the algorithms under a higher degree of calculation stress.

Zero-Injection buses: 347

Bus numbers are given in Table 6 below:

Table 6: Zero Injection buses in the 1009 bus system

1	81	168	275	352	409	501	606	675	712	767	806	872	923
2	84	169	276	353	410	509	607	683	713	768	808	873	924
4	85	190	277	354	411	534	608	684	715	770	809	874	925
5	86	192	278	355	412	556	614	685	717	773	810	882	934
6	87	193	279	356	416	557	632	686	721	777	811	883	954
10	88	194	280	358	417	558	637	688	722	783	812	884	956
13	89	195	281	359	421	559	640	689	725	787	815	885	959
33	90	196	282	360	423	560	641	690	726	788	816	886	960
34	98	198	285	378	424	561	642	691	727	789	839	891	963
35	105	199	286	379	425	562	643	692	728	790	840	894	964
36	119	203	290	380	426	563	644	693	730	791	841	895	979
37	120	206	294	381	427	585	645	694	731	792	842	896	980
38	146	220	322	382	428	586	646	695	737	793	843	897	982
39	147	223	324	383	429	587	647	696	738	794	845	905	994
42	148	256	327	384	430	588	650	697	739	795	846	907	996
56	149	257	332	395	435	589	651	698	740	796	850	908	998
66	150	258	334	396	441	590	664	699	741	797	851	909	999
67	151	259	335	400	492	593	665	700	747	798	856	915	1000
68	152	260	336	401	493	594	666	701	750	799	858	916	1005
69	153	261	337	402	494	598	667	703	751	800	862	917	1006
70	154	263	338	403	495	599	669	704	756	801	863	918	1007
71	155	264	347	404	496	600	670	706	761	802	868	919	1008
72	156	267	349	406	497	603	671	707	762	803	869	920	
73	160	268	350	407	498	604	672	709	765	804	870	921	
76	165	269	351	408	499	605	673	710	766	805	871	922	

4.1 The Simulation of the Standard Advanced Matrix Manipulation Algorithm

4.1.1 IEEE 14 Bus test System Optimal Placement Calculation

Without considering Zero-Injection buses

Approach from Left side algorithm

Table 7 shows that for full observability of the IEEE 14 bus test System, when not considering the zero-injection buses and approaching the vector from the left side, 4 buses require PMUs and the bus positions are:

Bus 2, 7, 11 and 13

Table 7: Standard AMM 14 bus, Without considering Zero-Injection buses, approach from the left

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
4	2, 7, 11, 13	7	0.116688 seconds	4	2, 6, 7, 9

Approach from Right side Algorithm

Table 8 shows that for full observability of the IEEE 14 bus test System, when not considering the zero-injection buses and approaching the vector from the right side, 4 buses require PMUs and the bus positions are:

Bus 2, 6, 7 and 9

Table 8: Standard AMM 14 bus, Without considering Zero-Injection buses, starting from the right

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
4	2, 6, 7, 9	7	0.087556 seconds	4	2, 6, 7, 9

Considering Zero-Injection buses

Approach from Left side algorithm

Table 9 shows that for full observability of the IEEE 14 bus test System, when considering the zero-injection buses and approaching the vector from the left side, 4 buses require PMUs and the bus positions are:

Bus 2, 7, 11 and 13

Table 9: Standard AMM 14 bus, Considering Zero-Injection buses, approach from the left

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
4	2, 7, 11, 13	7	0.105782 seconds	3	2, 6, 9

Approach from Right side Algorithm

Table 10 shows that for full observability of the IEEE 14 bus test System, when considering the zero-injection buses and approaching the vector from the right side, 3 buses require PMUs and the bus positions are:

Bus 2, 6 and 9

Table 10: Standard AMM 14 bus, Considering Zero-Injection buses, approach from the right

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
3	2, 6, 9	7	0.080977 seconds	3	2, 6, 9

4.1.2 IEEE 30 Bus test System Optimal Placement Calculation

Without considering Zero-Injection buses

Approach from Left side algorithm

Table 11 shows that for full observability of the IEEE 30 bus test System, when not considering the zero-injection buses and approaching the vector from the left side, 10 buses require PMUs and the bus positions are:

Buses 3, 5, 6, 9, 10, 12, 15, 20, 25 and 27

Table 11: Standard AMM 30 bus, Without considering Zero-Injection buses, approach from the left

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
10	3, 5, 6, 9, 10, 12, 15, 20, 25, 27	6, 9, 22, 25, 27, 28	0.198079 seconds	10	1, 2, 6, 9, 10, 12, 15, 18, 25, 27

Approach from Right side Algorithm

Table 12 shows that for full observability of the IEEE 30bus test System, when not considering the zero-injection buses and approaching the vector from the right side, 10 buses require PMUs and the bus positions are:
Buses 1, 2, 6, 9, 10, 12, 15, 20, 25 and 27

Table 12: Standard AMM 30 bus, Without considering Zero-Injection buses, approach from the right

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
10	1, 2, 6, 9, 10, 12, 15, 20, 25, 27	6, 9, 22, 25, 27, 28	0.197479 seconds	10	1, 2, 6, 9, 10, 12, 15, 18, 25, 27

Considering Zero-Injection buses

Approach from Left side algorithm

Table 13 shows that for full observability of the IEEE 30 bus test System, when considering the zero-injection buses and approaching the vector from the left side, 6 buses require PMUs and the bus positions are:

Bus 3, 5, 10, 12, 15 and 20

Table 13: Standard AMM 30 bus, Considering Zero-Injection buses, approach from the left

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
6	3, 5, 10, 12, 15, 20	6, 9, 22, 25, 27, 28	0.121228 seconds	7	1, 2, 10, 12, 15, 18, 27

Approach from Right side Algorithm

Table 14 shows that for full observability of the IEEE 30 bus test System, when considering the zero-injection buses and approaching the vector from the right side, 6 buses require PMUs and the bus positions are:

Bus 1, 2, 10, 12, 15 and 20

Table 14: Standard AMM 30 bus, Considering Zero-Injection buses, approach from the right

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
6	1, 2, 10, 12, 15, 20	6, 9, 22, 25, 27, 28	0.121178 seconds	7	1, 2, 10, 12, 15, 18, 27

4.1.3 IEEE 57 Bus test System Optimal Placement Calculation

Without considering Zero-Injection buses

Approach from Left side algorithm

Table 15 shows that for full observability of the IEEE 57 bus test System, when not considering the zero-injection buses and approaching the vector from the left side, 19 buses require PMUs and the bus positions are:
Buses 1, 6, 10, 14, 15, 19, 22, 26, 29, 30, 34, 36, 37, 39, 41, 45, 49, 52 and 55

Table 15: Standard AMM 57 bus, Without considering Zero-Injection buses, approach from the left

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
19	1, 6, 10, 14, 15, 19, 22, 26, 29, 30, 34, 36, 37, 39, 41, 45, 49, 52, 55	4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48	0.444353 seconds	17	1, 4, 9, 15, 20, 24, 28, 29, 30, 34, 38, 40, 41, 46, 50, 54, 57

Approach from Right side Algorithm

Table 16 shows that for full observability of the IEEE 57 bus test System, when not considering the zero-injection buses and approaching the vector from the right side, 20 buses require PMUs and the bus positions are:

Buses 1, 6, 10, 14, 15, 18, 21, 24, 25, 26, 29, 32, 36, 37, 39, 41, 45, 49, 52 and 55

Table 16: Standard AMM 57 bus, Without considering Zero-Injection buses, approach from the right

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
20	1, 6, 10, 14, 15, 18, 21, 24, 25, 26, 29, 32, 36, 37, 39, 41, 45, 49, 52, 55	4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48	0.466806 seconds	17	1, 4, 9, 15, 20, 24, 28, 29, 30, 34, 38, 40, 41, 46, 50, 54, 57

Considering Zero-Injection buses

Approach from Left side algorithm

Table 17 shows that for full observability of the IEEE 57 bus test System, when considering the zero-injection buses and approaching the vector from the left side, 16 buses require PMUs and the bus positions are:

Bus 1, 6, 10, 14, 15, 19, 21, 24, 29, 30, 34, 36, 41, 49, 52 and 55

Table 17: Standard AMM 57 bus, Considering Zero-Injection buses, approach from the left

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
16	1, 6, 10, 14, 15, 19, 21, 24, 29, 30, 34, 36, 41, 49, 52, 55	4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48	0.394520 seconds	12	1,9,15,20,28,29,30,38,41,50,54,57

Approach from Right side Algorithm

Table 18 shows that for full observability of the IEEE 57 bus test System, when considering the zero-injection buses and approaching the vector from the left side, 15 buses require PMUs and the bus positions are:
Bus 1, 6, 10, 14, 15, 18, 21, 25, 29, 32, 36, 41, 49, 52 and 55

Table 18: Standard AMM 57 bus, Considering Zero-Injection buses, approach from the right

Required PMUs with AMM	Buses with PMUs for AMM	Zero-Injection Bus locations	Computation time	Required PMUs with ILP	Buses with PMUs for ILP
15	1, 6, 10, 14, 15, 18, 21, 25, 29, 32, 36, 41, 49, 52, 55	4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48	0.380099 seconds	12	1,9,15,20,28,29,30,38,41,50,54,57

4.1.4 1009 bus Typical South African Utility System Optimal placement Calculation

Without considering Zero-Injection buses

Table 19 shows the results obtained from the ILP algorithm, when not considering zero-injection buses. It calculated that 282 PMUs are necessary for full observability. The bus Positions are given below.

Table 19: ILP RESULTS FOR NO ZERO-INJECTION CONSIDERATION

TABLE OF ILP RESULTS FOR NO ZERO-INJECTION CONSIDERATION										
Number of required PMUs	Bus Numbers									
282	2	126	237	338	451	559	646	772	879	997
	7	132	239	348	456	560	647	775	881	1001
	8	136	240	349	459	563	665	785	901	1002
	10	137	241	353	462	565	668	795	905	
	13	140	251	354	465	567	670	797	909	
	18	141	253	356	468	569	671	810	911	
	19	147	256	360	471	570	676	811	913	
	22	148	263	362	475	575	683	812	916	
	25	150	264	372	479	579	684	813	918	
	27	151	269	374	482	584	685	815	919	
	31	152	270	382	488	586	687	816	921	
	35	154	272	383	491	587	694	822	925	
	37	160	275	384	492	591	700	826	936	
	38	165	277	385	494	598	704	832	940	
	42	168	278	390	501	599	706	834	943	
	43	189	279	393	505	603	708	837	946	
	50	198	280	398	510	608	711	842	949	
	52	200	281	400	512	613	714	844	952	
	57	201	282	401	514	619	716	847	954	
	60	203	285	410	516	624	721	857	956	
	66	206	290	411	519	626	732	859	957	
	67	212	291	417	524	628	734	862	964	
	68	216	298	419	528	630	742	863	966	
	69	220	299	422	529	636	747	864	973	
	70	223	306	431	532	637	751	867	974	
	81	224	314	432	536	639	755	869	978	
	85	226	318	433	544	640	762	871	981	
	86	227	327	434	546	642	763	873	982	
	88	229	334	435	550	643	768	875	985	
	91	233	335	446	554	644	770	876	988	
	94	236	336	447	556	645	771	877	994	

Approach from Left side algorithm

Table 20 below shows the results of the Standard Advanced matrix manipulation Left sided algorithm for no zero-injection bus compensation. A total of 320 PMUs were calculated as necessary for full observability of the system.

Table 20: The Standard Advanced MM Left Sided Results For No Zero-Injection Bus Consideration

Number of buses	BUS NUMBERS							
320	2	156	266	363	496	631	787	902
	12	157	267	365	499	639	788	904
	21	160	270	367	502	645	794	906
	27	165	272	371	504	646	796	908
	32	166	274	374	506	651	799	913
	36	167	279	375	508	653	802	914
	37	169	281	376	511	654	805	916
	47	170	283	387	520	658	817	920
	50	171	285	389	522	660	822	924
	59	173	286	390	524	661	823	933
	62	177	290	391	529	662	824	937
	70	178	291	394	534	679	825	940
	71	181	293	402	535	683	826	943
	73	183	299	403	536	686	829	944
	87	189	300	404	537	693	830	945
	89	193	301	409	540	698	831	949
	90	195	302	414	541	699	837	951
	92	197	312	417	543	705	840	955
	94	205	313	422	546	709	844	956
	97	206	314	424	551	711	847	958
	99	207	317	425	556	712	850	959
	100	208	319	433	558	715	851	961
	102	216	322	434	565	718	854	965
	103	217	323	438	567	720	856	968
	108	218	326	439	571	724	857	969
	111	219	329	442	574	726	858	970
	123	232	330	443	575	728	861	971
	127	236	331	445	580	734	865	978
	128	239	335	451	589	737	869	980
	130	240	336	455	596	743	870	981
	133	242	338	461	597	753	874	983
	134	247	345	466	598	754	881	985
	136	248	346	467	599	755	883	986
	137	249	348	474	600	761	884	997
	139	253	350	478	601	763	891	998
	142	254	353	480	610	766	895	1000
	147	255	354	488	615	768	897	1001
	148	257	358	490	616	771	898	1005
	151	260	360	493	618	775	899	1006
	153	264	362	495	627	783	900	1007

Approach from Right side Algorithm

Table 21 below shows the results of the Standard Advanced matrix manipulation Right sided algorithm for no zero-injection bus compensation. A total of 313 PMUs were calculated as necessary for full observability of the system.

Table 21: Standard Advanced MM Right Sided Results For No Zero-Injection Bus Consideration

Number of buses	BUS NUMBERS							
313	1	135	267	392	515	648	743	887
	3	137	270	398	516	650	748	890
	5	141	271	404	520	651	749	896
	9	142	272	407	524	654	751	898
	11	143	274	408	526	655	753	902
	12	149	276	409	527	656	756	904
	16	150	277	410	531	658	764	905
	18	151	279	413	534	659	769	907
	23	160	285	419	539	662	770	910
	24	162	287	424	540	668	774	913
	31	163	292	428	541	670	777	920
	32	164	293	429	545	673	778	924
	33	165	305	435	550	676	782	931
	39	170	307	437	554	677	785	932
	41	171	308	449	555	678	788	938
	44	172	311	450	558	681	794	939
	50	173	313	457	569	682	798	944
	55	174	315	461	573	685	799	950
	56	178	318	463	574	686	800	952
	57	179	319	466	576	687	802	953
	58	184	320	469	583	688	806	955
	64	192	326	471	585	689	808	956
	71	194	329	475	586	698	810	958
	76	196	332	478	589	700	814	959
	79	199	335	481	590	702	819	969
	81	201	341	482	592	707	821	979
	82	203	343	483	599	709	830	982
	83	209	345	484	603	712	836	987
	92	211	355	485	604	713	837	988
	93	213	356	490	609	717	839	996
	95	223	360	492	611	721	845	1004
	96	232	363	493	612	722	846	1005
	98	234	366	494	613	724	858	1008
	100	237	370	495	620	725	866	
	104	238	371	500	624	728	871	
	108	255	373	501	631	735	872	
	109	256	374	503	635	737	874	
	117	258	379	504	643	739	875	
	119	259	383	509	644	740	879	
	128	260	385	511	646	741	880	

Considering Zero-Injection buses

Table 22 below shows the results obtained from the ILP simulation. A total of 226 PMUs were calculated to be necessary for full system observability when considering zero-injection buses. The bus positions are given in the table below.

Table 22: ILP RESULTS FOR ZERO-INJECTION CONSIDERATION

Table of ILP Results For Zero-Injection Consideration								
Number of Required PMUs	Bus Numbers							
226	7	201	314	459	565	700	857	978
	8	203	318	462	567	704	859	981
	13	206	336	465	569	706	862	982
	18	212	338	468	570	708	863	985
	19	216	348	471	575	711	864	988
	22	220	354	475	579	714	867	994
	25	223	356	479	584	716	875	997
	27	224	360	482	587	732	876	1001
	31	226	362	488	591	734	877	1002
	35	227	372	491	598	742	879	
	38	229	374	492	599	747	881	
	43	233	383	501	603	751	901	
	50	236	384	505	613	755	905	
	52	237	385	510	619	762	911	
	57	239	390	512	624	763	913	
	60	240	393	514	626	770	918	
	86	241	398	516	628	771	919	
	91	251	400	519	630	772	925	
	94	253	401	524	636	775	936	
	126	256	411	528	639	785	940	
	132	270	417	529	644	797	943	
	136	272	419	532	645	813	946	
	137	275	422	536	647	815	949	
	140	278	431	544	665	816	952	
	141	279	432	546	668	822	954	
	147	281	433	550	671	826	956	
	148	290	434	554	676	832	957	
	165	291	446	556	683	834	964	
	189	298	447	559	685	837	966	
	198	299	451	560	687	844	973	
	200	306	456	563	694	847	974	

Approach from Left side algorithm

Table 23 below shows the results of the Standard Advanced matrix manipulation Left sided algorithm with zero-injection bus compensation. A total of 203 PMUs were calculated as necessary for full observability of the system.

Table 23: Standard Advanced MM Left Sided Results With Zero-Injection Consideration

NUMBER OF BUSES	BUS NUMBERS					
203	12	208	367	540	824	986
	21	216	371	541	825	997
	27	217	374	543	826	1001
	47	218	375	546	829	
	50	219	376	551	830	
	59	232	387	565	831	
	62	236	389	567	837	
	92	239	390	571	844	
	94	240	391	574	847	
	97	242	394	575	854	
	99	247	414	580	861	
	100	248	422	596	865	
	102	249	433	597	881	
	103	253	434	601	898	
	108	254	438	610	899	
	111	255	439	616	900	
	123	266	442	618	902	
	127	270	443	631	906	
	128	272	445	639	913	
	130	274	451	653	914	
	133	283	455	654	933	
	134	291	461	660	937	
	136	293	466	661	940	
	137	299	467	679	943	
	139	300	474	705	944	
	142	301	478	711	945	
	157	302	480	718	949	
	166	312	488	720	951	
	167	314	490	724	955	
	170	317	502	734	958	
	171	319	504	743	961	
	173	323	506	753	965	
	177	326	508	754	968	
	178	329	511	755	969	
	181	330	520	763	970	
	183	331	524	771	971	
	189	345	529	775	978	
	197	346	535	817	981	
	205	348	536	822	983	
	207	362	537	823	985	

Approach from Right side Algorithm

Table 24 below shows the results of the Standard Advanced matrix manipulation Right sided algorithm with zero-injection bus compensation. A total of 187 PMUs were calculated as necessary for full observability of the system.

Table 24: Standard Advanced MM Right Sided Results With Zero-Injection Consideration

NUMBER OF BUSES	BUS NUMBERS				
187	3	174	385	574	821
	9	178	392	576	830
	11	179	398	583	836
	12	184	413	592	866
	16	201	419	609	875
	18	209	437	612	879
	23	211	450	613	880
	24	213	457	620	887
	31	232	461	624	890
	32	234	463	631	898
	41	237	466	635	904
	44	238	469	648	910
	50	255	471	654	913
	55	270	475	655	931
	57	271	478	656	932
	58	272	481	658	938
	64	274	483	659	939
	82	287	484	662	944
	83	292	485	668	950
	92	293	490	676	952
	93	305	500	677	953
	95	307	503	678	955
	96	308	504	681	958
	100	311	511	682	969
	104	313	515	687	987
	108	315	516	702	988
	109	318	520	724	1004
	117	319	524	735	
	128	320	526	743	
	135	326	527	748	
	137	329	531	749	
	141	341	539	753	
	142	343	540	764	
	143	345	541	769	
	162	363	545	774	
	163	366	550	778	
	164	370	554	782	
	170	371	555	785	
	171	373	569	814	
	172	374	573	819	

4.2 The Simulation of the Advanced Matrix Manipulation for Inside Input Algorithm

In this section, different amounts of phases and different amounts of PMUs which is chosen by the user, will be used for each test system. The aim is to show how the algorithm selects the highest observing PMU location group each time from the list of total minimum PMU positions calculated by the standard AMM for each respective system.

4.2.1 IEEE 14 Bus test System Phased Optimal PMU Placement Calculation for Inside Input

Considering Zero-Injection buses

Approach from Left side algorithm

This test will involve two phases of PMU installation. In the first phase two PMUs will be installed and another two in the second phase (As we already know from above that the total PMU positions for this algorithm is four). Table 25 and Table 26 summarise the results for AMM inside input considering zero injection buses with approach from the left.

For phase 1 it shows that the best 2 PMU positions are at Buses 2 and 7.

After the algorithm is run again for phase 2, it is shown that the next best two PMU positions are at buses 11 and 13. One could already have guessed this answer if the total optimal positions were made apparent.

Table 25: 14 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 1

Number of PMUs desired in phase 1	Total Amount of Necessary PMUs	Phase 1 PMU buses	Further PMUs required at the following buses
2	4	2, 7	11, 13

Table 26: 14 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 2

Number of PMUs desired in phase 2	Total Amount of Necessary PMUs	Phase 2 PMU buses	Further PMUs required at the following buses
2	4	11, 13	0

Approach from Right side Algorithm

Here there will again be two phases, wherein phase 1 2 PMUs will be placed and 1 PMU will be placed in phase 2. This comes from the knowledge that for this systems respective solution, only 3 PMUs are required. Table 27 and Table 28 summarise the results for AMM inside input considering zero injection buses with approach from the right.

It is shown that in phase 1 the best two positions for PMU placements are at Buses 2 and 9.
After the next calculation for phase 2, it is shown that the remaining PMU be placed at bus number 6 for full observability.

Table 27: 14 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 1

Number of PMUs desired in phase 1	Total Amount of Necessary PMUs	Phase 1 PMU buses	Further PMUs required at the following buses
2	3	2, 9	6

Table 28: 14 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 2

Number of PMUs desired in phase 2	Total Amount of Necessary PMUs	Phase 2 PMU buses	Further PMUs required at the following buses
1	3	6	0

4.2.2 IEEE 30 Bus test System Phased Optimal PMU Placement Calculation for Inside Input

Considering Zero-Injection buses

Approach from Left side algorithm

For this system where originally 6 PMUs are required, 3 stages of placements will commence with 2 PMUs placed in each phase. Table 29, Table 30 and Table 31 summarise the results for AMM inside input considering zero injection buses with approach from the left.

In phase 1, the first two PMU positions are calculated to be at buses 10 and 12.
In phase 2, the PMU positions were calculated to be at buses 3 and 15.
For phase 3 i.e. the final phase of installation, it was calculated that the final two PMUs be placed at buses 5 and 20.

Table 29: 30 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 1

Number of PMUs desired in phase 1	Total Amount of Necessary PMUs	Phase 1 PMU buses	Further PMUs required at the following buses
2	6	10, 12	3, 5, 15, 20

Table 30: 30 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 2

Number of PMUs desired in phase 2	Total Amount of Necessary PMUs	Phase 2 PMU buses	Further PMUs required at the following buses
2	6	3, 15	5, 20

Table 31: 30 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 3

Number of PMUs desired in phase 3	Total Amount of Necessary PMUs	Phase 3 PMU buses	Further PMUs required at the following buses
2	6	5, 20	0

Approach from Right side Algorithm

It was originally calculated that 6 PMUs be used in this system. Three phases of installation will be applied with two PMUs installed in each phase. Table 32, Table 33 and Table 34 summarise the results for AMM inside input considering zero injection buses with approach from the right.

For phase 1, the first two PMUs are seen to be placed at buses 2 and 10.
In phase 2, the second round of PMUs are placed at buses 12 and 15.
In phase 3, the final round of PMUs will be placed at buses 1 and 20.

Table 32: 30 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 1

Number of PMUs desired in phase 1	Total Amount of Necessary PMUs	Phase 1 PMU buses	Further PMUs required at the following buses
2	6	2, 10	1, 12, 15, 20

Table 33: 30 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 2

Number of PMUs desired in phase 2	Total Amount of Necessary PMUs	Phase 2 PMU buses	Further PMUs required at the following buses
2	6	12, 15	1, 20

Table 34: 30 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 3

Number of PMUs desired in phase 3	Total Amount of Necessary PMUs	Phase 3 PMU buses	Further PMUs required at the following buses
2	6	1, 20	0

4.2.3 IEEE 57 Bus test System Phased Optimal PMU Placement Calculation for Inside Input

Considering Zero-Injection buses

Approach from Left side algorithm

Originally for this system, 16 buses were calculated to include a PMU. Here 3 phases of installation will be used with PMUs installed in amounts 6, 6 and 4 respectively. Table 35, Table 36 and Table 37 summarise the results for AMM inside input considering zero injection buses with approach from the left.

After phase 1 calculation, the 6 best positions for PMU placements are at buses 1, 6, 10, 15, 41 and 49.
After phase 2 calculation, the next 6 positions for PMU placements are at buses 14, 19, 24, 29, 34 and 36.
After phase 3 calculation, the last 4 positions for PMU placements are at buses 21, 30, 52 and 55.

Table 35: 57 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 1

Number of PMUs desired in phase 1	Total Amount of Necessary PMUs	Phase 1 PMU buses	Further PMUs required at the following buses
6	16	1, 6, 10, 15, 41, 49	14, 19, 21, 24, 29, 30, 34, 36, 52, 55

Table 36: 57 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 2

Number of PMUs desired in phase 2	Total Amount of Necessary PMUs	Phase 2 PMU buses	Further PMUs required at the following buses
6	16	14, 19, 24, 29, 34, 36	21, 30, 52, 55

Table 37: 57 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 3

Number of PMUs desired in phase 3	Total Amount of Necessary PMUs	Phase 3 PMU buses	Further PMUs required at the following buses
4	16	21, 30, 52, 55	0

Approach from Right side Algorithm

Here it was originally calculated that 15 PMUs are required in this system. Again 3 phases will be used with 5 PMUs installed per phase. Table 38, Table 39 and Table 40 summarise the results for AMM inside input considering zero injection buses with approach from the left.

After phase 1 calculation, the following buses require PMUs: Buses 1, 6, 15, 18 and 49.
After phase 2 calculation, the following buses require PMUs: Buses 10, 14, 29, 32 and 41.
After phase 3 calculation, the following buses require PMUs: Buses 21, 25, 36, 52 and 55.

Table 38: 57 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 1

Number of PMUs desired in phase 1	Total Amount of Necessary PMUs	Phase 1 PMU buses	Further PMUs required at the following buses
5	15	1, 6, 15, 18, 49	10, 14, 21, 25, 29, 32, 36, 41, 52, 55

Table 39: 57 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 2

Number of PMUs desired in phase 2	Total Amount of Necessary PMUs	Phase 2 PMU buses	Further PMUs required at the following buses
5	15	10, 14, 29, 32, 41	21, 25, 36, 52, 55

Table 40: 57 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 3

Number of PMUs desired in phase 3	Total Amount of Necessary PMUs	Phase 2 PMU buses	Further PMUs required at the following buses
5	15	21, 25, 36, 52, 55	0

4.2.4 1009 bus typical South African Utility case Phased Optimal PMU Placement Calculation for Inside Input

Considering Zero-Injection buses

Approach from Left side algorithm

Table 41, 42 and 43 show the results for the First phase of PMU Installation for Inside Input using the Left sided Advanced MM algorithm. A total of 60 PMUs were installed in phase 1.

Table 41: 1009 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 1

NUMBER OF PMUS	PHASE1 BUSES	
60	27	461
	62	474
	94	478
	100	506
	128	511
	130	543
	134	580
	157	596
	167	610
	171	631
	181	705
	205	734
	207	755
	217	771
	255	822
	266	824
	299	825
	317	829
	323	847
	329	898
	375	902
	376	906
	389	951
	390	955
	422	961
	438	965
	442	969
	443	970
	451	978
	455	986

Phase 2 also comprised a choice of 60 PMUs to be installed. They are given below:

Table 42: 1009 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 2

NUMBER OF PMUS	PHASE 2 BUSES	
60	12	394
	59	439
	92	466
	97	490
	99	502
	102	508
	103	520
	127	524
	136	535
	137	565

	142	574
	178	575
	216	616
	219	639
	232	654
	239	661
	249	711
	291	720
	293	724
	319	753
	330	844
	331	899
	345	900
	346	937
	348	944
	362	945
	371	971
	374	981
	387	985
	391	1001

Phase 3 installs the remaining 83 PMUs as shown in bus numbers in Table 43.

Table 43: 1009 Bus AMM Inside Input Considering Zero-Injection buses, approach from the left, phase 3

NUMBER OF PMUS	PHASE 3 BUSES	
83	21	302
	47	312
	50	314
	108	326
	111	367
	123	414
	133	433
	139	434
	166	445
	170	467
	173	480
	177	488
	183	504
	189	529
	197	536
	208	537
	218	540
	236	541
	240	546
	242	551
	247	567
	248	571
	253	597

	254	601
	270	618
	272	653
	274	660
	283	679
	300	718
	301	743
		754
		763
		775
		817
		823
		826
		830
		831
		837
		854
		861
		865
		881
		913
		914
		933
		940
		943
		949
		958
		968
		983
		997

Approach from Right side Algorithm

Table 44, 45 and 46 show the results for the First phase of PMU Installation for Inside Input using the Right sided Advanced MM algorithm. A total of 60 PMUs were installed in phase 1.

Table 44: 1009 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 1

NUMBER OF PMUS	PHASE1 BUSES	
60	3	478
	9	483
	11	503
	18	504
	24	516
	31	524
	57	526
	93	539
	100	545

	104	550
	135	569
	142	583
	170	609
	174	624
	178	655
	201	656
	209	659
	232	681
	237	735
	238	748
	255	774
	271	778
	305	782
	366	879
	373	904
	419	910
	450	913
	457	938
	461	939
	471	1004

Phase 2 also installed 60 PMUs as shown in the bus numbers given in Table 45

Table 45: 1009 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 2

NUMBER OF PMUS	PHASE 2 BUSES	
60	16	437
	32	463
	44	475
	55	481
	58	485
	82	511
	95	515
	96	520
	109	527
	117	531
	128	541
	141	554
	162	573
	163	592
	164	612
	171	648
	211	682
	213	724
	272	743
	274	764
	287	785

	292	814
	308	821
	311	875
	319	880
	341	890
	363	931
	370	944
	385	952
	413	969

Phase 3 installs the remaining 67 PMUs in bus numbers as given in Table 46.

Table 46: 1009 Bus AMM Inside Input Considering Zero-Injection buses, approach from the right, phase 3

NUMBER OF PMUS	PHASE 3 BUSES	
67	12	469
	23	484
	41	490
	50	500
	64	540
	83	555
	92	574
	108	576
	137	613
	143	620
	172	631
	179	635
	184	654
	234	658
	270	662
	293	668
	307	676
	313	677
	315	678
	318	687
	320	702
	326	749
	329	753
	343	769
	345	819
	371	830
	374	836
	392	866
	398	887
	466	898
		932
		950

		953
		955
		958
		987
		988

It must be realised that there lots of other combinations of phase amounts and PMU amounts in each respective phase. The simulations serve as a demonstration of how the algorithm would provide a solution in each of these cases. It is to be noted that for this algorithm if there are more PMUs desired to be placed in a particular phase than the total number of PMU positions calculated by the standard Advanced Matrix Manipulation (for full system observability), the algorithm will just return the same solution as the standard algorithm. This is the limit set on this algorithm, because it is only meant to be used in the situation when there are no preferences above the minimal positions calculated by the standard algorithm. If one has preferences, then by all means proceed to use the algorithm presented in the following section, *Advanced Matrix Manipulation for Outside Input algorithm.*

4.3 Simulation of the Advanced Matrix Manipulation for Outside Input Algorithm

This algorithm will be presented differently from the Inside Input phasing algorithm. Instead of going through each phase of installation, only the first stage of installation will be addressed. The reason for this is that one of this algorithm's abilities is to do phasing calculations, but the main ability which is to be focused on, is how for any random amount and positions of PMUs (as long as this random amount and positions do not in the rare case actually compensate for the entire system's observability) can be inserted and the algorithm will determine the following:

- which buses are still unobserved
- how many PMUs are required in total
- the positions of all of these PMUs
- the positions of the PMUs to be installed in the following phases to render a fully observable system

The purpose of the random amount and positions is to allow clients to specify their requirements where PMUs are to be installed. Once the requirements are met, the observability of the rest of the system will be addressed where the optimal minimum amount of PMUs will be positioned. This algorithm is set such that if the complete Outside Input schedule of PMU positions calculated plus the user defined positions is more than the positions determined by the standard AMM algorithm plus the user defined positions, then the result will be that the PMUs to be installed in the following phase are at those locations selected from the standard algorithm solutions (after the user defined PMU positions have been used). If the standard algorithm's solution plus the user defined locations has more PMUs than the Outside Input complete schedule plus the user defined locations, then the locations for the following phase will be chosen from the Outside Input solution (after the user defined PMU positions have been used). Both solutions are however correct. One just requires less PMUs.

4.3.1 IEEE 14 Bus test System Phased Optimal PMU Placement Calculation for Outside Input

Considering Zero-Injection buses

Approach from Left side algorithm

Originally 4 PMUs were required for this system. The first phase is addressed and 2 PMUs are chosen to be placed at buses 4 and 3. The output of the Algorithm is given in Table 47.

A total of 5 PMUs are required (including those at buses 4 and3).

Table 47: 14 Bus AMM Outside Input with zero bus injection consideration, approach from left

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard AMM	Bus positions of standard PMUs
2	4, 3	5	1, 6, 10, 11, 12, 13, 14	5, 11, 13	4	2, 7, 11, 13

Approach from Right side Algorithm

Originally 3 PMUs were necessary for this system. For the first phase of installation, 2 PMUs will be placed at buses 4 and 3. The same inputs are chosen as in the left side algorithm in order for a comparison to be made. The output of the algorithm is given in Table 48.

A total of 5 PMUs are required including those at buses 4 and 3.

Table 48: 14 Bus AMM Outside Input with zero bus injection consideration, approach from right

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
2	4, 3	5	1, 6, 10, 11, 12, 13, 14	2, 6, 9	3	2, 6, 9

4.3.2 IEEE 30 Bus test System Phased Optimal PMU Placement Calculation for Outside Input

Considering Zero-Injection buses

Approach from Left side algorithm

For this system, originally 6 PMUs were required. A random choice of 4 PMUs will be placed at buses 3, 12, 17 and 25. The result is given in Table 49 below.

It shows that a total of 8 PMUs are required including the user defined ones.

Table 49: 30 Bus AMM Outside Input with zero bus injection consideration, approach from left

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
4	3, 12, 17, 25	8	2, 5, 6, 7, 8, 9, 11, 18, 19, 20, 21, 22, 23, 28, 29, 30	5, 10, 15, 20	6	3, 5, 10, 12, 15, 20

Approach from Right side Algorithm

For this system, originally 6 PMUs were required. A random choice of 4 PMUs will be placed at buses 3, 12, 17 and 25. The result is given in Table 50 below.

It shows that a total of 8 PMUs are required including the user defined ones.

Table 50: 30 Bus AMM Outside Input with zero bus injection consideration, approach from right

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
4	3, 12, 17, 25	8	2, 5, 6, 7, 8, 9, 11, 18, 19, 20, 21, 22, 23, 28, 29, 30	2, 10, 15, 18	6	1, 2, 10, 12, 15, 20

4.3.3 IEEE 57 Bus test System Phased Optimal PMU Placement Calculation for Outside Input

Considering Zero-Injection buses

Approach from Left side algorithm

For this system, originally 16 PMUs were required. A random choice of 9 PMUs will be placed at buses 5, 7, 10, 25, 34, 35, 38, 41 and 49. The result is given in Table 51 below.

It shows that a total of 19 PMUs are required including the user defined ones.

Table 51: 57 Bus AMM Outside Input with zero bus injection consideration, approach from left

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
9	5, 7, 10, 25, 34, 35, 38, 41, 49	19	1, 2, 14, 15, 16, 17, 19, 20, 21, 23, 26, 27, 28, 31, 32, 33, 39, 40, 45, 46, 52, 53, 54, 55, 57	1, 15, 20, 27, 32, 46, 53, 54, 56, 57	16	1, 6, 10, 14, 15, 19, 21, 24, 29, 30, 34, 36, 41, 49, 52, 55

Approach from Right side Algorithm

For this system, originally 16 PMUs were required. A random choice of 9 PMUs will be placed at buses 5, 7, 10, 25, 34, 35, 38, 41 and 49. The result is given in Table 52 below.

It shows that a total of 18 PMUs are required including the user defined ones.

Table 52: 57 Bus AMM Outside Input with zero bus injection consideration, approach from right

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
9	5, 7, 10, 25, 34, 35, 38, 41, 49	18	1, 2, 14, 15, 16, 17, 19, 20, 21, 23, 26, 27, 28, 31, 32, 33, 39, 40, 45, 46, 52, 53, 54, 55, 57	1, 14, 15, 19, 27, 32, 53, 54, 57	15	1, 6, 10, 14, 15, 18, 21, 25, 29, 32, 36, 41, 49, 52, 55

4.3.4 1009 bus typical South African Utility System Phased Optimal PMU Placement Calculation for Outside Input

In this section the Desired PMU Bus Positions comprise of the following 60 buses:

Table 53: selected buses for Outside Input for 1009 bus system

Number of PMUs	Bus Numbers	
60	19	437
	20	491
	33	507
	42	510
	44	511
	60	521
	85	544
	123	608
	138	610
	196	611
	204	669
	216	675
	217	687
	220	699
	268	717
	305	721
	311	726
	312	739
	335	742
	339	815
	343	826
	344	829
	359	847
	362	876
	395	881
	396	930
	400	966
	406	979
	426	991
	427	1001

Considering Zero-Injection buses

Approach from Left side algorithm

Originally the Standard AMM Left sided zero-injection consideration algorithm calculated that 203 PMUs are required in total. After this algorithm was applied, 197 additional PMUs are calculated as a requirement for full observability. In total this amounts to 257 PMUs. The result is shown in Table 54.

Table 54: 1009 Bus AMM Outside Input with zero bus injection consideration, approach from left

Number of PMUs	Bus Numbers				
197	5	256	430	651	823
	15	257	436	658	825
	16	258	439	659	828
	26	259	461	672	829
	35	260	476	674	834
	37	262	478	676	844
	40	268	480	685	853
	42	273	484	691	858
	56	274	492	695	859
	60	279	498	703	866
	70	280	506	704	867
	73	286	507	709	871
	86	290	509	712	872
	90	294	518	719	877
	95	304	521	724	878
	97	307	527	727	888
	99	308	530	736	898
	101	320	531	740	902
	114	322	537	742	905
	129	324	539	745	907
	133	329	541	747	920
	134	331	549	749	923
	147	332	551	752	929
	161	340	561	755	935
	174	343	575	758	939
	179	355	577	762	945
	182	357	582	764	946
	184	366	583	769	952
	187	368	585	772	971
	191	371	591	779	973
	192	374	592	781	982
	193	375	600	783	984
	214	386	604	785	985
	222	391	608	793	988
	225	394	612	799	993
	229	398	614	802	1005
	238	404	631	805	1009
	239	407	633	809	
	242	415	639	819	
	251	419	643	822	

Approach from Right side Algorithm

Originally the Standard AMM Right sided zero-injection consideration algorithm calculated that 187 PMUs are required in total. After this algorithm was applied, 178 additional PMUs are calculated as a requirement for full observability. In total this amounts to 238 PMUs. Results are shown in Table 55.

Table 55: 1009 Bus AMM Outside Input with zero bus injection consideration, approach from right

Number of PMUs	Bus Numbers				
178	12	258	483	716	924
	27	259	486	719	927
	37	262	490	725	929
	51	270	497	728	934
	66	271	498	729	939
	71	273	513	733	943
	73	275	522	736	948
	75	276	527	754	956
	77	277	532	758	959
	80	287	534	761	964
	90	294	539	769	969
	96	297	540	771	971
	103	300	543	775	972
	106	310	560	782	984
	112	315	563	783	992
	115	321	568	797	995
	134	322	573	803	998
	144	328	574	804	1007
	147	360	578	808	
	154	363	588	811	
	168	367	599	812	
	177	373	616	821	
	180	376	627	822	
	184	377	631	823	
	185	378	633	835	
	186	383	648	838	
	197	388	656	850	
	205	399	657	854	
	206	402	658	856	
	207	409	667	867	
	210	412	668	870	
	211	416	671	884	
	215	418	672	885	
	225	419	676	892	
	229	420	680	893	
	233	441	684	895	
	239	452	685	898	
	244	454	696	904	
	252	480	697	912	
	253	482	701	919	

One might wonder what would happen if in phase 2 or 3 bus positions other than the future calculated PMU positions (to be placed in the next phase) are requested, compared to when bus locations from the future positions list are used. The algorithm was designed to continuously calculate the observed and unobserved buses and compensate for this, even for this circumstance. Below is an example of the IEEE 14 bus system using the left sided algorithm to demonstrate the difference between the two scenarios. Firstly an example is shown of the case when a bus is chosen from the list of future PMU positions that was previously calculated and then an example where a bus not on the list is chosen.

Example 1(from 14 bus left sided algorithm):

```
input number of zero-injection buses
1

 Type the number of the bus(press enter after each entry)
7

 Type the number of PMUs installed in the previous phase
2

 Type the bus numbers of previous phase PMU locations (press enter after each entry)
4
3

 Type the number of PMUs to be installed in this phase
1

 Type the PMU positions for installation in this phase, press enter after each entry
5

 The following buses are still unobserved:

E =

     6
    10
    11
    12
    13
    14


PMUs =

     5


PMU_Vector_Positions =

     0     0     1     1     1     0     0     0     0     0     1     0     1     0


 All PMU positions

k =

     3
```

```
k =

     4


k =

     5


k =

    11


k =

    13


 PMUs still required at buses:

k =

    11


k =

    13

Elapsed time is 0.219255 seconds.
```

The results are displayed in Tables 56 and 57. Here it must be noted that one PMU less is now required (from the example of the 14 bus left side simulation) after the phase 2 PMU is installed at a suggested future location.

Table 56: 14 bus AMM Outside input when selected PMU buses are on the list of future PMU location suggestions, phase 1

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
2	4, 3	5	1, 6, 10, 11, 12, 13, 14	5, 11, 13	4	2, 7, 11, 13

Table 57: 14 bus AMM Outside input when selected PMU buses are on the list of future PMU location suggestions, phase 2

Desired amount of PMUs in phase 2	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
1	5	5	6, 10, 11, 12, 13, 14	11, 13	4	2, 7, 11, 13

Example 2 (for 14 bus left sided algorithm):

```
input number of zero-injection buses
1

 Type the number of the bus(press enter after each entry)
7

 Type the number of PMUs installed in the previous phase
2

 Type the bus numbers of previous phase PMU locations (press enter after each entry)
4
3

 Type the number of PMUs to be installed in this phase
1

 Type the PMU positions for installation in this phase, press enter after each entry
6

 The following buses are still unobserved:

E =

     1
    10
    14

PMUs =

     5

PMU_Vector_Positions =

     0     0     1     1     1     1     0     0     1     0     0     0     0     0

 All PMU positions

k =

     3

k =

     4
```

```
k =

     5

k =

     6

k =

     9

 PMUs still required at buses:

k =

     5

k =

     9

Elapsed time is 0.218914 seconds.
```

The results are displayed in Tables 58 and 59. Here once the PMU was placed at bus 6 different buses were made observable and therefore required different PMU positions for full observability; as shown with the future phase's positions to be at buses 5 and 9.

Table 58: 14 bus AMM Outside input when selected PMU buses are not on the list of future PMU location suggestions, phase 1

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
2	4, 3	5	1, 6, 10, 11, 12, 13, 14	5, 11, 13	4	2, 7, 11, 13

Table 59: 14 bus AMM Outside input when selected PMU buses are not on the list of future PMU location suggestions, phase 2

Desired amount of PMUs in phase 2	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
1	6	5	1, 10, 14	5, 9	4	2, 7, 11, 13

4.4 The Simulation of the Advanced Matrix Manipulation Bad data Aid Algorithm

The results in this section all render each bus observable with a minimum redundancy of two PMUs. The logic of this algorithm is similar to the standard AMM algorithm.

4.4.1 IEEE 14 Bus test System Optimal PMU Placement Calculation with Bad Data Detection Aid

Without considering Zero-Injection buses

Approach from Left side algorithm

Table 60 shows that for full observability of the IEEE 14 bus test System with at least 2 PMUs observing each bus, when not considering the zero-injection buses and approaching the vector from the left side, 10 buses require PMUs and the bus positions are:

Buses 2, 4, 5, 6, 7, 8, 9, 10, 11 and 13

Table 60: 14 Bus AMM Bad Data, without zero-injection consideration, approach from left

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	10	2, 4, 5, 6, 7, 8, 9, 10, 11, 13	7

Approach from Right side Algorithm

Table 61 shows that for full observability of the IEEE 14 bus test System with at least 2 PMUs observing each bus, when not considering the zero-injection buses and approaching the vector from the right side, 10 buses require PMUs and the bus positions are:

Buses 1, 2, 3, 6, 7, 8, 9, 11, 12 and 13

Table 61: 14 Bus AMM Bad Data, without zero-injection consideration, approach from right

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	10	1, 2, 3, 6, 7, 8, 9, 11, 12, 13	7

Considering Zero-Injection buses

Approach from Left side algorithm

Table 62 shows that for full observability of the IEEE 14 bus test System with at least 2 PMUs observing each bus, when considering the zero-injection buses and approaching the vector from the left side, 9 buses require PMUs and the bus positions are:

Buses 2, 4, 5, 6, 7, 9, 10, 11 and 13

Table 62: 14 Bus AMM Bad Data, with zero-injection consideration, approach from left

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	9	2, 4, 5, 6, 7, 9, 10, 11, 13	7

Approach from Right side Algorithm

Table 63 shows that for full observability of the IEEE 14 bus test System with at least 2 PMUs observing each bus, when considering the zero-injection buses and approaching the vector from the right side, 9 buses require PMUs and the bus positions are:

Buses 1, 2, 3, 6, 7, 9, 11, 12 and 13

Table 63: 14 Bus AMM Bad Data, with zero-injection consideration, approach from right

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	9	1, 2, 3, 6, 7, 9, 11, 12, 13	7

4.4.2 IEEE 30 Bus test System Optimal PMU Placement Calculation with Bad Data Detection Aid

Without considering Zero-Injection buses

Approach from Left side algorithm

Table 64 shows that for full observability of the IEEE 30 bus test System with at least 2 PMUs observing each bus, when not considering the zero-injection buses and approaching the vector from the left side, 21 buses require PMUs and the bus positions are:

Buses 1, 2, 3, 5, 6, 9, 10, 11, 12, 13, 15, 17, 19, 20, 22, 24, 25, 26, 27, 28 and 30

Table 64: 30 Bus AMM Bad Data, without zero-injection consideration, approach from left

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	21	1, 2, 3, 5, 6, 9, 10, 11, 12,13, 15, 17, 19, 20, 22, 24, 25, 26, 27, 28, 30	6, 9, 22, 25, 27, 28

Approach from Right side Algorithm

Table 65 shows that for full observability of the IEEE 30 bus test System with at least 2 PMUs observing each bus, when not considering the zero-injection buses and approaching the vector from the right side, 21 buses require PMUs and the bus positions are:

Buses 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 18, 20, 21, 24, 25, 26, 27 and 29

Table 65: 30 Bus AMM Bad Data, without zero-injection consideration, approach from right

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	21	1, 2, 3, 5, 6, 8, 9, 10, 11, 12,13, 15, 16, 18, 20, 21, 24, 25, 26, 27, 29	6, 9, 22, 25, 27, 28

Considering Zero-Injection buses

Approach from Left side algorithm

Table 66 shows that for full observability of the IEEE 30 bus test System with at least 2 PMUs observing each bus, when considering the zero-injection buses and approaching the vector from the left side, 16 buses require PMUs and the bus positions are:

Buses 1, 2, 3, 5, 10, 11, 12, 13, 15, 17, 19, 20, 24, 25, 27 and 28

Table 66: 30 Bus AMM Bad Data, with zero-injection consideration, approach from left

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	16	1, 2, 3, 5, 10, 11, 12, 13, 15, 17, 19, 20, 24, 25, 27, 28	6, 9, 22, 25, 27, 28

Approach from Right side Algorithm

Table 67 shows that for full observability of the IEEE 30 bus test System with at least 2 PMUs observing each bus, when considering the zero-injection buses and approaching the vector from the right side, 16 buses require PMUs and the bus positions are:

Buses 1, 2, 3, 5, 6, 9, 10, 12, 13, 15, 16, 18, 20, 24, 25 and 27

Table 67: 30 Bus AMM Bad Data, with zero-injection consideration, approach from right

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	16	1, 2, 3, 5, 6, 9, 10, 12, 13, 15, 16, 18, 20, 24, 25, 27	6, 9, 22, 25, 27, 28

4.4.3 IEEE 57 Bus test System Optimal PMU Placement Calculation with Bad Data Detection Aid

Without considering Zero-Injection buses

Approach from Left side algorithm

Table 68 shows that for full observability of the IEEE 57 bus test System with at least 2 PMUs observing each bus, when not considering the zero-injection buses and approaching the vector from the left side, 34 buses require PMUs and the bus positions are:

Buses 1, 3, 4, 6, 9, 10, 12, 14, 15, 19, 20, 22, 23, 25, 26, 27, 29, 30, 32, 34, 36, 37, 38, 39, 41, 43, 45, 47, 49, 51, 52, 54, 55 and 56

Table 68: 57 Bus AMM Bad Data, without zero-injection consideration, approach from left

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	34	1, 3, 4, 6, 9, 10, 12, 14, 15, 19, 20, 22, 23, 25, 26, 27, 29, 30, 32, 34, 36, 37, 38, 39, 41, 43, 45, 47, 49, 51, 52, 54, 55, 56	4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48

Approach from Right side Algorithm

Table 69 shows that for full observability of the IEEE 57 bus test System with at least 2 PMUs observing each bus, when not considering the zero-injection buses and approaching the vector from the right side, 35 buses require PMUs and the bus positions are:

Buses 1, 2, 4, 6, 9, 10, 11,12, 14, 15, 18, 19, 21, 22, 24, 25, 26, 27, 29, 30, 32, 34, 36, 37, 38, 39, 41, 45, 46, 49, 51, 52, 53, 55 and 56

Table 69: 57 Bus AMM Bad Data, without zero-injection consideration, approach from right

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	35	1, 2, 4, 6, 9, 10, 11,12, 14, 15, 18, 19, 21, 22, 24, 25, 26, 27, 29, 30, 32, 34, 36, 37, 38, 39, 41, 45, 46, 49, 51, 52, 53, 55, 56	4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48

Considering Zero-Injection buses

Approach from Left side algorithm

Table 70 shows that for full observability of the IEEE 57 bus test System with at least 2 PMUs observing each bus, when considering the zero-injection buses and approaching the vector from the left side, 29 buses require PMUs and the bus positions are:

Buses 1, 3, 6, 9, 10, 12, 14, 15, 18, 19, 21, 24, 28, 29, 30, 31, 32, 34, 36, 41, 45, 47, 49, 51, 52, 54, 55, 56 and 57.

Table 70: 57 Bus AMM Bad Data, with zero-injection consideration, approach from left

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	29	1, 3, 6, 9, 10, 12, 14, 15, 18, 19, 21, 24, 28, 29, 30, 31, 32, 34, 36, 41, 45, 47, 49, 51, 52, 54, 55, 56, 57	4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48

Approach from Right side Algorithm

Table 71 shows that for full observability of the IEEE 57 bus test System with at least 2 PMUs observing each bus, when considering the zero-injection buses and approaching the vector from the right side, 30 buses require PMUs and the bus positions are:

Buses 1, 2, 6, 9, 10, 12, 14, 15, 18, 19, 21, 24, 25, 27, 29, 30, 32, 34, 36, 37, 38, 41, 44, 46, 49, 51, 52, 53, 55 and 56.

Table 71: 57 Bus AMM Bad Data, with zero-injection consideration, approach from right

Required Redundancy Level	Minimum PMUs	Bus positions for Minimum number of PMUs	Zero-injection Bus positions
2	30	1, 2, 6, 9, 10, 12, 14, 15, 18, 19, 21, 24, 25, 27, 29, 30, 32, 34, 36, 37, 38, 41, 44, 46, 49, 51, 52, 53, 55, 56	4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48

4.4.4 1009 bus typical South African Utility System Optimal PMU Placement Calculation with Bad Data Detection Aid

Without considering Zero-Injection buses

Approach from Left side algorithm

The redundancy level chosen was again for a minimum of 2 PMUs observing each bus. The zero-injection buses are the same as originally mentioned at the start of this chapter. When this particular algorithm was run, a total of 740 PMUs were calculated as necessary for full observability of the network. The bus locations are given in below in Table 72.

Table 72: 1009 Bus AMM Bad Data, without zero-injection consideration, approach from left

Total PMUs	Bus Numbers							
740	1	151	275	405	543	685	822	955
	2	152	276	406	546	686	823	956
	3	153	277	407	547	687	824	957
	6	154	279	409	548	688	825	959
	7	155	280	410	549	689	826	960
	8	156	281	412	551	692	828	963
	11	158	282	413	552	693	829	964
	12	159	284	415	553	694	830	965
	13	160	285	416	554	695	831	967
	15	161	286	417	555	696	833	968
	17	162	287	419	556	697	834	969
	18	164	288	420	559	698	835	970
	19	165	290	421	561	699	836	971
	21	166	292	423	562	700	837	973
	22	167	293	424	563	701	839	974

24	168	295	427	564	703	840	975
25	169	296	428	565	704	841	977
26	170	297	429	567	709	842	978
28	171	298	430	568	711	843	981
29	172	299	431	569	712	844	982
30	173	300	432	570	713	845	985
33	174	301	433	571	715	846	986
34	175	302	434	573	716	847	989
36	176	303	435	575	717	849	990
37	177	304	436	576	718	850	991
38	179	308	437	577	719	851	992
41	180	309	438	578	720	852	993
42	181	310	439	580	721	853	994
43	182	311	441	581	722	854	995
45	183	313	444	585	723	855	997
46	184	314	445	586	724	856	998
47	185	315	447	587	725	857	999
48	186	316	448	588	726	859	1000
49	187	318	449	589	727	861	1001
51	189	319	450	591	728	864	1002
52	190	320	451	592	729	865	1003
53	191	322	452	593	730	866	1006
54	192	327	453	594	731	867	1007
55	193	330	455	597	732	868	1008
56	195	332	456	598	733	870	1009
57	197	334	458	601	734	871	
58	200	335	459	602	735	872	
59	201	336	460	604	736	873	
63	202	337	461	606	737	874	
64	203	338	464	607	739	875	
65	204	339	466	609	740	876	
68	205	340	467	610	741	878	
69	206	341	468	612	743	879	
70	207	342	469	613	744	880	
71	208	343	472	614	745	881	
72	209	344	474	615	746	883	
73	211	345	475	617	747	884	
74	212	346	477	618	748	885	
77	214	347	479	620	751	886	
78	215	348	480	621	754	889	
79	217	349	482	624	757	890	
80	218	350	483	625	759	891	
81	221	351	484	626	760	892	
82	222	352	486	627	763	896	
83	223	353	487	628	764	897	
85	224	354	488	629	765	900	
86	225	355	489	631	766	902	

	88	226	356	492	632	767	904	
	91	227	357	493	634	768	905	
	95	228	358	495	635	770	908	
	96	229	362	497	636	771	909	
	99	230	363	499	638	773	911	
	100	231	364	500	639	775	912	
	101	233	365	501	640	776	913	
	102	234	368	502	641	777	914	
	103	235	369	503	642	779	915	
	104	237	370	504	644	783	917	
	105	238	371	505	646	784	918	
	109	239	372	506	648	786	919	
	110	241	373	510	653	787	920	
	111	242	374	511	654	788	922	
	113	244	375	512	655	789	923	
	116	245	376	513	656	790	924	
	117	246	377	514	658	791	925	
	121	247	378	517	659	792	926	
	123	249	379	519	660	793	927	
	124	250	380	520	661	794	929	
	125	252	381	521	663	796	930	
	127	254	382	522	664	798	931	
	128	255	383	525	665	799	932	
	129	256	384	526	666	801	933	
	130	257	385	527	667	803	934	
	132	258	386	528	668	805	935	
	134	260	389	529	669	806	936	
	136	262	391	530	671	807	939	
	137	263	392	531	672	808	940	
	141	264	393	532	673	809	941	
	142	265	395	533	675	810	943	
	144	266	396	534	676	811	944	
	145	267	397	537	677	812	947	
	146	268	399	538	678	814	948	
	147	269	400	539	679	815	950	
	148	271	401	540	680	817	952	
	149	273	402	541	682	819	953	
	150	274	404	542	683	820	954	

Approach from Right side Algorithm

When this algorithm is applied, it calculated that 728 PMUs are required for full observability of the network. The bus locations are given below in Table 73:

Table 73: 1009 Bus AMM Bad Data, without zero-injection consideration, approach from right

Number of PMUs	Bus Numbers						
728	2	128	267	413	557	698	834	977
	3	129	268	414	558	700	836	979
	4	131	272	415	559	701	838	980
	5	132	274	416	560	702	839	982
	6	133	275	417	561	703	842	983
	7	134	277	418	562	704	843	984
	11	135	279	419	563	705	844	985
	12	136	280	421	564	706	845	986
	13	137	281	422	565	708	846	988
	14	138	282	423	566	709	847	989
	16	140	284	424	567	711	848	990
	18	141	285	425	568	713	851	991
	19	144	286	427	569	714	852	992
	20	145	287	428	570	715	854	993
	23	149	291	429	571	717	857	994
	24	150	292	430	573	718	858	995
	25	152	293	431	574	719	859	996
	26	153	294	432	575	720	860	998
	27	154	296	433	577	721	861	999
	28	155	297	434	578	722	862	1001
	30	156	298	436	579	723	863	1002
	31	157	299	439	580	724	864	1003
	32	158	300	441	581	725	866	1004
	33	160	301	442	583	726	867	1005
	34	162	302	443	585	727	868	1006
	35	163	303	444	586	728	869	1007
	36	165	304	445	587	729	870	1008
	40	167	305	449	589	733	871	1009
	41	168	306	450	591	734	872	
	42	169	307	454	592	737	873	
	43	170	308	456	594	738	874	
	44	172	309	458	596	739	875	
	45	173	310	459	597	740	876	
	46	174	311	460	599	742	877	
	47	175	312	461	600	743	878	
	50	176	313	463	601	744	880	
	51	177	314	464	603	745	881	
	53	178	315	465	605	747	883	
	54	180	317	468	606	748	884	
	55	181	318	470	607	749	885	
	56	182	319	471	608	750	887	
	57	183	320	473	610	751	890	

58	185	321	474	612	752	891	
59	186	323	475	613	753	892	
60	187	326	476	617	754	893	
62	188	329	477	618	755	894	
63	189	335	478	619	756	895	
64	191	336	479	620	757	896	
65	194	337	481	621	758	897	
67	195	338	482	622	759	899	
69	196	339	484	623	762	900	
71	197	341	488	624	765	901	
72	198	342	489	625	766	904	
74	199	344	490	626	768	906	
75	200	345	493	627	769	907	
76	201	349	494	628	770	908	
77	203	351	496	629	771	911	
78	204	352	497	630	772	912	
79	205	353	499	632	773	913	
80	207	354	500	633	775	914	
81	214	355	501	634	777	915	
82	215	357	502	635	778	917	
83	218	358	504	636	779	918	
84	219	359	505	637	781	919	
85	220	361	506	638	782	920	
87	222	363	507	639	784	921	
88	223	364	508	640	785	922	
89	225	365	509	642	786	923	
90	226	366	512	644	787	924	
91	229	367	513	645	788	926	
92	230	368	514	646	789	927	
93	231	369	516	649	790	929	
95	232	371	520	651	791	931	
96	233	372	521	653	795	932	
97	235	374	522	657	796	934	
98	236	375	523	659	799	935	
102	237	376	525	663	800	936	
103	239	378	527	664	802	937	
104	240	380	528	665	803	942	
106	241	381	529	666	804	943	
107	242	382	530	668	805	944	
108	243	384	532	669	806	945	
109	244	385	534	671	807	946	
110	245	387	535	672	812	948	
111	246	388	536	674	813	950	
112	247	389	537	675	814	951	
113	249	390	538	676	815	952	
114	251	392	539	677	818	953	
115	252	393	540	679	819	955	

	116	253	394	542	680	820	956	
	117	255	396	543	681	821	957	
	118	256	398	544	682	822	960	
	119	257	400	545	685	823	962	
	120	258	401	546	686	824	963	
	121	261	403	549	689	827	969	
	122	262	406	550	690	828	970	
	123	263	408	551	691	830	971	
	124	264	409	553	694	831	972	
	125	265	410	555	695	832	974	
	127	266	411	556	696	833	975	

Considering Zero-Injection buses

Approach from Left side algorithm

Now when zero-injection buses are considered, the Left sided algorithm calculated that 523 PMUs were required for full observability when at least 2 PMUs are required to observe each bus. The bus numbers are given below in Table 74:

Table 74: 1009 Bus AMM Bad Data, with zero-injection consideration, approach from left

Number of PMUs	Bus Numbers					
523	3	184	347	527	743	968
	7	185	348	528	744	969
	8	186	357	529	745	970
	11	187	362	530	746	971
	12	189	363	531	748	973
	15	190	364	532	754	974
	17	191	365	533	757	975
	18	195	368	537	759	977
	19	197	369	538	760	978
	21	200	370	539	763	981
	22	201	371	540	764	985
	24	202	372	541	768	986
	25	203	373	542	771	989
	26	204	374	543	775	990
	28	205	375	546	776	991
	29	207	376	547	779	992
	30	208	377	548	783	993
	37	209	379	549	784	995
	41	211	380	551	786	997
	42	212	381	552	791	1001

43	214	384	553	803	1002
45	215	385	554	807	1003
46	217	386	555	808	1009
47	218	389	564	814	
48	221	391	565	817	
49	222	392	567	819	
51	224	393	568	820	
52	225	397	569	822	
53	226	399	570	823	
54	227	400	571	824	
55	228	405	573	825	
57	229	410	575	826	
58	230	413	576	828	
59	231	415	577	829	
63	233	416	578	830	
64	234	419	580	831	
65	235	420	581	833	
69	237	424	585	834	
71	238	431	587	835	
74	239	432	589	836	
77	241	433	591	837	
78	242	434	592	844	
79	244	436	597	847	
80	245	437	601	849	
81	246	438	602	851	
82	247	439	609	852	
83	249	444	610	853	
86	250	445	612	854	
91	252	447	613	855	
95	254	448	615	856	
96	255	449	617	857	
99	256	450	618	859	
100	257	451	620	861	
101	258	452	621	864	
102	262	453	624	865	
103	264	455	625	866	
104	265	456	626	867	
109	266	458	627	875	
110	271	459	628	876	
111	273	460	629	878	
113	274	461	631	879	
116	277	464	634	880	
117	281	466	635	881	
121	284	467	636	886	
123	287	468	638	889	
124	288	469	639	890	
125	292	472	648	891	

	127	293	474	653	892	
	128	295	475	654	900	
	129	296	477	655	902	
	130	297	479	656	904	
	132	298	480	658	911	
	134	299	482	659	912	
	136	300	483	660	913	
	137	301	484	661	914	
	141	302	486	663	922	
	142	303	487	668	926	
	144	304	488	673	927	
	145	308	489	676	929	
	156	309	492	677	930	
	158	310	499	678	931	
	159	311	500	679	932	
	161	313	501	680	933	
	162	314	502	682	935	
	164	315	503	687	936	
	166	316	504	711	939	
	167	318	505	716	940	
	169	319	506	718	941	
	170	320	510	719	943	
	171	330	511	720	944	
	172	336	512	721	947	
	173	337	513	723	948	
	174	339	514	724	950	
	175	340	517	729	952	
	177	341	519	730	953	
	179	342	520	732	954	
	180	343	521	733	955	
	181	344	522	734	957	
	182	345	525	735	965	
	183	346	526	736	967	

Approach from Right side Algorithm

When zero-injections are considered, the Right sided algorithm calculated that 508 PMUs are required for full observability when at least 2 PMUs observe each bus. The bus numbers are given below in Table 75:

Table 75: 1009 Bus AMM Bad Data, with zero-injection consideration, approach from right

Number of PMUs	Bus Numbers					
508	3	177	365	550	779	993
	7	178	366	551	781	995
	11	180	367	553	782	1001

12	181	368	555	784	1002
14	182	369	559	785	1003
16	183	371	562	786	1004
18	185	372	564	787	1005
19	186	374	565	788	1009
20	187	375	566	790	
23	188	376	567	803	
24	189	378	568	807	
25	191	385	569	813	
26	194	387	570	814	
27	197	388	571	818	
28	200	389	573	819	
30	201	390	574	820	
31	204	392	575	821	
32	205	393	577	822	
40	207	394	578	823	
41	214	398	579	824	
43	215	400	580	827	
44	218	413	581	828	
45	219	414	583	830	
46	222	415	591	831	
47	225	418	592	832	
50	226	419	596	833	
51	229	422	597	834	
53	230	423	601	836	
54	231	431	610	838	
55	232	432	612	839	
57	233	433	613	844	
58	235	434	617	846	
59	236	436	618	847	
60	237	439	619	848	
62	239	442	620	852	
63	240	443	621	854	
64	241	444	622	857	
65	242	445	623	859	
74	243	449	624	860	
75	244	450	625	861	
77	245	454	626	864	
78	246	456	627	866	
79	247	458	628	867	
80	249	459	629	875	
82	251	460	630	876	
83	252	461	633	877	
91	253	463	634	878	
92	255	464	635	880	
93	256	465	636	881	
95	262	468	638	887	

96	263	470	639	890	
97	265	471	649	891	
102	266	473	651	892	
103	272	474	653	893	
104	274	475	657	899	
106	284	476	659	900	
107	285	477	663	901	
108	287	478	668	904	
109	291	479	669	906	
110	292	481	676	911	
111	293	482	677	912	
112	296	484	679	913	
113	297	488	680	914	
114	298	489	681	926	
115	299	490	682	927	
116	300	494	702	929	
117	301	497	704	931	
118	302	500	705	932	
121	303	502	708	935	
122	304	504	711	936	
123	305	505	714	937	
124	306	506	715	942	
125	307	507	718	943	
127	308	508	719	944	
128	309	512	720	945	
129	310	513	723	946	
131	311	514	724	948	
132	312	516	729	950	
133	313	520	733	951	
134	314	521	734	952	
135	315	522	742	953	
136	317	523	743	955	
137	318	525	744	957	
138	319	527	745	962	
140	320	528	748	969	
141	321	529	749	970	
144	323	530	752	971	
145	326	532	753	972	
150	329	535	754	974	
157	339	536	755	975	
158	341	537	756	977	
162	342	538	757	983	
163	344	539	758	984	
167	345	540	759	985	
170	349	542	768	986	
172	354	543	769	988	
173	357	544	771	989	

	174	361	545	772	990	
	175	363	546	775	991	
	176	364	549	778	992	

4.5 Intentional Islanding Detection via PMUs

The issue of unintentional Islanding detection will not be addressed, because as mentioned in Chapter 3, this can be compensated for by means of the Standard AMM Algorithm since PMUs placed anywhere will detect if the buses to which they are connected have become live.

The *Intentional Islanding detection Algorithm* will be demonstrated on the IEEE 30 bus system by testing each of the three scenarios mentioned in Chapter3. These scenarios being:

- Apply the Standard AMM algorithm to locate Optimal PMU positions
- Apply the Standard AMM algorithm and then improve the solution by additionally applying any of the developed algorithms to the respective islands
- First form the Islands and then apply the Standard AMM algorithm on each island to make the buses observable

The Islanding detection algorithms are not specificly coded algorithms. The Islanding detection problem is solved by using the *Standard AMM* and *Outside Input AMM* algorithms. Only the versions of these two algorithms which consider zero-injection buses will be used.

4.5.1 Case1: Island design after PMU placement

This method uses the *Standard AMM* algorithm to calculate the PMU positions. The Islands are segmented only after the entire Network is completely observable. Therefore the results from the Standard AMM algorithm for the IEEE 30 bus system with left sided and right sided approaches can be quoted. This is shown in Tables 76 and 77 below:

The Left Sided Algorithm calculated:

Table 76: 30 Bus AMM Islanding, start from left, Case 1

Required PMUs	Buses with PMUs	Zero-Injection Buses	Computation time
6	3, 5, 10, 12, 15, 20	6, 9, 22, 25, 27, 28	0.121228 seconds

The Right sided Algorithm calculated:

Table 77: 30 Bus AMM Islanding, start from right, Case 1

Required PMUs	Buses with PMUs	Zero-Injection Buses	Computation time
6	1, 2, 10, 12, 15, 20	6, 9, 22, 25, 27, 28	0.121178 seconds

The islands are formed after the PMUs have been placed. The locations and amount of buses included in each island are constrained by the PMUs observing them. An island cannot share a PMU with another island, therefore all the buses in each island must be observed by PMUs within that island.

4.5.2 Case2: Island design after PMU placement, with Island minimisation

Initially the same is done here as done in Case1: perform a Standard AMM calculation. The islands can be defined in any size that the user desires. For the IEEE 30 bus system, 3 Islands are established and are shown in figure 22. The Islands are distinguished by the colours red, blue and green.

Island 1 (Red) consists of the following 10 buses:
1, 2, 3, 4, 5, 6, 7, 8, 9 and 11

Island 2 (Blue) consists of the following 9 buses:
10, 12, 13, 14, 16, 17, 20, 21 and 28

Island 3 (Green) consists of the remaining 11 buses:
15, 18, 19, 22, 23, 24, 25, 26, 27, 29 and 30

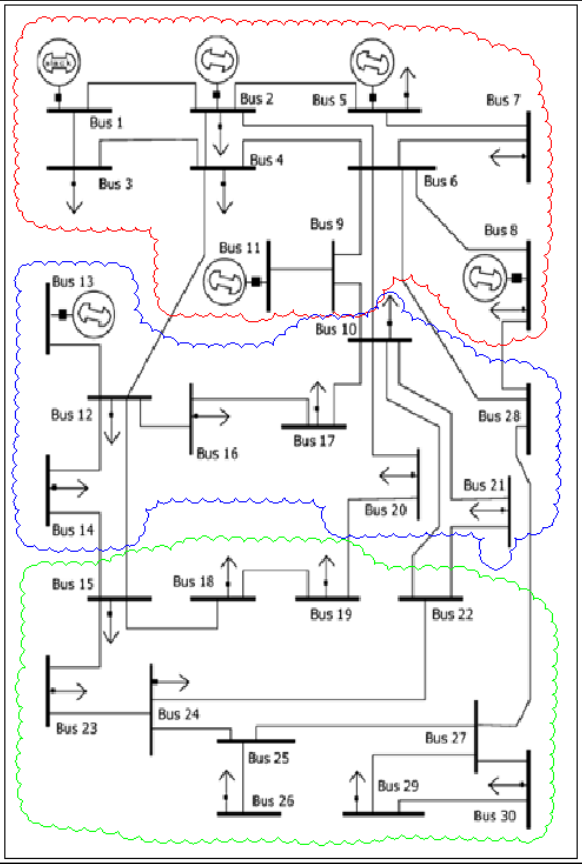

Figure 22 Three Islands chosen for Case 2 of Intentional Islanding detection

Now it must be checked that each Island is fully observable and independent of PMUs of other islands. The right sided solution above will be used from here on. The PMUs are at buses 1, 2, 10, 12, 15 and 20.

Island 1 is fully observable by PMUs at buses 1 and 2.

Island 2 has all its buses observed by PMUs at buses 10, 12 and 20, except for bus 28. Therefore a PMU is required at bus 28.

Island 3 only has one PMU at bus 15. This situation is not as simple as in Island 2 and will require the *Outside Input AMM* calculation to be applied. The new connectivity Matrix will comprise of all the buses in Island 3.

The result was that 4 PMUs are necessary in this island. The bus locations of these PMUs are at buses 15, 18, 22 and 25.

Now all of the islands are observable independently of one another.

4.5.3 Case3: Island design before PMU placement

In this case, the Islands will be segmented before the PMUs are placed. It is meant that maximum flexibility be given to the Island designers. A proposed system of 3 Islands is chosen where each Island consists of:
Island 1 (Red) consists of 11 buses:
5, 6, 7, 8, 9, 10, 11, 17, 20, 21 and 28

Island 2 (Blue) consists of 8 buses:
1, 2, 3, 4, 12, 13, 14 and 16

Island 3 (Green) consists of the remaining 11 buses:
15, 18, 19, 22, 23, 24, 25, 26, 27, 29 and 30

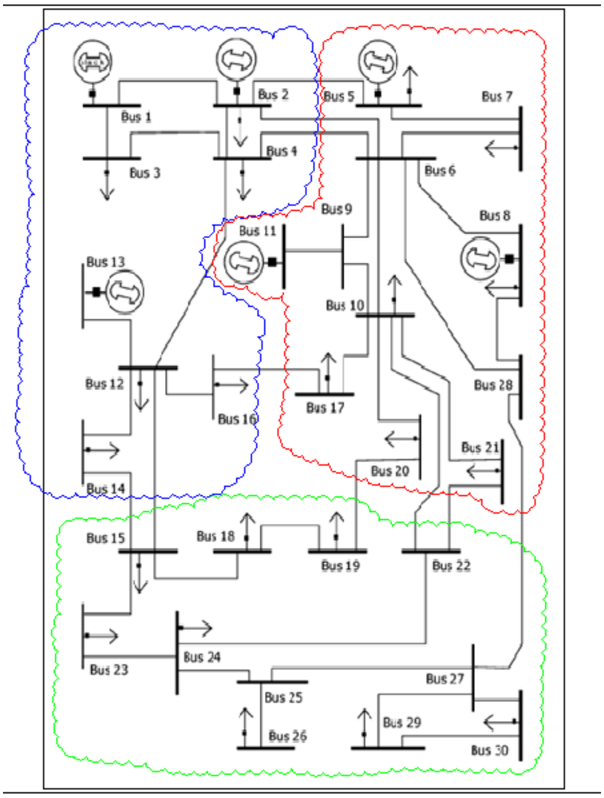

Figure 23 Three Islands chosen for Case 3 of Intentional Islanding

The PMUs will be placed according to what islands they fall into. A separate Connectivity Matrix is formed for each Island and is simulated by the *Standard AMM* algorithm.

Island 1 was simulated and the required PMUs for full observability were 2. These PMUs are to be placed at buses 7 and 10.

Island 2 was simulated and the required PMUs for full observability were 2. These PMUs are to be placed at buses 1 and 12.

Island 3 was simulated and a total of 4 PMUs were calculated as necessary for full observability. These PMUs are to be placed at buses 15, 18, 22 and 25.

All of the Islands are now independently observed by their own set of PMUs. The total amount of PMUs for the entire system is shown to be 8 instead of the 6 calculated by the *Standard MM* algorithm.

4.6 Conclusion

The results captured in this chapter are solutions obtained when running the respective algorithms on each respective bus network in Matlab. The detailed solutions are presented for viewing in the attached Results Appendix document. The algorithms have each been run multiple times and the same results were obtained each time. The results were then manually checked against the various connectivity matrices of the bus networks and proved to meet the respective objectives each and every time. This validates the correctness of the solutions. The results presented in Chapter 4 are discussed in Chapter 5.

5. Discussion of results

The discussions in this chapter will cover the main observations of the simulation results presented in Chapter 4. The key results will be addressed and the order of the algorithms follows the order in the results chapter.

For all the test systems, the results for optimal PMU placement obtained by Standard Advanced Matrix Manipulation algorithm are compared to those obtained using ILP algorithm. The ILP algorithm was computed on Matlab by using the *Tomlab®* toolbox and uses the CPLEX function for Binary Integer Linear Programming. ILP was used as a comparison as it is recognised by many experts as one of the best optimisation algorithms for placement of PMUs.

5.1 Standard Advanced Matrix Manipulation

This algorithm consists of Left sided and right sided solutions. This refers to the manner in which the final solution is approached in the main part of the algorithm. Therefore the results that these two algorithms produce are not the same. This is because the ANDing function that Matlab offers was not usable in the exact manner that was required. This then required the development of a separate customized ANDing function. This developed function simulates what the AND function does, in steps where instead of ANDing the entire row at a time with another row, the AMM Algorithm ANDs one row element at a time with the corresponding column element of the following row. The elements are either processed from the right or left side of the solution vector.

When zero-injection buses are not considered, the AMM algorithm and ILP provide the same results for IEEE 14 and 30 bus systems with 4 and 10 PMUs required, respectively as shown in Table 78 below. The IEEE 57 bus system and the 1009 bus Typical South African utility system results however display variances between ILP and AMM. This is because as the systems become larger and more complex, the true nature of the applied algorithm is revealed. This refers to different optimisation logic followed by each algorithm, as discussed throughout the thesis. Despite the fact that they both aim at minimising the total amount of PMUs, their methods of approach are different. This causes the difference in results in terms of number of PMUs placed. No two algorithms will always give the same result. It is shown especially with the 1009 bus typical South African utility network that the ILP produces a much better result than the AMM Left or Right side approach solution when zero-injection buses are not considered. This says that before zero-injection consideration, ILP becomes progressively better than AMM as the network becomes larger.

When zero-injection buses are considered, the IEEE 14 bus system Right side approach solution is the only result that exactly matches ILP's solution where both algorithms calculated that 3 PMUs are required, while the Left side solution gave 4 PMUs required. The AMM Left and Right side approaches both calculated better results for the IEEE 30 bus system than the ILP where the latter calculated 7 PMUs required and both AMM approaches calculated 6 PMUs. However for the IEEE 57 bus system, the AMM Left and Right side approaches are inferior to the ILP result as shown in Table 78. The 1009 bus utility system solution is much better than ILP's optimal solution, where the standard AMM Right and Left side solutions calculated 187 and 203 PMUs necessary, respectively; while ILP calculated a minimum of 226 PMUs. This result was confirmed after running the algorithm multiple times over as well as manually checking the solution against the connectivity matrix of the 1009 bus system.

The 1009 bus system has 347 zero-injection buses, which equates to over a third of the entire network. The AMM's zero-injection bus solver is very unique since it allows zero-injection buses to observe other zero-injection buses. This advantage is only realised in a system where the topology consists of zero-injection buses that are connected to other zero-injection buses. The 1009 bus system is such a system, as are most large networks. Therefore it is expected that the AMM algorithm will display its full ability when faced with such a system, as is demonstrated in the relevant result.

Table 78: Summary and comparison of results derived from AMM and ILP simulations

Bus Network	Without considering Zero-Injection buses			With considering Zero-Injection buses		
	ILP	AMM		ILP	AMM	
		Left	Right		Left	Right
IEEE 14 bus	4	4	4	3	4	3
IEEE 30bus	10	10	10	7	6	6
IEEE 57 bus	17	19	20	12	16	15
1009 bus Typical South African Network	282	320	313	226	203	187

5.2 Advanced Matrix Manipulation for Inside Input

This algorithm was designed by using the Standard Advanced Matrix Manipulation's engine and adding the ability to phase installations of PMUs by using the same list of PMUs determined by the standard AMM algorithm and then scheduling which PMUs are to be installed before others in the respective phases. The results displayed are a reflection of which buses in the system have higher observability capabilities than others. Depending on this, whatever amounts of PMUs are required in each phase, the algorithm will suggest which bus locations would be ideal for maximising total system observability.

If however the client requests an amount of PMUs in the first phase more than that determined by the standard algorithm, then the algorithm will place the buses at the standard algorithm's solution locations. This is because it is sticking to the purpose of the algorithm, which is not to place more PMUs than what is necessary.

If the client indeed has any special requirements which must be adhered to, that entails specific buses to receive PMUs, then that is addressed by the next algorithm discussed in Section 5.3.

In the simulations of the test systems, the same inputs were fed into the left and right sided algorithms in order to note the different results that may be obtained. It is unpredictable when the right or the left algorithm will provide a better result, so it was decided that both are always used on the same system. The number of PMUs to be installed per phase was to be as equally spread as possible per phase (for example, an average of 3 PMUs be installed per phase) in order to display the different observability levels of the PMUs at their respective buses. The total locations for PMU placements are exactly the same as for the standard algorithm for the respective left side and right sided algorithms, for each specific bus system. This algorithm therefore makes apparent which locations observe more buses than others. It must be noted that this algorithm runs by normally taking consideration of Zero-injection buses. The consideration of as well as non-consideration of zero-injection buses would not have affected the ordering of the PMUs. Therefore it was seen as unnecessary to have both and it was decided to use the algorithm that considers zero-injection buses since then there are fewer PMUs to work with.

Table 79 shows the result of the IEEE 30 Bus System simulated by the AMM Inside Input algorithm. The Left and Right side approaches in the Standard AMM algorithm both calculated 6 PMUs necessary. These 6 PMUs, although at different buses for the Left and Right side approach, was divided into 3 phases with 2 PMUs installed in each phase. These number of phases and number of PMUs per phase were selected by the user. Table 79 shows which buses are to be receive a PMU in each phase.

Table 79: IEEE 30 Bus system Inside Input PMU bus number scheduling

Test System	Approach from side	Bus locations for PMUs in Phase1	Bus locations for PMUs in Phase 2	Bus locations for PMUs in Phase3
IEEE 30 Bus system	Left	10, 12	3, 15	5, 20
	Right	2, 10	12, 15	1, 20

The observability order of the buses can be noted by looking at the buses which are similar to both solutions, ie. Bus 10, 12, 15 and 20. In both phasing solutions, they appear in the same order with respect to one another. This shows that the buses ordered 10:12:15:20 are from highest level of observability to lowest.

5.3 Advanced Matrix Manipulation for Outside Input

The main objective when designing the AMM Outside Input algorithm was to make it as flexible to user requirements as and when necessary. This algorithm allows for phasing installation of PMUs such as in *AMM for Inside Input,* but now allows the user to specify which buses would require a PMU placement. The algorithm will calculate and make a suggestion of where the remaining PMUs should be placed to fulfil the amount of PMUs desired to be installed in the next phase, as well as the rest of the PMU locations for full observability of the network. It uses what can be referred to as a "rolling minimum" where the full optimal solution (all the PMUs for full observability) is continuously updated for each phase, depending on the user input for the required bus locations. This algorithm only runs when considering zero-injection buses, for the same reason as with the Inside Input algorithm.

As all the results are presented in similar form, therefore by random choice the IEEE 30 bus simulation will be discussed. For the Left and Right side approaches of the standard AMM algorithm, both solutions gave a total minimum of 6 PMUs required for full observability. The same user input buses are used for both the left and right side approaches. The user chose 4 buses to receive a PMU where 2 of these buses corresponded to the standard AMM Left side solution and only one of the user-required buses corresponding to the standard AMM Right side solution. The Outside input algorithm then calculated that after these user-required buses were incorporated, a total of 8 PMUs is required for full observability in both the Right and Left sided approaches. It is interesting to note that in Table 80 (quoted from Table 49), the Left side simulation calculated that the remaining four PMUs (of the total 8) are the 4 PMUs which were not used from the standard AMM simulation for the Left side approach by the users requirement. This means that the 2 PMUs which the user requested, which was not part of the standard AMM solution list, did not contribute at all to the effective observability of the remaining unobserved buses. If they had, then the remaining 4 PMUs would not have been exactly the same as the remaining PMUs from the standard AMM solution, as occurs with the Right side approach in Table 81 (quoted from Table 50).

Table 80: 30 Bus AMM Outside Input with zero bus injection consideration, approach from left

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
4	3, 12, 17, 25	8	2, 5, 6, 7, 8, 9, 11, 18, 19, 20, 21, 22, 23, 28, 29, 30	5, 10, 15, 20	6	3, 5, 10, 12, 15, 20

Table 81: 30 Bus AMM Outside Input with zero bus injection consideration, approach from right

Desired amount of PMUs in phase 1	Bus positions of Desired PMUs	New total amount of PMUs for full observability	Currently Unobserved buses	Future PMUs to be placed at buses	Total amount of PMUs through standard Matrix Manipulation	Bus positions of standard PMUs
4	3, 12, 17, 25	8	2, 5, 6, 7, 8, 9, 11, 18, 19, 20, 21, 22, 23, 28, 29, 30	2, 10, 15, 18	6	1, 2, 10, 12, 15, 20

In the results of the IEEE 14, 30, 57 bus systems as well as the 1009 bus system, it is shown that the total number of PMUs required after the random user input was given, is more than the total number of PMUs determined by the standard AMM algorithm. This demonstrates that when user inputs are given outside the optimal group of locations, the total amount of required PMUs tends to increase. When the algorithm was run, only the first phase of installations after the user required PMUs were installed is shown. This is all that is required to show how the algorithm works. For the second phase the only difference would be that when inserting which PMUs are user requirements, insert all the previously installed PMUs (including the original user required PMUs). The result optimizes the remaining PMU locations.

It is then demonstrated in an example (at the end of Section 4.3.4) of the IEEE 14 bus system that the *Outside Input algorithm* can act as an *Inside Input algorithm* if the bus locations which are suggested, are in fact used in the following phase of installation. The example shows that the required list of PMUs gets smaller should a suggested location be used in the next phase (See Tables 56 and 57). When a location is used that is not present in the list, the rolling minimum is updated but is still equal to a minimum of 5 PMUs necessary for full observability. The buses which are observed will be different to that when a suggested PMU location was used (When Tables 56 and 57 are compared to Tables 58 and 59, respectively), due to the fact that different buses are connected to different buses.

Each phase can be carried out as shown in the example in Section 4.3.4. This algorithm's solutions depend on the user input for the amount of phases, PMUs per phase and which buses require PMUs. Therefore the final placements for PMUs may be very different to what was calculated by the standard AMM algorithm.

5.4 Advanced Matrix Manipulation for Bad Data Aid

The Bad Data Aid algorithm serves to increase the redundancy levels of observability collectively for all the buses – or plainly all the buses will have the same minimum number of PMUs observing them. It requires the user to insert the number of PMUs necessary to observe each bus. The operational logic follows that it will firstly do a standard Advanced Matrix Manipulation calculation to obtain a placement solution. This solution is then checked if each row meets the required redundancy levels. For the rows that do not meet the required redundancy level, PMU placement is done through the standard AMM calculation again. This second round of the standard calculation will set a minimum redundancy level of 2 PMUs per bus. It follows that each round of the standard calculation can increase the redundancy level by one, if applied.

After each round of the standard calculation, the same redundancy checks are carried out on each of the rows that did not meet the user redundancy requirement in the previous stage of the standard calculation. Then these buses are used in the proceeding round to be observed by additional PMUs (which is determined by the calculation in each round).

From all of the results (displayed in Table 82 below) it is observed that there is a clear difference in the amount of required PMUs between the respective calculations for zero-injection bus consideration and for no zero-injection bus consideration. This confirms that even in the redundancy of measurements requirement, the zero-injection buses still serve to minimise the total amount of PMUs. The most noticeable difference between the results of the Standard AMM verses the Bad Data Detection Aid algorithms in Tables 78 and 82, respectively, is the much higher amounts of PMUs required for the Bad data Detection Aid when only one level higher observability redundancy is required (not forgetting that the Standard AMM algorithm uses a default minimum observability redundancy of at least one PMU observing each bus). This algorithm can therefore be used to inform utilities how much more money they will need to spend for the task of detecting bad (false) data by PMUs.

Table 82: Summary of results for Bad Data Detection Aid algorithm for minimum redundancy of 2 PMUs observing each bus

Bus Network	Without considering Zero-Injection buses		With considering Zero-Injection buses	
	AMM		AMM	
	Left	Right	Left	Right
IEEE 14 bus	10	10	9	9
IEEE 30bus	21	21	16	16
IEEE 57 bus	34	35	29	30
1009 bus Typical South African Network	740	728	523	508

The maximum redundancy required used in the simulations was level 2. The reason for this is that it is highly unlikely that level 3 or higher will be used in the present “expensive PMU” situation. The algorithm was therefore limited to compensate for a maximum of two PMUs observing each bus. If 3 PMUs were desired to observe each bus, an extra constraint must be added. This constraint will limit terminal buses to have a maximum of 2 PMUs observing it due to its topological position in the network branch (at the end of the branch, therefore only connected to one other bus).

Branch outages are not considered by this algorithm. This algorithm compensates only for PMU outages. If branch outages were considered in addition to PMU outages, then more PMUs will be required and the programming will be more complex. The reason for this is that some PMUs are set to be observed twice through the same zero-injection bus. If branch outages are considered, then PMUs will only be allowed to be observed once through zero-injection buses, due to the possibility that the zero-injection bus may become defective for whichever reason. This can be done if required, but does not fall within the scope of this thesis.

5.5 Islanding Detection PMU placement algorithm

This algorithm serves to place PMUs so that all of the buses are observed in the event of Islanding. In unintentional Islanding, the buses which will be livened after a DG energises them are unpredictable. Therefore all of the buses need to be observed with equal importance.

It was then formulated that in the case of unintentional Islanding, a *Standard AMM* calculation will be run on the network to place PMUs optimally for full observability. The algorithm will run when taking consideration of zero-injection buses. The IEEE 30 bus system was simulated using the *Standard AMM Right and Left sided algorithm.* The results are that a total of 6 PMUs be placed. The results are the same as that of Tables 13 and 14 in Chapter 4.

Intentional Islanding detection

The intentional Islanding strategy is there to assist restoration to a collapsed network. If these Islands are to operate properly, they will each need to be completely observed independently of one another. PMUs are set to observe these Islands and the method describing how the Advanced Matrix manipulation deals with this is discussed in Chapter 4 section 5.

Case 1

In Case 1 the strategy is to place PMUs in the same manner as for unintentional islanding detection. Therefore the solution is the same as for the *Standard AMM Right and Left sided algorithm* where a minimum of 6 PMUs are necessary for full observability calculated by both approaches. Case 1 is the most basic of the Islanding detection strategies and provides the least flexibility to the Islanding designer.

Case 2

In Case 2 the entire bus network is defined as the first island. According to the solution of the *Standard AMM Right sided algorithm* (in Table 14), the entire network is observable with 6 PMUs, as in case 1. The main island is then divided in three and is shown in figure 22. It is noted that islands 1 and 2 are completely observed by the PMUs within their barriers. However, island 3 requires more PMUs to be made completely observable. A new connectivity matrix was defined for this island and was simulated by the *Outside AMM Right sided algorithm.* The result was that 4 PMUs be placed in that island, inclusive of the one PMU placed by the *Standard AMM Right sided algorithm* which was applied to originally. All of the islands were then acceptably observed independently of each other.

Case 3

For Case 3 it was allowed that the Islanding designer forms the islands before PMUs were placed. The intention was to give full flexibility to the Islanding design process; the monitoring of the Islands by the PMUs is treated as secondary to the positioning of the Islands. Therefore first locate all of the islands, then locate the PMUs.

Three islands were then selected as the start-off islands. Figure 23 shows the topology of the islands. It was then necessary for each of the islands to be made completely observable by PMUs. This was done via the Standard AMM algorithm, treating each island as a network.

The results show which buses require PMUs for full observability of each island respectively. It is noted that the total amount of PMUs required for full observability of the entire network (8 PMUs) is more than that determined by the *Standard MM Right and Left sided algorithms* (6 PMUs).

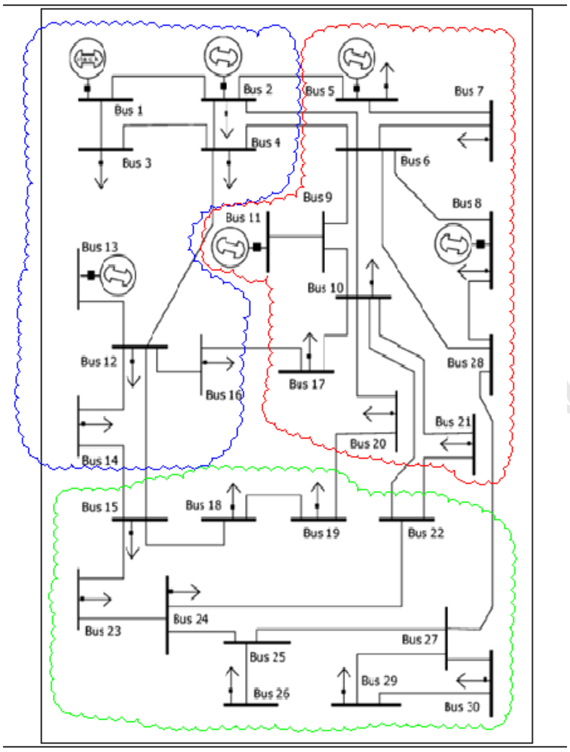

Figure 23: Selected Islands of the IEEE 30 bus network

In Islanding applications, the zero-injection bus advantage can be used as long as the buses which are observed through the zero-injection buses fall within the respective Island.

Despite being the least flexible of the 3 cases, at the moment Case 1 is probably the most realistic option. This is due to the high costs of PMUs.

5.6 AMM-ILP Hybrid Algorithm

Throughout the course of this thesis, it has been questioned what was the reason for developing another placement algorithm, when there already are others available. From the results it is true that without consideration for zero-injection buses, the Standard Advanced Matrix Manipulation provides mediocre results compared to ILP. However when zero-injection buses are compensated for (especially in large systems), this algorithm provides much better results than ILP. It was then realised that the advantage behind the standard Advanced Matrix Manipulation is its zero-injection bus compensator. For this reason, just as a matter of interest, the zero-injection bus compensator was combined with the ILP algorithm to form the ILP-AMM Hybrid Algorithm. The Hybrid algorithm does not have a Left and Right side approach as with the Standard AMM algorithm. Instead it has one algorithm approach to the optimal solution steered by the fundamentals of the ILP algorithm. This combination was tested on the 1009 bus system and its results are displayed below in Table 83:

Table 83: ILP-AMM Hybrid applied to 1009 bus system

Minimum Number of required PMUs	Bus positions of the PMUs					
174	3	204	376	554	688	838
	13	211	381	556	696	846
	21	225	382	562	698	852
	26	230	390	570	703	866
	28	232	392	573	704	867
	31	236	398	574	715	882
	63	237	399	575	719	893
	67	248	401	576	723	896
	72	254	417	585	732	903
	74	259	418	587	734	904
	78	265	422	589	738	906
	83	270	425	590	739	915
	88	282	431	593	744	930
	89	289	435	595	753	937
	97	290	439	597	759	948
	106	293	440	614	760	950
	110	301	442	616	765	959
	114	302	449	619	778	967
	127	312	451	627	780	974
	133	313	454	634	781	979
	136	316	457	636	785	980
	139	334	466	638	787	985
	148	344	484	644	790	991
	158	348	485	646	796	1002
	165	350	500	651	814	

	183	351	501	663	817	
	191	356	513	669	822	
	196	361	522	670	823	
	200	369	526	674	833	
	203	374	545	677	835	

This is an improvement to the standard Advanced Matrix Manipulation's result of 187 PMUs. The reason for being better than the standard AMM's solution is that ILP provides a better base result than the ANDing function developed in the standard AMM algorithm (where base refers to result with no zero-injection consideration). ILP on its own however, will not be able to minimise the solution to such an extent. The algorithms combined produce a better result than each one does by itself. Although, the ILP-AMM Hybrid's solution will not be significantly better than the standard AMM solution, as the ANDing function though imperfect, still does a good job. There is a general limit to the amount of PMUs that can be removed from the solution to leave the network completely observable after removing them. The standard AMM algorithm gets close to this amount, but the ILP-AMM algorithm gets closer.

The computer used is an Intel® 4 CPU 3.00 GHz with 960 MB of RAM

6. Conclusion

The main aim of this thesis was to place PMUs strategically within a Power network for the purpose of State Estimation. The task was to present an algorithm that provides placements that are cost effective and are as good as or better than solutions of other algorithms. It can be stated that this task was successfully completed in the thesis by presentation of the Standard Advanced Matrix Manipulation Algorithm. The algorithm successfully generated solutions which make the entire network completely observable through PMUs. These solutions were also quite similar and sometimes demonstrated to use less PMUs than the solutions generated by the ILP algorithm for the same respective networks.

When phasing of PMU installations is concerned, the objective of optimally scheduling the installations was accomplished through the development of the Placement for Inside Input and Placement for Outside Input algorithms. These algorithms serve to schedule the installations where different levels of requirements are addressed in terms of specific buses requiring a PMU, where these requirements are additional to the entire network being completely observed by PMUs. This ranges from no specific bus to several buses required to have a PMU. Placement for Inside Input deals with no requirement on any bus to have a PMU and Placement for Outside Input deals with multiple buses requiring a PMU at each bus. At large, the solutions provide the locations of the amount of PMUs funding is available for in each specific stage. These locations were based on those which provide the highest observability for surrounding buses without increasing the total minimum amount of required PMUs, as well as consideration of the specific buses requiring PMUs.

The Bad Data Detection Aid algorithm is another separately coded algorithm which was produced to assist Bad Data Detection methods of State Estimation. The engine of this algorithm, as with the phasing algorithms is based on the Standard Advanced Matrix Manipulation algorithm. The only change for the Bad Data Detection Aid algorithm was to give a variable redundancy level of observability of each bus. This means that every bus was to be observed the amount of times set by the user input. However this was limited to a redundancy level of 2 due to anything higher being impractical in terms of monetary issues concerning PMUs. The solutions show that by doubling the redundancy level, the amount of PMUs required does not double as well. This is due to taking advantage of the network topology to optimize the observability levels of each bus. Ofcourse some buses would be observed more than twice, but twice is the minimum. It is therefore concluded that the system monitoring will always be safer than what you require, in terms of redundant measurements.

The Islanding detection objective was achieved by using the Standard AMM algorithm. Detection of unsuspectedly live lines was explained to be picked up by a normal arrangement of PMUs observing the network, where these PMUs are to be powered from the buses they are connected to. This means that if there was to be a blackout and certain lines remained live due to distributed generation facilities, they would power the PMUs meant to observe them which would be followed by the PMU observing the presence of the live line.
When Islands were intentionally designed into the system network, independant observing of the islands by PMUs was necessary. A range of flexibility to the Island network designer was given from only choosing island areas from where the Standard AMM algorithm would place the PMUs to the situation where the designer does not have to take into account at all where the PMUs will be placed in

order to setup the islands. In the first case a normal Standard AMM algorithm is applied to the system and this information is presented to the designer. The latter required tweaking of the connectivity matrix to use each island as a separate network. In this way the Standard AMM was applied to each island to give a PMU placement schedule.

For all of the algorithms (besides the Standard AMM) presented in this thesis, the aim was to show when the user is given more flexibility, there is a tendency for an increase in the amount of required PMUs. The reasons being that the more specific the user requirements are, the stricter the monitoring conditions will become, which means more buses will require PMUs.

AMM is still not in its perfect state. If further work is to be done, perhaps an improved AND function can be developed for PMU placement. A perfect AND function will provide the absolute minimum for a binary vector problem such as the optimal placement of PMUs. The perfect AND was pursued and the current programming is the closest that could be achieved.

The Advanced Matrix Manipulation was found to produce very good results for systems where zero-injection buses are connected to other zero-injection buses. The zero-injection bus solver which was developed for the AMM algorithm is designed to take advantage of the bus topology where zero-injection buses are connected to other zero-injection buses. This topology is common in very large systems such as the utility case-file 1009 bus system tested in this thesis.
The advantage the AMM algorithm has over other algorithms when applied to such systems is therefore its zero-injection bus solver. It was then out of a matter of interest, that this zero-injection bus solver be combined with the Integer Linear Programming algorithm to form the Hybrid ILP-AMM algorithm. This algorithm was tested on the 1009 bus system and produced even better results than the Standard AMM algorithm. Where the Standard AMM calculated a requirement of 187 PMUs and the Hybrid ILP-AMM calculated 174 PMUs. ILP on its own however calculates a required amount of 226 PMUs. It must be stressed that all of these solutions are correct, for they all produce PMU placements that make the entire network observable, some just show that less PMUs are required when they are placed at different locations.

It can therefore be concluded that optimization algorithms are not perfect. Their results vary from each other because they converge to a final solution using different operational logic. These different styles of optimization work better on certain network topologies compared to others. The best strategy when pursuing the placement of PMUs is to use multiple placement algorithms on the same system to get a choice of results of which the best will be used. An example of this is the Standard AMM algorithm compared to the ILP algorithm when applied to the 1009 bus system.

The author therefore strongly recommends that the Advanced Matrix Manipulation algorithm be included as one of the optimization methods used to place PMUs within the Power network. The AMM algorithm style of optimization is different to most optimization methods since it is not subject to a cost function or constraint functions; it is completely Boolean Algebra. Its zero-injection bus solver may prove to be useful for various networks. The same would apply to the ILP-AMM algorithm.

When Phasing of PMU installations are a concern, the AMM algorithm offers different levels of user input flexibility to include users who do not have any specific preferences where the PMUs should be placed, to users who require a specific list of bus locations for PMU installations. It allows the user to simply define how many PMUs funding is available for during each stage of installation, then provides results instantly.

If it is desired that PMUs be placed to provide redundant measurements, then AMM's Bad Data Detection Aid algorithm will be able to place the PMUs using very similar logic to the Standard AMM algorithm. Each bus will be observed by the amount of PMUs set by the user, while the algorithm works hard at minimising the total number of PMUs to accomplish this.

Through specialised application of the Standard AMM algorithm, Islanded networks are made independently observable by PMUs. The Islanding criteria govern this method to strategically place the PMUs.

As Power Systems become more advanced and efficiently operated, it might not be long before PMUs are made an absolute priority for monitoring purposes. If that situation presents itself, then topics such as that covered in this thesis have the potential to largely benefit this cause and to save money through optimal designing. If any aspect of this thesis can promote the usefulness of Optimal PMU placement, then the efforts made for this work was not wasted.

7. Bibliography

[1] Robert E. Wilson, "Satelite Synchronisation measurements confirm power equation," 1994.

[2] M.M. Ahmed and W.L. Soo, "Supervisory Control and Data Acquisition System (SCADA) Based Customised Remote Terminal Unit (RTU) for Distribution Automation System," in *IEEE International Conference on Power and Energy*, Johor Baharu, 2008.

[3] Daniela Hossu, Ioana Fagarasan, Sergiu Stelian Iliescu, Daniel Razvan Costianu Nicoleta Arghira, "Modern SCADA Philosophy in Power System Operation," *U.P.B. Sci. Bull.*, vol. 73, no. 2, 2011.

[4] Marcus A. Donolo, "Advantages of Synchrophasor Measurements Over SCADA Measurements for Power System State Estimation," Schweitzer Engineering Laboratories, Inc, Hopkins Court, Pullman, USA, Application Note AN2006-10, 2006.

[5] Motorola, "SCADA Systems," Motorola, Schaumburg, USA, Company Product Information White Paper: SCADA Systems, 2007.

[6] P. Zhang, "Phasor Measurement Unit Implementation and applications," EPRI, March 2010.

[7] M. C. Valenti, and A. Feliachi B. Naduvathuparambil, "“Communication Delays in Wide Area Measurement Systems” ," in *Proceedings of the Thirty Fourth Southeastern Symposium*, 2002, pp. 118-122.

[8] "http://www.phasor-rtdms.com/phaserconcepts".

[9] Chuck Petras PE, Chris Anderson and Ken Fodero II Roy Moxley PE, "Display and Analysis of Transcontinental Synchrophasors," Schweitzer Engineering Laboratories, Inc, Pullman, WA USA, 2004.

[10] Eren Ersonmez, "Use Synchrophasors in SCADA/EMS," Schweitzer Engineering Laboratories, Inc, Hopkins Court, Pullman, USA, Application Note AN2009-26, 2009.

[11] Elias Karam, "Implementation and Simulation of Communication Network for Wide Area Monitoring and Control Systems in OPNET," KTH, Stockholm, Sweden, MSc Thesis 2008.

[12] Hongga Zhao, "A New State Estimation Model of Utilizing PMU Measurements," in *International Conference on Power System Technology*, 2006.

[13] Student Member IEEE, George K. Stefopoulos, Member IEEE, George J. Cokkinides, Senior Member IEEE and A.P. Meliopoulos, Fellow IEEE Evangelos Farantatos, "PMU-Based Dynamic State Estimation for Electric Power Systems," 2009.

[14] Antonio Gomez Exposito Ali Abur, *Power System State Estimation - Theory and Implementation*. New York , United States of America: Marcel Dekker, 2004.

[15] Michael Hurtgen and Jean-Claude Maun, "Advantages of Power System State Estimation using Phasor Measurement Units," in *16th PSCC*, Glasgow, Scotland, July 14-18 2008.

[16] K. Jamuna and K.S. Swarup, "Two Stage State Estimator with Phasor Measurements," in *2009 Third International Conference on Power Systems*, Kharagpur, India, December 2009.

[17] Member IEEE and A.G. Phadke, Life Fellow IEEE R.F. Nuqui, "Hybrid Linear State Estimation Utilizing Synchronised Phasor Measurements," *PowerTech*, pp. 1665-1669, 2007.

[18] S. Nouri-Zadeh, A.M. Ranjbar and M.R. Pishvaie S.A. Nezam-Sarmadi, "An Islanding Algorithm to Restore a PMU Installed Power System," , April 2010.

[19] L. H. Fink M. Adibi, "Power System Restoration Planning," *IEEE Transaction of Power Systems*, vol.

9, no. 1, pp. 22-28, February 1994.

[20] A. Monticellli and A. Garcia, "Reliable Bad Data Processing for Real Time State Estimation," *IEEE Trans. Power Applications Syst.*, vol. PAS-102, no. 5, pp. 1126-1139, May 1983.

[21] A.G. Phadke, "Synchronised Phasor Measurements in Power Systems," *IEEE Computer Applications in Power*, April 1993.

[22] Sweitzer Engineering Laboratories, "www.selinc.com," 2009.

[23] Member IEEE and Elias Kyriakides, Member IEEE Saikat Chakrabarti, "PMU Measurement Uncertainty Considerations in WLS State Estimation," *IEEE Transactions on Power Systems*, vol. 24, no. 2, May 2009.

[24] Sanjay Dambhare, Rajeev Kumar Gajbhiye, Student Member IEEE and S.A. Soman, Member IEEE Devesh Dua, "Optimal Multistage Scheduling of PMU Placement: An ILP Approach," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, October 2008.

[25] Member IEEE and Elias Kyriakides, Member IEEE Saikat Chakrabarti, "Optimal Placement of Phasor Measurement Units for Power System Observability," *IEEE Transactions on Power Systems*, vol. 23, no. 3, August 2008.

[26] L. Mili, M.B. Boisen and R. Adapa T.L. Baldwin, "Power System Observability with minimal phasor measurement placement," *IEEE Trans. Power Syst.*, vol. 8, no. 2, pp. 701-715, May 1993.

[27] T. Amraee, A.R. Shirani M. Hajian A.M. Ranjbar, "Optimal Placement of Phasor Measurement Units: Particle Swarm Optimization Approach," January 2008.

[28] Devesh Dua, Rajeev Kumar Gajbhiye and S.A. Soman Sanjay Dambhare, "Optimal Zero Injection Considerations in PMU Placement: An ILP Approach," in *16th PSCC*, Glasgow, Scotland, July 2008.

[29] Member IEEE and Arun G. Phadke, Life Fellow IEEE Reynaldo F. Nuqui, "Phasor Measurement Unit Placement Techniques for Complete and Incomplete Observability," *IEEE Transactions on Power Delivery*, vol. 20, no. 4, October 2005.

[30] Student Member IEEE, Caro Lucas, Senior Member IEEE, Amin Khodaei and Mahmud Fotuhi-Firuzabad, Senior Member IEEE Farrokh Aminifar, "Optimal Placement of Phasor Measurement Units using Immunity Genetic Algorithm," *IEEE Transactions on Power Delivery*, vol. 24, no. 3, July 2009.

[31] F. Garcia-Lagos, G. Joya and F. Sandoval F.J. Marin, "Genetic Algorithms for Optimal placement of Phasor Measurement units in Electrical Networks," *Electronic Letters*, vol. 39, no. 19, 18 September 2003.

[32] David A Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*. Singapore: World Scientific Publishing Co. Pte. Ltd, 1999.

[33] Brahim Gasbaoui and Boumediene Allaoua, "Ant Colony Optimization applied on Combinatorial Problem for Optimal Power Flow solution," bechar University, Faculty of Sciences and Technology, Department of Electrical Engineering, Bechar, Algeria,.

[34] Student Member, IEEE and Ali Abur, Fellow IEEE Jian Chen, "Placement of PMUs to Enable Bad Data Detection in State Estimation," *IEEE Transaction on Power Systems*, vol. 21, no. 4, November 2006.

[35] Nicola Locci, Carlo Muscas, IEEE and Sara Sulis, Member IEEE Andrea Carta, "A Flexible GPS-Based System for Synchronized Phasor Measurement in Electric Distribution Networks," *IEEE Transactions on InstrumentaTION AND mEASUREMENT*, vol. 57, no. 11, nOVEMBER 2008.

[36] Marcus M. Edvall and Anders Goran, "http://tomopt.com/tomlab," February 2009.

[37] Student Member IEEE and Ali Abur, Fellow IEEE Jian Chen, "Placement of PMUs to Enable Bad Data Detection in State Estimation," *IEEE Transaction on Power Systems*, vol. 21, no. 4, November 2006.

8. Appendices

This chapter displays the coding examples used for the IEEE 14, 30 and 57 bus systems for the ILP algorithm

8.1 IEEE 14 bus system with ILP formulation using CPLEX function

```
Name='Weingartner 1 - 2/28 0-1 knapsack';
% Problem formulated as a minimum problem
A = [1    1   0   0   1   0   0   0   0   0   0   0   0   0
   1    1   1   1   1   0   0   0   0   0   0   0   0   0
   0    1   1   1   0   0   0   0   0   0   0   0   0   0
   0    1   1   1   1   0   1   0   1   0   0   0   0   0
   1    1   0   1   1   0   0   0   0   0   0   0   0   0
   0    0   0   0   0   1   0   0   0   0   1   1   1   0
   0    0   0   1   0   0   1   1   1   0   0   0   0   0
   0    0   0   0   0   0   1   1   0   0   0   0   0   0
   0    0   0   1   0   0   1   0   1   1   0   0   0   1
   0    0   0   0   0   0   0   0   1   1   1   0   0   0
   0    0   0   0   0   1   0   0   0   1   1   0   0   0
   0    0   0   0   0   1   0   0   0   0   0   1   1   0
   0    0   0   0   0   1   0   0   0   0   0   1   1   1
   0    0   0   0   0   0   0   0   1   0   0   0   1   1];

b_U = [ inf inf inf inf inf inf inf inf inf inf inf inf inf inf]';

c   = [1 1 1 1 1 1 1 1 1 1 1 1 1 1]';      % 30 weights

% Make problem on standard form for mipSolve
[m,n]    = size(A);
x_L      = [ 0  0 0 0 0 0 0 0 0 0 0 0 0 0]';
x_U      = [1 1 1 1 1 1 1 1 1 1 1 1 1 1]';
x_0      =[];

fprintf('Knapsack problem. Variables %d. Knapsacks %d\n',n,m);

IntVars = [1:n];  % All original variables should be integer
x_min    = x_L; x_max   = x_U; f_Low   = -1E7; % f_Low <= f_optimal must hold
b_L       = [1 1 1 1 1 1 1 1 1 1 1 1 1 1]';
f_opt    = 10;

nProblem  = []; % Problem number not used
```

```
fIP        = []; % Do not use any prior knowledge
xIP        = []; % Do not use any prior knowledge
setupFile = []; % Just define the Prob structure, not any permanent setup file
x_opt      = []; % The optimal integer solution is not known
VarWeight = []; % No variable priorities, largest fractional part will be used
KNAPSACK  = 1;  % Run with the knapsack heuristic

% Assign routine for defining a MIP problem.
Prob       = mipAssign(c, A, b_L, b_U, x_L, x_U, x_0, Name, setupFile, ...
                      nProblem, IntVars, VarWeight, KNAPSACK, fIP, xIP, ...
                      f_Low, x_min, x_max, f_opt, x_opt);

Prob.optParam.IterPrint = 0; % Set to 1 to see iterations.
Prob.Solver.Alg = 2;    % Depth First, then Breadth search

% Calling driver routine tomRun to run the solver.

Result = tomRun('cplex', Prob, 3);
```

8.2 IEEE 30 bus system with ILP formulation using CPLEX function

```
Name='Weingartner 1 - 2/28 0-1 knapsack';
% Problem formulated as a minimum problem


A=[1    1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0
1   1   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
1   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   1   1   1   0   1   0   0   0   0   0   1   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   1   0   0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   1   0   1   0   1   1   1   1   1   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   1   0   0
0   0   0   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0
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   0   0   1   0   0
0   0   0   0   0   1   0   0   1   1   1   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   1   0   0   1   1   0   0   0   0   0   0   1   0   0   1
1   1   0   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   0   0
0   0   0   1   0   0   0   0   0   0   0   1   1   1   1   1   0   0   0   0
0   0   0   0   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   0   0
0   0   0   0   0   0   0   0   0   0   0   1   0   1   1   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   1   0   1   1   0   0   1   0   0
0   0   1   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   1   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   1   1   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   1
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   1   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   0   0   0   0   1   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   0   0   1   0   0   0   0   0   0   0   0   0   0
1   1   0   1   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
0   0   1   1   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   1   1   1   1   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   1   1   1   1   0   0   0
0   0   0   0   0   0   0   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   0   0   0   0   0   0   0   0   0
0   0   0   0   1   0   1   1   1   1
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   0   1   1   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   1   0   1   1
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   1   0   1   1
];


b_U = [ inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf]';  % 2 knapsack capacities
c   = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]';

% Make problem on standard form for mipSolve
[m,n]    = size(A);
x_L      = [ 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]';
x_U      = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]';
x_0      =[];

fprintf('Knapsack problem. Variables %d. Knapsacks %d\n',n,m);

IntVars = [1:n];  % All original variables should be integer
x_min    = x_L; x_max   = x_U; f_Low   = -1E7; % f_Low <= f_optimal must hold
b_L       = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]';
f_opt    = 9; % this the desired value, does not affect calculation

nProblem  = []; % Problem number not used
fIP       = []; % Do not use any prior knowledge
xIP       = []; % Do not use any prior knowledge
setupFile = []; % Just define the Prob structure, not any permanent setup file
x_opt     = []; % The optimal integer solution is not known
VarWeight = []; % No variable priorities, largest fractional part will be used
KNAPSACK  = 1;  % Run with the knapsack heuristic
```

```
% Assign routine for defining a MIP problem.
Prob       = mipAssign(c, A, b_L, b_U, x_L, x_U, x_0, Name, setupFile, ...
                       nProblem, IntVars, VarWeight, KNAPSACK, fIP, xIP, ...
                       f_Low, x_min, x_max, f_opt, x_opt);

Prob.optParam.IterPrint = 0; % Set to 1 to see iterations.
Prob.Solver.Alg = 2;    % Depth First, then Breadth search

% Calling driver routine tomRun to run the solver.

Result = tomRun('cplex', Prob, 3);
```

8.3 IEEE 57 bus system with ILP formulation using CPLEX function

```
Name='Weingartner 1 - 2/28 0-1 knapsack';
% Problem formulated as a minimum problem


A=[1    1   0   0   0   0   0   0   0   0   0   0   0   0   1   1   1   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   1   1   1   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   1   1   1   1   0   0   0   0   0   0   0   0   0   0   0   1   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   1   1   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   1   1   1   1   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   1   1   1   1   1   1   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0
0   0   0   0   0   0   0   0   1   1   0   1   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0
0   0   0   0   0   0   0   0   1   0   1   0   1   0   0   0   0   0   0   0
0   0   0   0   0   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   0   0   0   0   0   1   1   0   1   1   0   0   1   1   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
0    0    0    0    0    0    0    0    1    0    1    1    1    1    1    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    1    1    1    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0
1    0    1    0    0    0    0    0    0    0    0    0    1    1    1    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0
1    0    0    0    0    0    0    0    0    0    0    1    0    0    0    1    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
1    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    1    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1    1
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1
1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1
1    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
1    1    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    1    1    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    1    1    1    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    1    1    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    1    0    1    1    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    1    1    1    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    1    1    1    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    1    1    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    1    0    0    0    0    1    1    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    1    1    1    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    1    1    1    1    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    1    1    1    0    0    0    0    0    0
0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 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
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 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 1 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 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 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0
```

```
0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   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   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
1   1   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   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1
];

b_U = [ inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf ]';
% 2 knapsack capacities
c   = ones(1,57)';

% Make problem on standard form for mipSolve
[m,n]    = size(A);
x_L      = zeros(1,57)';
x_U      = ones(1,57)';
x_0      =[];

fprintf('Knapsack problem. Variables %d. Knapsacks %d\n',n,m);

IntVars = [1:n];  % All original variables should be integer
x_min   = x_L; x_max  = x_U; f_Low   = -1E7; % f_Low <= f_optimal must hold
b_L      = ones(1,57)';
f_opt   = 9; % this the desired value, does not affect calculation

nProblem  = []; % Problem number not used
fIP       = []; % Do not use any prior knowledge
xIP       = []; % Do not use any prior knowledge
setupFile = []; % Just define the Prob structure, not any permanent setup file
x_opt     = []; % The optimal integer solution is not known
VarWeight = []; % No variable priorities, largest fractional part will be used
KNAPSACK  = 1;  % Run with the knapsack heuristic

% Assign routine for defining a MIP problem.
Prob      = mipAssign(c, A, b_L, b_U, x_L, x_U, x_0, Name, setupFile, ...
                      nProblem, IntVars, VarWeight, KNAPSACK, fIP, xIP, ...
                      f_Low, x_min, x_max, f_opt, x_opt);

Prob.optParam.IterPrint = 0; % Set to 1 to see iterations.
Prob.Solver.Alg = 2;    % Depth First, then Breadth search

% Calling driver routine tomRun to run the solver.

Result = tomRun('cplex', Prob, 3);
```